University of Zagreb
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Marija Majda Perišić

MULTI-AGENT SYSTEM FOR SIMULATION OF TEAM BEHAVIOUR IN PRODUCT DEVELOPMENT

DOCTORAL THESIS

Zagreb, 2020

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Marija Majda Perišić

MULTI-AGENT SYSTEM FOR SIMULATION OF TEAM BEHAVIOUR IN PRODUCT DEVELOPMENT

DOCTORAL THESIS

Supervisors:
Associate Professor Vedran Podobnik, Ph.D.
Professor Mario Štorga, Ph.D.

Zagreb, 2020

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Marija Majda Perišić

VIŠEAGENTSKI SUSTAV ZA SIMULACIJU PONAŠANJA TIMA U RAZVOJU PROIZVODA

DOKTORSKI RAD

Mentori:
Izv. prof. dr. sc. Vedran Podobnik
Prof. dr. sc. Mario Štorga

Zagreb, 2020.

Doctoral thesis was made at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Telecommunications and University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Department of Design.

Supervisors:

Associate Professor Vedran Podobnik, Ph.D.

Professor Mario Štorga, Ph.D.

Doctoral thesis contains: 237 pages

Doctoral thesis number: __________

ABOUT THE SUPERVISORS

Vedran Podobnik was born in Zagreb in 1982. He received M.Eng. (2006, Electrical Engineering) and Ph.D. (2010, Computer Science) degrees from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, as well as M.Phil. (2013, Technology Policy) degree from the University of Cambridge, Judge Business School, Cambridge, UK.

From 2006, he works at the Department of Telecommunications at FER (from 2016 as the Associate Professor). He is the founder and Director of the “Social Networking and Computing Laboratory (socialLAB)” and co-founder of the “FER’s student startup incubator SPOCK”. He led several national and international scientific and industrial projects. Currently, he is a work package leader in the only national centre of research excellence (CoRE) in the field of technical sciences, the “CoRE for Data Science and Cooperative Systems”, and the management board member at FER’s “Center for Artificial Intelligence”. His teaching and research activities are in transdisciplinary fields of network and data science, social computing, and technology policy. He co-authored over 100 scientific and professional papers, including publications in Information Sciences, Information Technology & People, International Journal of Energy Research and AI Magazine journals. From 2018, he advises the global technology company Hewlett Packard Enterprise in the fields of data platforms, data analytics and artificial intelligence. He received scientific titles in two fields of engineering - electrical engineering and computer science.

Assoc. Prof. Podobnik is a member of IEEE, ACM, INFORMS, AIS and KES International associations, as well as the Cambridge Union Society. He has participated in the program and organizing committees of many scientific conferences and summer schools, and has served as a peer reviewer in various international journals. He coordinated an international team that received the “Success Story Award” for a particularly successful ERASMUS+ Strategic Partnerships project, the first such award in the higher education sector in Croatia (2018, awarded by the European Commission). He was a leader of an interdisciplinary team which was awarded the highest national award for notable achievements in the education activity (2015, awarded by the Croatian Parliament) as well as the annual national PMI Project of the Year Award (2016, awarded by the world’s leading project management professionals association PMI). As a junior researcher, he received the Croatian Annual National Award for Science in the field of technical sciences (2011, awarded by the Croatian Parliament), as well as the Silver Medal “Josip Loncar” award for outstanding doctoral dissertation and particularly successful scientific research (2010, awarded by FER).

Mario Štorga was born in Varaždin, Croatia in 1974. He acquired the PhD degree in September 2005 after having completed PhD thesis titled “The Design Ontology - Contribution to the Design Knowledge Exchange and Management” at University of Zagreb Faculty of Mechanical Engineering and Naval Architecture (UNIZG FSB). He was promoted to the Scientific Adviser Tenured in 2017 and is the Full professor at UNIZG FSB from 2018. From November 2016 to September 2019 he was the Head of the Chair of Design and Product Development.

He was visiting researcher at the Technical University of Denmark, Ecole Centrale de Paris France, University of Bath UK, Stevens Institute of Technology USA, Lulea University of Technology Sweden and Japan Advanced Institute of Science and Technology. In 2014/15 he has been Distinguish Research Fellow of Royal Academy of Engineering UK at the University of Bath University of Bristol UK. Since October 2015, he is continuously a Visiting Professor at Lulea University of Technology Sweden. He participated in 9 multidisciplinary scientific projects (5 national and 4 international) in the role of the principal investigator or associate and 7 research and development projects for industry. Currently is a principal investigator of the Team Adaptability for Innovation-Oriented Product Development – TAIDE project funded by Croatian Science Foundation, (2018-2022). He published over 100 scientific papers (journal and conference) and more than 50 industrial reports. He was invited speaker for numerous academic and industrial workshops and seminars in Croatia and abroad.

Since 1998, Prof. Štorga is serving on the organisational and scientific boards for International DESIGN conference biannually held in Croatia with more than (300 international participants at every event. He is a reviewer for 6 international journals and is member of the Editorial Board of the Journal of Engineering Design - JoED (Taylor and Francis) and the International Journal of Design Creativity and Innovation (Taylor&Francis UK). He served on the scientific advisory boards for ICED (Europe, USA, Australia), DCC (Europe, USA), ASME (USA) and ICoRD (India) engineering design conferences series. Since 2011, he is a member of the Advisory Board of the international organisation the Design Society and has been involved in the organisation of the Summer School for Engineering Design Research. For the paper published in the Journal of Engineering Design (Q1, Taylor and Francis, UK) he was awarded with the best paper in 2015 award. With two co-editors he edited the scientific book Experimental Design Research - Approaches, Perspectives, Applications, that was in 2016 published by Springer.

O MENTORIMA

Vedran Podobnik rođen je u Zagrebu 1982. godine. Diplomirao je u polju elektrotehnike te doktorirao u polju računarstva na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER), 2006. odnosno 2010. godine. Također je magistrirao u području upravljanja tehnologijom 2013. godine na Sveučilištu u Cambridgeu, Judge Business School (Ujedinjeno Kraljevstvo).

Od 2006. godine radi na Zavodu za telekomunikacije FER-a (od 2016. kao izvanredni profesor). Utemeljitelj je i voditelj "Laboratorija za društveno umrežavanje i društveno računarstvo (socialLAB)" te jedan od utemeljitelja "FER-ovog studentskog startup inkubatora SPOCK". Bio je voditelj većeg broja domaćih i međunarodnih znanstvenih projekata te projekata suradnje s gospodarstvom. Trenutno je voditelj radnog paketa u jedinom nacionalnom znanstvenom centru izvrsnosti (ZCI) u području tehničkih znanosti "ZCI za znanost o podatcima i kooperativne sustave" te član Upravljačkog vijeća FER-ovog "Centra za umjetnu inteligenciju". Provodi nastavne i istraživačke aktivnosti u transdisciplinarnim područjima znanosti o mrežama i podacima, društvenog računarstva i upravljanja tehnologijom. Objavio je više od 100 znanstvenih i stručnih radova, uključujući radove u časopisima Information Sciences, Information Technology & People, International Journal of Energy Research i AI Magazine. Od 2018. godine savjetuje globalnu tehnološku kompaniju Hewlett Packard Enterprise (HPE) u području podatkovnih platformi, analitike te umjetne inteligencije. Izabran je u znanstvena zvanja u dva polja tehničkih znanosti – elektrotehnici i računarstvu.

Izv. prof. Podobnik član je stručnih udruga IEEE, ACM, INFORMS, AIS i KES International te društva Cambridge Union Society. Sudjelovao je u programskim i organizacijskim odborima mnogih znanstvenih konferencija i ljetnih škola, te sudjeluje kao recenzent u većem broju inozemnih časopisa. Koordinirao je međunarodni tim koji je dobio priznanje "Success Story" za iznimno uspješno provedeni projekt u programu ERASMUS+ Strateška partnerstva, što je prva takva nagrada u sektoru visokog obrazovanja u Republici Hrvatskoj (2018., dodijelila Europska komisija). Bio je voditelj interdisciplinarnog tima koji je nagrađen najvišom hrvatskom državnom nagradom u području edukacije (2015., dodijelio Hrvatski sabor) te godišnjom nacionalnom nagradom PMI Projekt godine (2016., dodijelila vodeća svjetska profesionalna udruga projektnih menadžera PMI). Kao mladi istraživač primio je godišnju nacionalnu nagradu za znanost u području tehničkih znanosti (2011., dodijelio Hrvatski sabor), kao i srebrnu plaketu "Josip Lončar" za posebno istaknutu doktorsku disertaciju (2010., dodijelio FER).

Mario Štorga rođen je u Varaždinu 1974. godine. Doktorirao je na Fakultetu strojarstva i brodogradnje Sveučilišta u Zagrebu 2005. godine s temom "Model rječnika za računalnu razmjenu informacija u distribuiranom razvoju proizvoda". U zvanje znanstvanog savjetnika u trajnom zvanju izabran je 2017. godine, a u zvanje redovitog profesora na FSB 2018 godine, Od studenog 2016. godine do rujna 2019. godine bio je voditelj Katedre za konstruiranje i razvoj proizvoda.

Bio je gostujući znanstvenik na Technical University of Denmark, Ecole Centrale Paris, University of Bath UK, Lulea University of Technology Švedska, Stevens Institute of Technology SAD, Singapore University of Technology and Design – MIT International Design Centre, te Japan Advance Institute of Science and Technology. Royal Acedemy of Engineering UK, mu je 2014. dodijelila stipendiju Distinguish Research Fellow, na University of Bath i University of Bristol UK. Od listopada 2015. godine kontinuirano je angažiran kao gostujući profesor na Lulea University of Technology Švedska. Bio je uključen u 9 multidisciplinarnih znanstvenih projekata (5 nacionalna i 4 međunarodna) u kojima je imao ulogu suradnika i voditelja, te u 7 razvojno-istraživačkih projekata za industriju. Trenutno je glavni istraživač na HRZZ projektu TAIDE – timska adaptabilnost u razvoju inovativnih proizvoda (2018-2022.). Do sada je objavio 100-njak znanstvenih, te 50-ak stručnih radova, te je bio pozvani predavač na mnogobrojnim skupovima u Hrvatskoj i inozemstvu.

Od 1998. godine član je organizacijskog i znanstvenog odbora međunarodnih konferencija iz serije DESIGN koje se u dvogodišnjem ciklusu održavaju u Hrvatskoj s preko 300-njak sudionika. Obavlja i ulogu recenzenta u 6 međunarodnih časopisa, a član je uredničkog odbora časopisa Journal of Engineering Design (Taylor&Francis UK) te International Journal of Design Creativity and Innovation (Taylor&Francis UK). Član je znanstvenih odbora konferencija iz serija ICED (Europa, Azija, Sjeverna Amerika, Australija), DCC (Europa, SAD), ASME (SAD), te ICoRD (Indija). Član je međunarodnog udruženja The Design Society u kojem je 2011. izabran u savjetodavni odbor. Više puta je sudjelovao u organizaciji ljetne škole Summer School on Engineering Design Research u suradnji s kolegama sa stranih sveučilišta. Za rad objavljen u Journal of Engineering Design (Q1, Taylor and Francis, UK) 2015. godine, nagrađen je nagradom časopisa za najbolji rad u 2015. godini. S dvojicom ko-urednika uredio je znanstvenu knjigu "Experimental Design Research - Approaches, Perspectives, Applications" koju je 2016. godine izdao Springer.

ACKNOWLEDGEMENT

First and foremost, I wish to thank my supervisors: Associate professor Vedran Podobnik, for always being supportive and approachable, and Professor Mario Štorga, for his guidance, encouragement, and willingness to answer each of my questions with incredible patience and knowledge. Similarly, I would like to express my sincere gratitude to Professor John Gero, whose insights and expertise shaped my research and way of thinking in more ways than I could count. I would also like to thank the examiners, Professor Ignac Lovrek and Professor Mario Kušek for reviewing the thesis, but also for their continuous support during every stage of my PhD studies.

This thesis would not exist if it weren't for my family and friends. Thank you, mama, tata, Petar Lucija, Pavla, baka, nona, and the rest of the family – for every hug, talk, message, joke and encouragement. Thanks also go to all of my friends who spent countless hours listening to my PhD worries and did not give up believing in me. Special thanks to Petra - for our Saturday 'rituals', Nera - for long conversations, and Barbara - for daily messages. Finally, I would like to thank each and every one of my dear colleagues and friends at the Chair of design and product development (and a few others) - I could not have wished for the better work environment.

ABSTRACT

Teams play a crucial role in many product development activities. However, teams - as complex socio-technical systems - are challenging to study in a real-world setting. Computational models and simulations offer easily controllable and cost-effective tools for supporting team studies. Computational models prove especially valuable in studies on the sufficient conditions for the emergence of various team properties and behaviours. As a consequence, the computational models offer a possible explanation of the observed phenomena and provide means for theory-testing and hypothesis generation. To support cognitive studies on product development teams, this dissertation develops a multi-agent system capable of representing various aspects of product development teams. Building on the existing theories, empirical studies and current teamwork models, this work derives a detail theoretical model of a designer, its cognition, its tasks, and interactions with others. The thesis then implements the theoretical model in a multi-agent system directed at studies of the emergence of team properties and behaviours – in particular, team learning and adaptation. Extensive testing of the derived computational model concludes the goal of obtaining a multi-purpose research tool for studies of product development teams and confirms the appropriateness of the chosen simulation technique – agent-based modelling - for the intended research goal.

Keywords:

Agent-based modelling and simulations; product development teams; emergent team properties; team learning; team adaptation

VIŠEAGENTSKI SUSTAV ZA SIMULACIJU PONAŠANJA TIMA U RAZVOJU PROIZVODA

Timski rad je neizostavan element djelovanja modernih organizacija. Kako bi uspješno odgovorile na zahtjeve tržišta, organizacije se često oslanjaju na suradnju stručnjaka iz različitih područja, nadajući se da će komplementarnost njihovih znanja i vještina rezultirati tržišnom prednošću. Dosadašnja istraživanja potvrđuju mnoge prednosti timskog rada (u usporedbi s izoliranim djelovanjem pojedinaca). Primjerice, razni izvori ističu da timski rad rezultira bržim i jeftinijim razvojem proizvoda, većom inovativnošću te kvalitetom proizvoda, kao i boljim rješavanjem problema. No mnoga istraživanja naglašavaju probleme do kojih dolazi pri timskom radu. Poznati su slučajevi gdje želja za postizanjem dogovora rezultira stvaranjem malog broja ideja, nedostatnim razmatranjem opcija ili prihvaćanjem suboptimalnih rješenja (eng. *groupthink*). U drugim slučajevima, pak, ponekad dolazi do konflikata među sudionicima, gdje razlike u prioritetima i viđenju problema onemogućavaju tim u donošenju odluka. Mnogi istraživači, stoga, naglašavaju potrebu za dodatnim istraživanjima usmjerenim ka razumijevanju timskih procesa, timskog ponašanja i svojstava tima koja proizlaze iz interakcija članova tima. Posebno je istaknuta važnost potrebe za dodatnim istraživanjima razvoja timskog znanja i iskustva, tj. kako se znanje stvara, dijeli te koristi unutar tima.

Kako bi proučavali timove i njihovo ponašanje, istraživači se često oslanjaju na laboratorijske studije u kojima su članovi tima suočeni s nekim problemom te zajednički pokušavaju doći do rješenja. Pri ovakvim istraživanjima sakupljaju se podaci o timskim procesima obrade informacija te detalji o interakcijama među članovima tima. Međutim, provedba laboratorijskih studija je skupa i vremenski zahtjevna. Jedna od metoda kojima se može potpomoći laboratorijske studije je računalno modeliranje i simulacije. Korištenjem računalnih modela timskog rada moguće je izvršiti veliki broja istraživanja u kratkom vremenu. Pritom računalni modeli dopuštaju ponavljanje studija, kontrolu i praćenje varijabli te modifikacije ulaznih podataka kako bi se promatrao njihov utjecaj na ishod simulacije. Navedene karakteristike čine računalne modele važnim alatom pri proučavanju sustava na čije ponašanje utječe velik broj teško mjerljivih čimbenika. Izradom računalnog modela timskog rada omogućava se efikasno testiranje postojećih teorija te formiranje novih hipoteza, posljedično utječući na buduće empirijske studije.

Cilj ovog istraživanja je razvoj i validacija teorijskog i računalnog modela timskog rada u razvoju proizvoda. Posebice, željeni model timskog rada treba omogućiti simulaciju i proučavanje sposobnosti tima da uči te se prilagodi promjenama u vanjskim ili unutarnjim okolnostima koje se javljaju tijekom aktivnosti razvoja proizvoda. Ovim istraživanjem verificira se hipoteza da je modeliranje zasnovano na agentskoj paradigmi (ABM) prikladno za ispunjenje cilja istraživanja: proučavanje simuliranog ponašanja timova u razvoju proizvoda te proučavanje prilagodbe tima kao posljedice promjene u unutarnjim ili vanjskim okolnostima koje se javljaju tijekom aktivnosti razvoja proizvoda.

Razvoj željenog modela prati istraživačku metodologiju po kojoj se istraživanje može se podijeliti u tri povezana ciklusa aktivnosti. Prvi ciklus, Ciklus definiranja utjecaja, sastoji se od pregleda područja čiji cilj je identifikacija važnih koncepata, odabir odgovarajućih mjera te stvaranje preliminarnih modela raznih aspekata timskog rada. Kao rezultat ove faze istraživanja postiže se razumijevanje problema, njegovih granica i postojećih rješenja, te se odabire pogodna tehnika za izradu željenog modela timskog rada. U drugoj fazi istraživanja, Ciklusu razvoja modela, preliminarni modeli nastali u prvoj fazi istraživanja se detaljiraju te implementiraju koristeći odabranu tehniku izrade računalnog modela. Razvijeni modeli se testiraju kako bi se verificirala implementacija te osigurala podudarnost s pojavama uočenim u empirijskim studijama. Konačno, u trećoj fazi razvoja, Ciklusu utvrđivanja valjanosti, razvijeni modeli integriraju se u koherentni istraživački okvir te se isti primjenjuje u simulacijama namijenjenim proučavanju timskog učenja i adaptabilnosti. Ishodi simulacija se uspoređuju s predviđanjima zasnovanim teorijama i podacima pronađenim u dostupnoj literaturi, što rezultira identifikacijom prednosti, ali i nedostataka razvijenog računalnog modela timskog rada. Otkriveni nedostaci, kao i rezultati daljnjih empirijskih istraživanja timskog rada, služe kao smjernice za unaprjeđenje modela. S druge strane, razvijeni model omogućava istraživačima korištenje simulacija kako bi povećali razumijevanje međudjelovanja raznih aspekata timskog rada, čime produbljuje znanje o sustavu i usmjerava buduća empirijska istraživanja.

Doktorski rad strukturiran je tako da prati opisanu metodologiju istraživanja. Podijeljen je u devet poglavlja od kojih prva tri odgovaraju Ciklusu definiranja utjecaja. Četvrto, peto i šesto poglavlje opisuju razvoj i testiranje računalnog modela čime prikazuju drugu fazu istraživanja – Ciklus razvoja modela. Rezultati posljednje faze istraživanja, Ciklusa utvrđivanja valjanosti, opisani su u sedmom poglavlju.

Preciznije, prvo poglavlje („*Introduction*") donosi uvod u temu doktorskog rada. Motivacija provedenog istraživanja dana je kroz opis potrebe za istraživanjima timskog rada u razvoju proizvoda, a posebice potreba za studijama timskog učenja i prilagodbe. Naglašena je potencijalna korist razvoja računalnog modela za simulaciju i istraživanje timskog rada, čime je istaknut cilj istraživanja. Nadalje, opisana je hipoteza, očekivani znanstveni doprinosi disertacije te metodologija istraživanja. Poglavlje završava pregledom ostalih poglavlja doktorskog rada.

Drugo poglavlje („*Research background*“) uvodi osnovne pojmove definirajući važne koncepte vezane uz timove, timsko ponašanje, timske procese te timska svojstva koja proizlaze iz interakcija pojedinaca u timu. Posebna pažnja posvećena je identifikaciji elemenata koje je potrebno modelirati kako bi se prikazalo djelovanje tima. Opisana je uloga timova u razvoju proizvoda, kao i poteškoće s kojima se susreću istraživači pri proučavanju timskog rada. Konačno, nekoliko računalnih tehnika za modeliranje i simulaciju kompleksnih sustava je prikazano i uspoređeno s obzirom na njihovu prikladnost za dani problem. Modeliranje pomoću mreža (NS), modeliranje diskretnih događaja (DES), sistemska dinamika (SD) te modeliranje zasnovano na agentskoj paradigmi (ABM) su izdvojene kao najčešće korištene tehnike modeliranja. Raspravljena je njihova sposobnost da prikažu inteligentno ponašanje pojedinaca, hijerarhijske veze i interakcije među pojedincima, nelinearnost i dinamičnost sustava, kao i druge važne aspekte timskog rada, čime je ustanovljeno da je modeliranje zasnovano na agentima najpogodnija tehnika za razvoj željenog modela timskog rada. Nekoliko postojećih modela timskog rada (u raznim domenama) izrađenih ABM tehnikom je opisano kako bi se prikazale mogućnosti odabrane tehnike.

U trećem poglavlju („*Related work: Agent-based models of product development teams*“) prikazani su rezultati sistematskog pregleda literature usmjerenog ka identifikaciji i analizi postojećih agentskih modela timskog rada u razvoju proizvoda. Četrdeset modela uspoređeno je s obzirom na njihovu namjenu, domenu primjene te karakteristike prikaza pojedinačnih članova tima, interakcija među članovima te timskog okruženja u vidu prikaza konstrukcijskog problema i resursa. Detalji o pojedinom modelu izdvojeni su u Dodatku A („*Scope and characteristics of agent-based models of product development teamwork*“). Prikazana analiza ističe nedostatke promatranih modela, ali i identificira mnoga važna postojeća rješenja koja stvaraju podlogu za razvoj željenog modela timskog rada.

Četvrtim poglavljem („*Model specification and theoretical foundation*“) uvodi se teorijska pozadina razvijenog modela timskog rada u razvoju proizvoda. Kako bi se utvrdili aspekti ljudskog ponašanja čije modeliranje je ključno za vjerodostojni prikaz ponašanja čovjeka u društvenom okruženju, analizirani su postojeći istraživački okviri i modeli pojedinca. Pri modeliranju kognitivnih procesa čovjeka, dosadašnje studije naglašavaju važnost modeliranja ograničenosti ljudskog znanja i vremena pri donošenju odluka; raznolikosti čovjekovih sposobnosti i karakteristika ličnosti; te sklonost predrasudama i razvijanju obrazaca ponašanja proizašlih iz prethodnih iskustava. Stoga se kao važne komponente modela čovjeka ističu: prikaz njegova znanja, prethodnih iskustava, procesa obrade informacija, percepcije situacije (npr. percepcija drugih članova tima i viđenje problema s kojim su suočeni), i dinamike mentalnog modela, tj. učenje i zaboravljanje; prikaz čovjekove osobnosti i mentalnih kapaciteta; te prikaz čovjekova emotivnog stanja koje utječe na mentalne procese. Na temelju literature iz domena psihologije, sociologije, organizacijskih znanosti, kognitivnih znanosti, razvoja proizvoda te računalnog modeliranja i simuliranja, predložena je arhitektura agenta kojom se obuhvaćaju identificirane važne komponente modela čovjeka. Uz navedenu arhitekturu, ovo poglavlje izlaže teorijsku pozadinu pojedine komponente te rezultira skupom zahtjeva koje višeagentski sustav za simulaciju ponašanja tima treba ispuniti kako bi se postigao cilj doktorskog rada.

Peto poglavlje („*Model implementation*“) opisuje implementaciju razvijenog višeagentskog sustava za simulaciju tima u razvoju proizvoda. U razvijenom modelu, agent predstavlja pojedinca, tj. jednog člana razvojnog tima. Najsloženiji aspekt agenta je njegov mentalni model. Procesi rasuđivanja modelirani su kao širenje aktivacije po mreži agentova znanja. Agentovo znanje apstrahirano je oslanjajući na FBS ontologiju (Funkcija – Ponašanje - Struktura), time omogućavajući razlučivanje pojedinih procesa konstruiranja poput analize, sinteze i evaluacije. Modelirani su i mehanizmi inhibicije, učenja, prisjećanja, određivanje prioriteta, stvaranja novih struktura i veza među jedinicima znanja, kao i grupiranje jedinica znanja u veće elemente. Dodatno, razvijeni agent je karakteriziran svojom osobnošću, kognitivnim kapacitetom te emotivnim stanjem, a sposoban je i formirati percepciju svojih suradnika te na osnovu njih izgraditi povjerenje u druge te odlučivati o suradnji. Da bi se omogućila interakcija među agentima, modelirana su pravila i utjecaji komunikacije i interakcije među agentima. Razvijeni model implementiran je u alatu MASON te je pri implementaciji korišten modularni pristup prikazu pojedinih komponenti sustava. Primjerice, emotivna komponenta, utjecaj karaktera ili komponenta koja regulira percepciju drugih sudionika mogu jednostavnim odabirom biti isključeni iz simulacije. Time je omogućena veća sloboda pri provođenju eksperimenata te je olakšana verifikacija i validacija modela.

Implementacija svih elemenata oslanja se na dostupne teorije i empirijske studije, ali i na poznate kognitivne arhitekture (za općenite namjene). U razvijenom modelu se, tako, općenite kognitivne teorije i arhitekture stapaju s teorijama o konstruiranju, stvarajući kognitivni model pojedinca u razvoju proizvoda. Razvijeni agent je sposoban stvarati nove strukture, donositi zaključke, produbljivati svoje znanje, učiti i zaboravljati – što mu, posljedično, omogućava da promijeni svoje ponašanje na osnovu prethodnog iskustva. Ova karakteristika ključna je kako bi se simulirala dinamika timskog znanja te promjena timskog ponašanja koja proizlazi iz interakcija članova tima.

Šesto poglavlje („*Verification and validation of the developed model*“) prikazuje rezultate testiranja raznih komponenti razvijenog modela. Testiranje modela, u svrhu njegove validacije i verifikacije, provedeno je hijerarhijski: od simulacije jednostavnih scenarija prema kompleksnijim. Tako je model prvo testiran u okruženju gdje samo jedan agent rješava zadatke. Pritom je promatrano agentovo ponašanje na jednom zadatku, potom i kako se to ponašanje mijenja kada je agent opetovano suočen s istim zadatkom te, konačno, kako se agentovo ponašanje mijenja tijekom rješavanja niza različitih zadataka. Na ovaj način redom su testirani: agentova sposobnost učenja, zaboravljanje nekorištenih jedinica znanja, inhibicija te utjecaji emocija, prethodnih sjećanja, područja stručnosti te kognitivne sposobnosti. Po završetku testiranja agentova ponašanja pri izoliranom radu, model je testiran na scenarijima koji uključuju više agenata. Učestalost, sadržaj te utjecaj komunikacije među agentima su promatrani kako bi se osigurala valjanost implementacije interakcija među agentima. Potom je promatran utjecaj agentove osobnosti - po pitanju ekstraverzije i ugodnosti - na uspjeh tima i dinamiku timskih procesa. Konačno, testiran je utjecaj povjerenja među agentima na ishod timske aktivnosti. Navedeni eksperimenti provedeni su s ciljem prikazivanja sposobnosti razvijenog računalnog modela da simulira obrasce uočene u stvarnom svijetu, no i da ukažu na nedostatke razvijenog modela čime se daju smjernice za njegov daljnji razvoj.

U sedmom poglavlju („*Study of design team learning and adaptation*“), razvijeni model je upotrijebljen da bi se proučavale promjene ponašanja timova u razvoju proizvoda, time omogućavajući praćenje timske sposobnosti učenja i prilagodbe. Prikazani su rezultati dvaju studija. Prva je za cilj imala ispitivanje kako se timsko ponašanje mijenja kada je tim ponovno suočen s nekim zadatkom. Između dvaju izvršavanja danog zadatka, agenti su suočeni s nizom drugih zadataka čime im je omogućeno učenje te sakupljanje iskustva. Uspoređujući uspjeh tima pri prvom i drugom izvršavanju zadatka dobivena su saznanja o utjecaju stečenog iskustva na timsko ponašanje. Druga studija je usmjerena ka ispitivanju trendova u timskom ponašanju na nizu od nekoliko različitih zadataka. Da bi se izolirao utjecaj učenja i iskustva na timsko ponašanje, timski uspjeh na pojedinom zadatku uspoređen je s onim koji bi dani tim agenata postigao da nije bio izložen prijašnjim zadacima. U obje studije kao metrike uspješnosti tima korišteni su postotak uspješno riješenih zadataka te brzina konvergencije ka rješenju (tj. broj simulacijskih koraka), a kao pokazatelji timskog učenja izdvojeni su podaci o komunikaciji među agentima te broju novonastalih elemenata znanja. Rezultati provedenih simulacija uspoređeni su s postojećim teorijama i zaključcima empirijskih studija, te dokazuju da razvijeni model udovoljava uvjetima za proučavanje ponašanja i prilagodbe tima u razvoju proizvoda, omogućavajući vrednovanje teorijskih postavki i testiranje hipoteza.

Osmo poglavlje („*An example of developed model's usage*") opisuje primjer uporabe razvijenog modela pri proučavanju kognitivnog ponašanja timova u razvoju proizvoda. Preciznije, u danom primjeru, model je korišten kako bi se ispitao utjecaj timskih kognitivnih aktivnosti na procjenu jedne od ključnih komponenti kreativnosti – noviteta predloženih rješenja. Novitet pojedine strukture aproksimiran je njenom udaljenošću od postojećih rješenja (u prostoru definiranom pomoću nekoliko svojstava struktura). Uporabom modela moguće je pratiti širenje prostora postojećih rješenja te ispitati koje bi strukture bile smatrane novima u pojedinom trenutku simulacije. Na ovaj način uočava se da, razvojem prostora rješenja, neke strukture prestaju biti smatrane novima, no i da (kako se prostor rješenja proširuje u određenom smjeru) neke strukture mogu ponovno početi biti smatrane dovoljno drugačijima od drugih.

Doktorski rad zaključen je devetim poglavljem („*Conclusion*") u kojem je dan osvrt na nedostatke provedenog istraživanja i razvijenog modela, ali je i istaknut ostvareni znanstveni cilj te potvrda postavljene hipoteze. Rad prikazan u ovoj disertaciji rezultirao je trima izvornim znanstvenim doprinosima.

1. Razvijen je teorijski model timskog rada u razvoju proizvoda. Razvijeni model se sastoji od modela pojedinog člana tima, modela timskog okruženja u vidu zadataka, resursa i organizacijskog konteksta, te definicije mehanizama koji opisuju interakcije među članovima tima, te između članova tima i njihove okoline.
2. Izrađen je računalni prototip, temeljen na teorijskom modelu timskog rada te implementiran korištenjem višeagentske paradigme, koji se koristi za proučavanje ponašanja tima i timskih osobina koje proizlaze iz interakcija članova tima.
3. Osmišljen je istraživački okvir za kalibraciju i validaciju razvijenog modela i računalnog prototipa timskog ponašanja, s posebnim naglaskom na timsko učenje i prilagodbu.

Ključne riječi: Modeliranje i simulacije zasnovane na agentima; timovi u razvoju proizvoda; svojstva tima koja proizlaze iz interakcija njegovih članova; timsko učenje; timska prilagodba.

CONTENTS

1 INTRODUCTION

The first chapter serves to introduce the research goal and clarify the scope of the research with regards to computer science and design science. The chapter describes the motivation driving the work, explains research aims and hypothesis, and outlines the methodology adopted in the conducted research. The expected scientific contribution is discussed. Finally, a short overview of the dissertation outline is presented.

1.1 Motivation

Over the past few decades, organisational structures transformed from mostly revolving around individual jobs to the teams-based organisation of work [1]. The global competition and pressure for innovation drive a need for intertwining diverse skills and expertise in order to meet the market demands and adapt to economic and technical changes [2]. As a result of such trends, teams have emerged as an essential component of organisational work. Thus, in [3], Edmondson and Nembhard commence their work with a statement:

> *The value of teams in new product development is undeniable*.

Indeed, the complexity of engineering projects introduces the necessity of contributions from multiple specialists [4], [5], rendering teams as the core building blocks of product development (or product design) organisations. Reported benefits of teamwork (as opposed to working in isolation) range from advancement in speed to market, increase in levels of innovation, reduction in development cost, and improvement in product quality [6], to superior performance in concept evaluation [7] and better problem-solving [8]. Thus, the prevalence of the view that a team's efficiency is higher than that of individuals is not surprising.

However, findings of several empirical [9], [10] and computational [11] studies show there are scenarios in which teamwork may be inferior to the isolated work of designers. Possible explanations for such results include [10], [12]: social lofting [13], groupthink [14], incompatibility of team members' personalities, and task configuration; but more research is needed to understand such occurrences adequately. Similarly, numerous other aspects of design teamwork require additional studies. For example, researchers identified a need for future research on team decision-making [5], team learning [6], communication [15], leadership influence [16], and team behaviour in creative design activities [17].

Research on teams in design is hampered by the difficulty and cost of conducting empirical studies. McComb [12] emphasises how typical manner in which teams are examined – i.e.

experimental cognitive studies in which a group of designers collaboratively work on a task – requires a lengthy process of conceptualisation, data collection and analysis. As a result, months-, or even a year-, long cognitive studies on teams are a norm. This poses a significant difficulty in conducting and advancing design team research.

Another obstacle the real-world studies are faced with is the complexity and intangible nature of properties influencing human behaviour. Within the design field, a method frequently used to uncover designer's cognitive processes is protocol analysis, which offers an information processing perspective of the design process [18]. By recording observables such as speech [19], posture [20], gestures [21] or drawings [22], protocol studies enable capturing the designers' approaches to designing. Thus, protocol analysis has frequently been used to report information processing steps taken throughout the design process [18], [22]. Nevertheless, the interplay of processes underlying the observables remains challenging to understand. Questions such as: how do the cognitive and behavioural processes and affective mechanisms interact and result in the emergence of observed behaviours; how do individual's cognitive behaviour and actions affect others within the team; and how do the team properties, processes and behaviours emerge from actions of individuals, all require further research.

The problems of studying team cognition and other emergent team behaviours and processes are not unique to product development domain. In [23], the authors elaborated the importance of cognitive studies for advancing (general) research on teams stating:

"*While team research, […] has been fruitful, considerable work remains to be done. Several lines of investigation may be particularly beneficial […]. Foremost, perhaps, are the implications of cognitive theory, for team performance. One needs to understand how teams function as information-processing units; that is, how knowledge is acquired, shared and acted on. How do individual team members contribute vital pieces to the problem-solving puzzle, and how do those contributions build shared mental models and promote situational awareness? If one can understand this process, team performance measurement tools training can be shaped to capitalize on it. The next frontier in team measurement is to develop team assessment tools to capture cognitive phenomena. This is a must if progress is to be made understanding team functioning in complex systems."* (p. 1068-1069).

The cited thought emphasises the need for studies of team cognition. Such studies should be concerned with the emerging nature of team learning, shared mental model development and situation awareness formation. The work presented herein, thus, aims to tackle the problem of studying team learning by offering a computational tool which enables conducting multiple controlled experiments and exploring the sufficiency of implemented assumptions for the emergence of behaviours observed in the real world. In this manner, the developed research tool should be capable of offering a possible explanation of the processes underlying the observed team phenomena [24].

Computational modelling and simulations are frequently employed to study complex socio-technical systems (such as teams) since they provide researchers with a level of control unattainable in real-world studies. For example, one can easily re-run a simulation experiment (e.g. in identical setting) multiple times, or alter the input parameters and model assumptions one by one to observe the effect of such changes. Such flexibility in experiments promotes new insights, reveals the shortcomings of existing (implemented) theories, offers guidance of one's intuition and can lead to a definition of new research questions to be tackled by empirical studies [25]. The listed advantages are especially important for team studies where multiple dimensions (e.g. regarding cognitive, affective or motivational aspects) of human behaviour are intertwined in their influence on the overall team behaviour. Thus, a computational model which addresses various dimensions of teamwork and enables running numerous experiments aimed at studying their co-influences on the team performance could be of great value in advancing team studies. As further elaborated in the following section, the work presented herein aims at providing one such computational model, with a specific focus on enabling insights in team learning and adaptation.

1.2 Research aims and hypothesis

Following the presented motivation driving this work, one can conclude that the desired model of teamwork should enable cognitive studies of teams, supporting the simulation of various scenarios and capturing the emergence of different team properties. The desired model's implementation can be seen as a 'computational laboratory' serving for theory-testing and what-if analyses. It is important to note, however, that obtaining a highly-realistic, comprehensive model of designers' behaviour in a team setting is unattainable within the scope of this work. Instead, this work serves to set the ground for such a model by creating an easily-extendable, multi-purpose tool which integrates several of the existing theories and models.

The aim of the model is not to replace empirical studies, but – as emphasised previously - to provide a means for exploring the co-influences of multiple theories and studies of the sufficient conditions for the emergence of team properties and behaviours - in particular, team learning and adaptation. The results obtained by utilising the developed model should serve to guide intuition and inform future empirical studies. The results of such empirical studies should, in turn, be used to refine the model further and improve its veridicality.

Of specific interest within this study is a team's capability to learn and change its behaviour over time. To study how knowledge is acquired by and shared among team members, it is convenient to model each team member as a separate entity. Such setting enables studies of team learning and adaptation as emerging from the interactions among team members [26]. One simulation technique offering a means to implement the desired bottom-up perspective is agent-based modelling [27]. This work explores its suitability for the described research goal.

Thus, the goal and hypothesis of this work can be formulated as follows:

Research goal: The main goal of the doctoral research is to develop and validate a theoretical and computational model of teamwork in order to enable simulation and study of team's capability to adapt to changes in circumstances occurring during the product development activities.

Hypothesis: An agent-based model (ABM) is appropriate to study simulated emergent behaviour related to teams developing products, and to provide a framework for studying team adaptation as a response to changes in circumstances occurring during the product development activities.

1.3 Research methodology

The methodology followed for deriving the desired model of product development teamwork is developed after [28] and shown in Figure 1.1. Based on this methodology, the research can be divided into three research cycles: (i) *Relevance cycle*, (ii) *Development cycle* and (iii) *Rigour cycle*.

The first, *Relevance cycle*, is directed towards understanding the current state-of-the-art, defining the problem boundaries, describing the phenomenon of interest and exploring the existing solutions. Within this research phase, the computational modelling environment is linked to the circumstantial environment by specifying the research needs. These needs are stated in the form of requirements, constraints and specifications for a solid understanding of the product development teams and teamwork. Firstly, the phenomenon and variables of interest are identified and defined, along with the description of how to measure them reliably and validly. Secondly, the scientific principles and the nature of variables' relationships with the observed phenomenon (i.e. emergent team properties) in the context of product development are examined. Thirdly, the computational techniques (e.g. agent-based modelling) for simulation and exploration of the relationships between variables and the studied phenomenon are introduced and analysed. A review of relevant theories, existing models and empirical studies is conducted, and a set of preliminary models is developed to establish a valid basis for the development of a research framework focusing on team's emergent properties. As an outcome of this research phase, the primary variables and factors that are relevant for the analysis of team's emergent properties are identified, and a suitable computational modelling paradigm is selected.

In the *Development cycle*, findings from the previous phase are implemented in the form of a computational model. More precisely, this cycle begins with a definition and development of a conceptual model to be used for simulation of emergent team properties. Model elements are defined, and rules for their behaviour, evolution and interaction are explicated. To ease implementation, testing and subsequent experimentation, the derived model is separated into several sub-models of lower complexity. Sub-models aimed at representing specific behaviour or properties of importance for teamwork simulation are designed and implemented. Once these models are implemented, a set of testing experiments are designed and conducted. Iterative testing is employed to ensure internal validity and to enable refinement directed at an increase in consistency with the available theories and empirical findings.

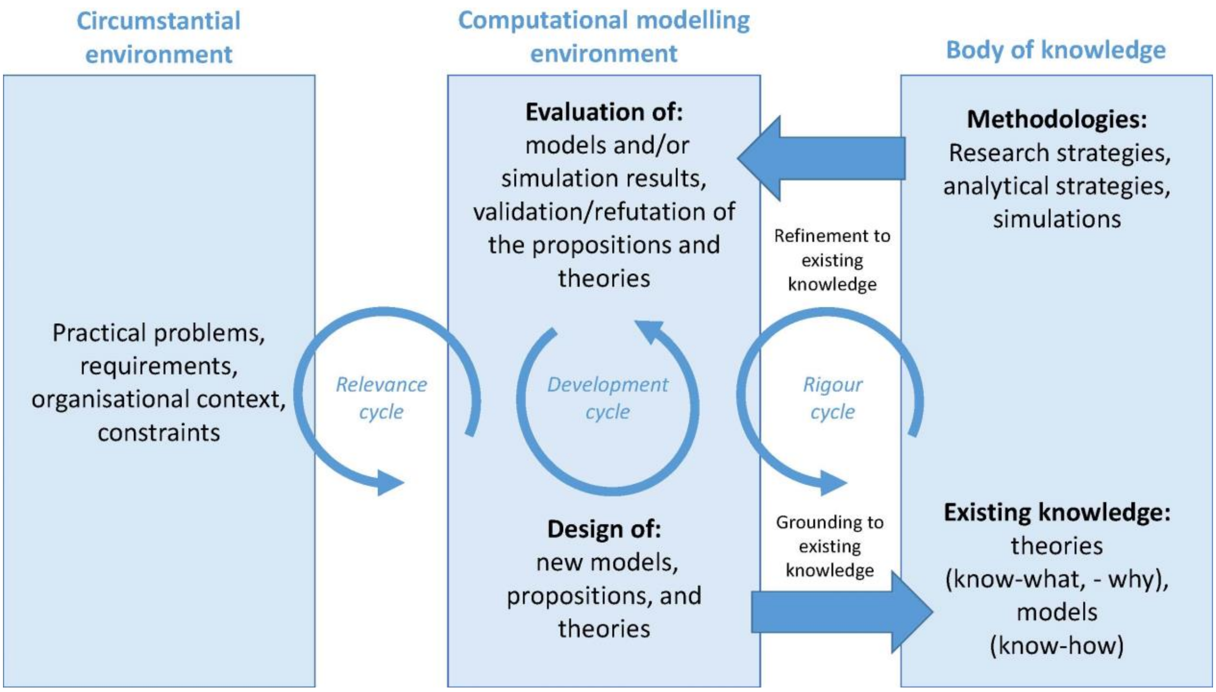


Figure 1.1 Research methodology (based on [28])

Finally, in the *Rigour cycle*, the models developed in the previous cycle are integrated into a coherent simulation framework. The developed computational system is tested to examine its capability to simulate product development teams and enable studies of emerging team properties. Findings based on a comparison of the outputs with the literature-based predictions and empirical studies (i.e. the body of knowledge) serve for further refinement and remodelling. In turn, once a sufficient level of system's credibility is achieved, the derived tool can be utilised for its purpose: testing of theories and informing the future team studies. In other words, the final research phase contains an iterative loop of theory testing and theory-building in which both, the developed model and the understanding of the real-world system are refined.

1.4 Scientific contribution

To obtain the emphasised research goal, multiple theories, empirical studies and existing models should be reviewed, integrated and build upon to derive a coherent computational and theoretical model. The review of the existing work and further integration, extension and testing activities lead to important outputs of the work presented herein. The expected scientific contribution of this work is manifested through:

1. A theoretical model of a product development team which consists of a model of an individual team member, a model of a team environment regarding tasks and resources, and a definition of mechanisms guiding the interactions among team members, and between team members and their environment.
2. A computational prototype in the form of a multi-agent system which is built based on the theoretical model and which enables the study of emergent team properties and behaviour.
3. A framework for calibration and validation of the developed model and computational prototype of team behaviour, with a focus on the component of the team adaptability and learning.

1.5 Dissertation outline

The thesis is divided into nine chapters, which are organised as follows.

Chapter 1 introduces the scope of the research by describing the motivation for the conducted work and by emphasising the research goal, hypothesis and contributions. The chapter outlines the research methodology followed to achieve the desired goal.

Chapter 2 serves to provide a brief research background in several domains relevant for this work. In the first part of the chapter, the terminology regarding teams is introduced, and several important aspects of team performance modelling are discussed. Then, the relevance and difficulties of team studies in product development setting are described. The second half of the chapter introduces computational techniques for modelling and simulation and discusses their suitability for the study at hand. Finally, several examples of agent-based models of teams are given to illustrate the range of modelling capabilities provided by this modelling technique.

Chapter 3 gives a detailed overview of existing agent-based models of product development teams and teamwork. First, the models' purpose and domain of application are briefly discussed (a more detail overview can be found in Appendix A), and the chapter then focuses on each of the main elements of agent-based models: agent, environment and interactions. The models are compared based on their representation of the product development tasks, as well as details in modelling the team members: their mental models, learning capabilities, expertise, personality, attitudes and emotions. Finally, the chapter explores how emerging team properties, behaviours and processes have been studied by the existing agent-based models. The detailed review of 40 models developed to date serves to highlight research gaps and opportunities, but also provides a basis upon which the model presented in this work is built.

Chapter 4 sets a theoretical background for the developed model. The literature was searched in order to derive the architecture of the multi-agent system, with particular attention given to determining the components of the agent's architecture. Further, the relevant research and theories describing each of the identified agent's components are reviewed to inform the model development. This chapter, coupled with the problems and existing approaches detailed in previous chapters, specify relevant primary variables and factors in modelling the emergent team properties and performance. Thus, this chapter concludes the work corresponding to the Relevance cycle of the methodology followed in the multi-agent system's development.

Chapter 5 presents the main contribution of this work: the developed multi-agent system for simulation of product development teams. Implementation details of each of the components identified in Chapter 4 are described, and connections among components are defined. Since this work focuses on providing a research tool capable of informing future cognitive studies of teams and their emerging properties, a strong emphasis is placed on modelling team members' cognitive behaviour and interactions with others.

The model introduced in Chapter 5 is extensively tested in a series of experiments whose results are reported in **Chapter 6.** This chapter aims to demonstrate the developed system's behaviour and to compare it to the prominent theories and empirical research findings. The conducted experiments, thus, highlight not just the model's capabilities, but also several of the model's limitations and opportunities for future refinements. The work reported in this chapter corresponds to the Development cycle of the employed research methodology, and (along with the work outlined in previous chapters) concludes the development of the theoretical model and computational prototype specified as first two scientific contributions of this work.

Chapter 6 describes the results of hierarchical testing of each of the modelled components or sub-models directed at simulating specific aspects of teamwork. As described in Section 1.3, after completion of the Development cycle, the sub-models are integrated into a coherent simulation framework. In particular, the developed multi-agent system is tested for the capability to capture the change in the team's behaviour due to learning (i.e. team adaptation). **Chapter 7** outlines the results of such experiments, thus reporting on the outcomes of the research phase corresponding to the Rigour cycle. The work presented in Chapters 6 and 7 results in the third scientific contribution specified in Section 1.4.

One example of the experiment conducted with the developed multi-agent system is presented in **Chapter 8.** The experiment results demonstrate the capability of the developed model to inform the cognitive studies of product development teams regarding the team search for the solution (i.e. design solution space exploration and expansion) and its impact on the solution creativity assessment.

Chapter 9 concludes this work by reflecting on the research goal, hypothesis and scientific contributions. The developed model is compared to the existing agent-based models of product development teams (reviewed in Chapter 3), which provides a frame for discussion of the model's limitations and advantages. Finally, following the description of the Rigour cycle of the employed methodology, the model's future refinements and opportunities for future work are discussed.

2 RESEARCH BACKGROUND

This chapter presents a brief research background and explores the suitability of modelling techniques to represent aspects of teamwork relevant to the research goal specified in the first chapter. The chapter commences with the introduction of important concepts regarding teams in general and then describes the setting in which product development teams are operating. Next, computational techniques for modelling and simulation of complex systems are introduced and briefly reviewed with the regards to their capability of capturing dynamics of a team. Finally, a closer look into several agent-based models of teamwork is taken to present the capacity and potential of the agent-based modelling technique. The chapter is concluded with a short discussion on the suitability of modelling techniques for the problem at hand. The work presented in this chapter builds on the work published in [29].

2.1 Teams, teamwork and team performance modelling

Numerous definitions of the term *team* can be found in the literature. For example, while some regard dyads as a team (e.g. Salas et al. [30]), others argue that a minimum of three individuals is necessary to form a team [31]. The authors in [31], thus, reviewed several perspectives and integrated them in the following definition (p. 79):

> *"[A team is a group of] (a) two or more individuals; (b) who interact socially (often face-to-face or, increasingly, virtually); (c) possess one or more common goals; (d) are brought together to perform organizationally relevant tasks; (e) exhibit interdependencies with respect to workflow, goals, and outcomes; (f) have different roles and responsibilities; (g) and are together embedded in an encompassing organisational system, with boundaries and linkages to the broader context and task environment.".*

In particular, specialised roles, the interdependence of tasks, and shared goals are crucial features distinguishing teams from small groups [23]. But Salas et al. [32] argue that even more is needed to form an effective team. The authors stress the importance of effective coordination, cooperation and communication among members to develop a shared understanding of a task and a team. In short, the authors emphasise *teamwork* as a necessary ingredient for a group to form an effective team. Motivated by the question of the essence of teamwork, the authors in [32] conducted an extensive literature review and identified five core components promoting team effectiveness and influencing team performance. The *'Big Five of teamwork'* are team leadership, mutual performance monitoring, back-up behaviour, team orientation, and adaptability. In order to enable, update and support the development of Big Five factors, the coordinating mechanisms are needed. These coordinating mechanisms are shared mental models, mutual trust and closed-loop communication. Although additional variables are found to influence teamwork, the authors in [32] argue that the ‘Big Five’ represent the core teamwork components.

In accordance with the stance taken by Salas et al. [32], Kozlowski et al. [33] note that the notion of teamwork is captured by *team processes*. In their work, Marks et al. [34] define team processes as interdependent acts of team members that perform cognitive, verbal, and behavioural activities aimed at organising taskwork to lead towards collective goals achievement. This definition clearly distinguishes taskwork (i.e. interaction with tools, machines and systems [34]) from team processes, for which interaction with other members is a critical feature.

Another essential concept distinct from but related to team processes is *emergence*. Emerging team states arise from interactions among team members, depend on the context and team experiences, and are dynamic in nature. In the context of team processes, emergent states can be considered as both, input and proximal output [34], since they arise from but also impact team processes. Kozlowski et al. [33], [35] note that emergent phenomena (states and processes) result from the interplay among individual cognition, affect, motivation, and behaviour, are amplified by interactions among individuals and manifest at a higher level – as a collective phenomenon. As the most critical emerging states and processes for the team effectiveness, Kozlowski and Chao [26] emphasise *team cohesion* and *team cognition*.

Team cohesion is defined as the commitment of team members to their team and their desire to maintain team membership ([31], [36]), and is closely associated to one of the Big Five factors – team orientation. However, Salas et al. [32] note that team orientation is more general: while team orientation deals with a general tendency to working with others, team cohesion is directed towards a specific team.

Team cognition encompasses both: the process of *learning* and *knowledge outcomes*. Although the distinction between the two is often not clear [26], distinguishing them is especially important in cases where the goal is to examine the knowledge emergence from learning in teams [37]. The authors [26] describe how learning begins by one’s acquiring internalised knowledge (i.e. individual learning). The knowledge gets externalised by communication, information sharing and exchange of ideas. Thus, interactions are necessary for the shared knowledge to emerge from distributed knowledge. Outcomes of learning are the formation of collective knowledge, shared mental models, transactive memory and macro-cognition [37]. Such shared cognition supports the development of “Big Five” of teamwork, thus influencing team effectiveness and performance [32].

When modelling teams and team performance, one may utilise work presented in [38]. With the aim of providing a framework for team performance modelling and simulations, Salas et al. [38] extract key ‘ingredients’ to consider when developing a teamwork model. Identified aspects of teamwork are further classified based on necessity of their inclusion in a model of team performance. Thus, three level of importance are defined: ‘must-be-modelled’, ‘should-be-modelled’, and ‘would-like-to-model’.

The ‘*must-be-modelled’* category comprises team components needed to sufficiently present the task, team and their environment. In particular, the authors argue that individuals must be characterised by their cognitive ability and personality, and their work assignment and communication structure must be explicated. Similarly, task load and time pressure need to be included to represent environmental factors, while task type and interdependencies among tasks must be specified to sufficiently describe tasks team members are performing. To characterise a team, Salas et al. [38] emphasise modelling of team type, structure and size. Finally, the authors note the importance of modelling noise and its influence on team member’s interactions and information processing.

The *‘should-be-modelled’* components focus on representing diversity among team members. Thus, the authors [38] note that details on the team member’s expertise area and cultural factors should be added to the model. Additionally, an individual’s behaviour should be detailed through modelling of mental models, motivation and attitudes. Salas et al. [38] also discuss some of the influences among several of the included elements (e.g. the impact of team size and task difficulty on an individual’s motivation).

As proposed in [32], team effectiveness depends on the Big Five elements of teamwork and the coordinating mechanisms. Modelling of these team competencies (i.e. team leadership, mutual performance monitoring, back-up behaviour, adaptability, team orientation, shared mental models, closed-loop communication and mutual trust) necessarily depends on knowledge, skills and attitudes of the individuals forming a team. Similarly – as already discussed - team cohesion emerges from the interactions among team members. The authors [38] add the notion of team climate: another emerging state reflecting the ‘ambient stimulus’. These emerging states and processes form a *‘would-like-to-modelled’* category. Finally, the authors add team norms to this category.

Altogether, these categories define a set of important components to be considered when developing a team performance model [38].

2.2 Teamwork in product development

Many of the design tasks are conducted by teams. In fact, goal formulation, ideation and decision-making are all typically conducted in a team setting [39]. Following the paradigm shift from individual to team-based work organisation described in the first chapter, it is not surprising that the research focus changed along. Ensici et al. [5] thus note how in the 1960 the focal research question changed from ‘How does a designer design?’ to ‘How do designers design as a team?’. However, despite this increase in research interest, numerous areas of design teamwork remain insufficiently understood.

One reason for slow progress in understanding (design) teamwork lays in its complexity. A design team is formed of individuals who - aside from interacting with one another - interact with resources in order to produce a description of a product satisfying the requirements. Additionally, the environment in which design teams are situated is dynamic, with market demands and availability of resources constantly influencing the team and their goal. Thus, while designing, the design team, non-human resources and their work environment operate as a socio-technical system, as presented in Figure 2.1 [29].

The socio-technical system comprised of team members (human), non-human resources and the environment (represented in Figure 2.1) displays dynamic, non-linear, emergent and self-organising behaviour. This behaviour arises from the interdependence of the system’s parts and is difficult to determine by observing the parts in isolation. In other words, the presented system possesses all of the characteristics of complex systems [40], [41].

The complexity is amplified by several characteristics of design teams – in particular, cross-functionality [3]. Design teams are usually composed of individuals with diverse knowledge background which - although beneficial in promoting creativity and tackling the variety of product requirements – also increases the level of conflict and misunderstandings among team members [3]. Thus, it is not surprising that research on team cognition and development of shared team mental models [42] and effective communication and cooperation [43] are regarded as of great importance within the design field.

Additionally, design teams often have fluid team boundaries and temporary membership [3], which emphasises the need for understanding how team expertise develops, persists and evolves in design teams [44]. Frequent changes within the team, as well as changes in the environment (e.g. new technologies or task requirements) render another component of the "Big Five" as crucial for the design team success: adaptability.

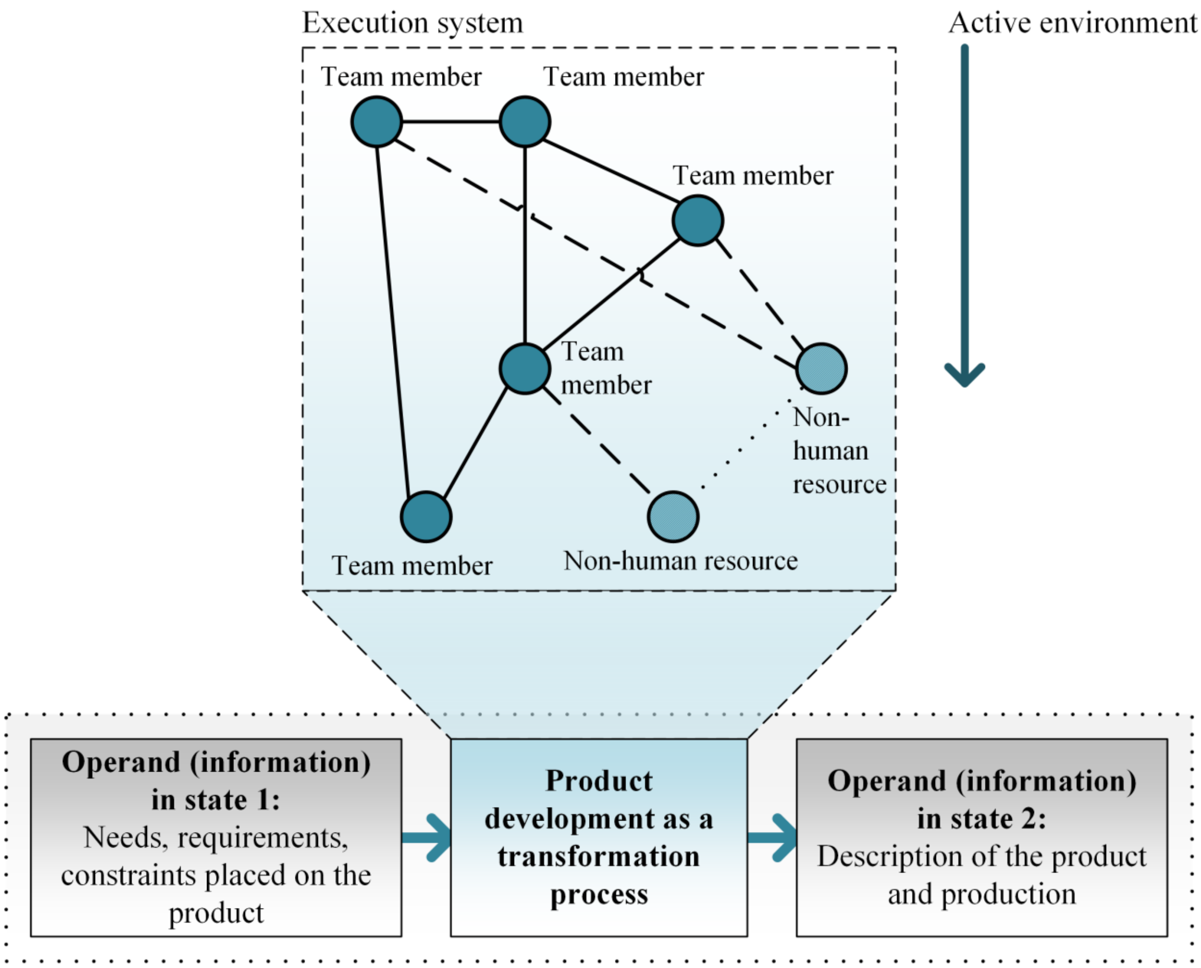


Figure 2.1 Model of design process as a socio-technical system (developed after [45])

Similarly, one can identify numerous other areas of design teamwork that require additional research. For example, researchers (e.g. [15]) discussed the potential relation of team cohesion and design team effectiveness, while [46] explored the emergence of team creativity with regard to team composition, coordination and management processes. Every of the listed 'Big five' aspects, the coordinating mechanisms, and other emergent team states and processes presents a potential domain of future work. A manner in which such studies may be supported is by developing *simulation models.* A *model* is an approximation of the system – a simplified portrayal of reality whose manipulation and analysis helps in gaining a deeper understanding of the problem.

2.3 Computational modelling and simulations

Building on the conceptualisation of a design team and its environment as a complex socio-technical system (Section 2.2), when deciding on the most suitable technique to obtain a teamwork simulation model, common techniques for modelling and simulation of complex systems are considered.

Some of the well-known such techniques are, for example, Petri Nets, Cellular Automata, Discrete Event, Markov Chain, System Dynamics and Agent-based modelling. However, the exhaustive list is difficult to provide since the line between these approaches is not strictly defined [40], and many models utilise aspects of several techniques. Nevertheless, the author in [40], argues that Network, System Dynamics, Discrete Event and Agent-based modelling techniques have the ability to satisfy the widest span of modelling needs.

Network Simulation (NS) – as evident from its name – represents a system of interest in the form of a network, i.e. graph where vertices represent entities, and relationships between entities are represented by edges. Relationships can have different weight, can be directed or of various types. Erdös and Rényi [47] first used networks to represent and study complex systems, laying the groundwork for numerous later applications [48]. Other important research results on networks can be found in the work of Watts and Strogatz [49] and Albert and Barabási [50]. NS is an approach often used to simulate communication networks, social proximity or exchange of data packages over the Internet.

System Dynamics (SD) was developed by J. Wright Forrester in 1950s [51]. It is a top-down approach that models continuous changes within a system. Changes are driven by cause-effect dependencies that can be represented mathematically as a system of differential equations. Thus, System Dynamics approach models high-level aggregates: it uses level variables to describe the systems' state [52] and represents modelled elements at an aggregate level. Since individual entities are not modelled, interactions and individual characteristics are difficult to represent.

Discrete Event Simulation (DES) was introduced by G. Gordon in the 1960s [53] and serves for simulating a sequence of events, i.e. discrete points in time when entity's state variables change. Various discrete-event worldviews such as event scheduling, process interaction, activity scanning, state machines and others are listed in [54]. However, two most common are event-oriented and process-oriented [55]. In the process-oriented worldview entities simply move through processes that require some resources and a certain amount of time. Rules are implemented on a system level, and an emphasis in the simulation is placed on entities' life cycle. In the event-oriented worldview, modeller concentrates on events and their impact on the state of the simulated system. Usually, Discrete-Event modelling is used to explore problems of bottlenecks and queues in modelled system.

Agent-based Simulation (ABS) is a bottom-up approach where each entity, i.e. agent, is autonomous and has unique characteristics and variable values. A typical agent-based model consists of a set of agents, an environment and a set of interaction rules guiding agents' interaction with others and the environment. First agent-based models date to 1970's (for example, Schelling's model of ethnic segregation [56]), but agent-based modelling was not in wide use until 1990's. By modelling each agent individually and defining rules for their interaction (with other agents and with the environment they are situated in) and observing the outcomes of the simulation, researchers can explore the macro-level behaviour emerging from agents' interactions.

The brief overview of several aspects underlying effective teamwork (Section 2.1), and the described setting in which design teams are operating emphasise the criteria for evaluating the available computational techniques. The goal of enabling cognitive studies of design teams poses a requirement of creating artificial representations of designers' cognitive behaviour. Similarly, the goal of studying team properties and processes as emerging phenomenon, stresses the need of capturing interactions among members. Finally, dynamic, non-linear and adaptive behaviours of design teams should be modelled.

Comparison of modelling techniques with respect to the relevant criteria is presented in Figure 2.2 [29]. The comparison (discussed in [40]) emphasises the superiority of agent-based modelling regarding modelling flexibility. However, such expressiveness comes at the expense of difficult creation and validation. Nevertheless, the analysis indicates the suitability of ABS technique for developing the desired model of a design team.

This is not to say that NS, DES and SD models are not useful in studying design team performance – rather that ABS is the most flexible in capturing aspects relevant for cognitive studies of design teams. Other techniques are, on the other hand, often preferred to ABS for simulating design processes. As argued in [57], SD models are particularly useful in capturing iteration and concurrency of the design processes, while - for example – work in [58] enables simulation of distributed product development processes for small and medium enterprises by utilising DES modelling technique.

Evaluation criteria / Method	Network Models	Discrete Event Models	System Dynamics Models	Agent-based Models
Nonlinearity – ability to model disproportionate causes-to-effects	Good	Very Good	Very Good	Excellent
Interactions – ability to model the effect of interdependencies between entities	Very Good	Good	Poor	Very Good
Intelligent Agents – ability to model sentience of the entities	Very Poor	Poor	Poor	Excellent
Represent Hierarchies – ability to represent the organisational hierarchy of the entities	Very Good	Good	Good	Excellent
Emergent Behaviour – ability to provide insight into the macroscopic behaviours that cannot be elucidated from the analysis of individual entities in isolation	Good	Poor	Poor	Excellent
Adaptation – ability to model capability to change of the individual entities, closely related to the ability to model sentience	Very Poor	Poor	Poor	Excellent
Dynamic Behaviour – ability to model time-dependent effects and changes of state	Poor	Very Good	Very Good	Very Good
Ease of Creation – how much time and effort must be devoted to developing the model	Excellent	Poor	Good	Very Poor
Ease of Validation – how much time and effort must be expanded in validating the model	Very Good	Good	Good	Very Poor

Figure 2.2 Comparison of techniques for modelling and simulation (after [40])

2.4 Agent-based modelling of teams

In their discussion on the suitability of agent-based models to represent organisations and teams, the authors in [59], [60] emphasise that, due to its bottom-up approach, agent-based models do not have to be based on well-structured formal theories. This creates an advantage for management, organisational studies and psychology as these domains rarely provide formal equations. Researchers [59] argue that by shifting the focus from solution finding (taken in equation-based techniques) to exploring dynamics and evolution of the system, agent-based modelling approach provides a powerful tool for studies of emergence within complex systems [61].

One example of such a research tool is provided in [62]. Dionne et al. [62] build on McComb's theory of three-phase mental model development [63] to simulate shared mental model convergence in a team. The model is further used to study the influence of leadership and shared mental models on team performance. By modelling differences in agents' view of the team goal, the inclusion of agents' beliefs of others' expertise, and a variable of self-confidence, the authors capture many interesting phenomena in a computational model.

Hughes et al. [64] emphasise advantages of agent-based modelling in the form of its capability to integrate multiple theories and a plurality of data sources at the level of the agent's interaction and behaviour rules. While other approaches also enable multi-source data integration, the authors argue that ABM offers a different level of granularity which results in broader and more interdisciplinary models than the model commonly employed in organisational psychology domain. The capability to integrate various aspects of – for example - human behaviour enables the creation of increasingly veridical systems.

One example of the models where multiple theories are integrated to derive a detail social behaviour is Construct [65], [66]. Construct uses social network analysis for detecting risks in organisations and for simulating social change emerging from communication and learning of agents forming a team. It models the agent's beliefs, information exchange, learning, and an effect of change in the agent's beliefs and knowledge on activities. Agents are creating beliefs about not just each other's expertise, but also their similarity. This enables Construct agents to display homophily. The dynamic relationships among Construct agents are derived from organisation theory, structuration theory and influence theory, and give rise to complex temporal behaviour.

It is important to note, however, that a call for highly veridical agents does not diminish the value of agent-based models in which agents are modelled as simple information-processing units.

An example of the model where simple behavioural rules lead to significant findings is the model of Hong and Page [67]. Hong and Page [67] modelled groups of problem-solvers as agents with different problem-solving heuristics. The problem space is represented as a plot of a random function where the height represents the quality of the solution. Based on their heuristics, agents are roaming the space in search of the function's maximum. Once an agent determines it cannot find any higher value position (i.e. it is stuck in a local optimum), it shares its findings with the others, and the next team member starts its search where the previous member finished. By using this model, authors proved that a randomly selected subgroup of problem-solvers with diverse search heuristics could outperform the group of problem-solvers who individually scored the best results (who themselves can be diverse) if the initial population of problem-solvers is sufficiently large. The importance of diversity in teams is also emphasised by LiCalzi and Surucu [68], who developed a model equivalent to Hong and Page's model but, while Hong and Page studied large problem-solving populations, LiCalzi and Surucu studied large problem spaces [68]. None of the agents as implemented in [67] and [68] learns (in terms of increase of skills or expertise), changes motivational states, or dynamically forms social relations with others. Agents and environment in these models are rather simple - agents behave following few simple rules and interactions between agents are limited to transfer and evaluation of current solution. Nor agents' characteristics, nor rules for their interactions, nor the environment changes during the simulation, and agents possess no model of their team members' knowledge or capability. However, such simple definition enables the simulation of large populations of agents, facilitates the result analysis and code verification, provides means to simulate a great variety of different scenarios in a reasonable amount of time, and enhances our understanding of the processes which can further be enriched and extended afterwards.

2.5 Research implications

Teams are complex systems, and much remains to be explored and understood about their functioning and dynamics. Team performance and processes cannot be predicted by observing individual members in isolation. Thus, when developing tools for team studies, ones capable of capturing emergent team states and processes are of particular value.

Macal [69] argues that any system in which entities are autonomous, heterogeneous, strategic, capable of learning, are anticipating other entity's reactions, have dynamic relationships and form organisations, makes a good candidate for an agent-based model. Teams, regarded as a system of interacting individuals embedded in an organisational setting, have all of the listed characteristics. Team members, their interactions and organisational context can naturally be translated to the elements of an agent-based model: agents, interactions and environment. The bottom-up approach taken in agent-based modelling enables exploring dynamics and evolution of the system, thus facilitating studies of emergence.

Existing agent-based models integrate findings from several theories and explore team processes difficult to capture in the real-world setting. Simulations obtained by utilising these models increase the understanding of the mechanisms underlying efficient teamwork and provide a stepping stone for future, more detailed and veridical models.

Additionally, one feature of the body of research on design may prove beneficial in developing agent-based models of design teams. Early design studies focused on the works of individuals [5], and to this date, studies of individual are prevalent (to design team-oriented studies) [17], [18]. Thus, the majority of what is known about the design process fits the level of an individual [18]. Although one must be careful to not generalise the individual-level findings to a group setting, some of the developed theories and collected data can be used to guide the detailing of the individual agent's behaviour. Then, simulations and subsequent empirical studies could test if such models of interacting individuals enable the emergence of the observed team properties and processes.

3 RELATED WORK: AGENT-BASED MODELS OF PRODUCT DEVELOPMENT TEAMS

This chapter takes a closer look at the existing agent-based models of product development teams. The models included in this chapter were obtained by searching the Web of Science, Scopus, ACM DL, and IEEE databases using the query:

> *agent* AND (model* OR simulat*) AND (team* OR group*) AND ("product development" OR "product design" OR "engineering design"),*

or its equivalent for each database. The results were further filtered to include only the articles written in English. Articles where a) agents were used as a human avatars intended to display realistic behaviour, b) the model is used to study aspects of product development team behaviour or performance, and c) sufficient details were provided to gain understanding of the model (i.e. agents, environment and interactions are described, although the implementation may not be performed) were included. However, several exceptions to these rules have been made. Models ([70]–[72]) were included because they explicitly model the behaviour of individual designers with a similar goal, though these designers do not necessarily belong to the same team. Searching the citations and references further extended the set of identified articles. Overall, 40 different models were detected. Their application area, main purpose, key characteristics and limitations are reported in Appendix A. Additional models, for example, Team-RUP ([73]) or model developed in [74], although not included in this overview, may serve as a source of implementation ideas. Team-RUP [73] is used for simulation of cooperative team behaviour in software development teams. Particularly, it is aimed at simulating the effects of turbulence (i.e. change of the requirements and employee turnover) on the team efficiency and effectiveness – thus tackling the problem of studying team adaptation. However, the agents within Team-RUP model represent teams (rather than individuals), thus restricting the study of influence individuals' traits have on team performance, and preventing studies of team adaptability as a property emerging from individual-level adaptation and interactions between team members. Similarly, the model in [74] presents an interesting study of the effect of initial mental characterisations (regarding inter- and intra- subsystems' interactions) on product development performance. Although this model introduces useful ideas on how to represent a design space and simulate agent's mental model update through social learning and imitation, like in Team-RUP, the simulated agents serve to represent teams rather than individuals.

In this chapter, a short review of the identified models is presented. The discussion is organised to examine the differences in models' purpose and trends related to the three elements of agent-based models: agent, environment and interactions' representations. The overview does not aim to cover every aspect of each model, but to provide several observations on the commonalities and differences among reviewed models. The work presented in this chapter builds on and extends the overview published in [29].

3.1 Model's purpose and domain of application

The identified models differ greatly in their scope and purpose. Majority of the models are intended to model particular team properties, behaviours or processes, and enable studies of their impact on the team. For example, the model described in [75] explores how the use of analogies influences team cohesion and collaboration, [76] deals with the formation of transactive memory system, [77] studies the impact of work overload on team innovation, and [78] tackles the question of the effect of team structures on brainstorming outcomes. Based on the classification provided in [79], these models are *generators* as they explore whether a theory can produce a behaviour observed in the real world, and serve to increase understanding of little-understood aspects of design teamwork. The prevalence of such models is not surprising: the authors in [79] note that, in complex fields without firmly established theories, the majority of developed models tend to be generators.

Rather than providing theory-testing research tools, other models (e.g. [57], [80]–[86]) are primarily concerned with the estimations of project duration, cost and quality of outcomes, or anticipating problems in project execution. Most often, these models are developed as managerial tools which aim to support team formation and process planning. One of the first and best-known such models is Virtual Design Team (VDT) model [80], [87], [88], which was later commercialised as SimVision®, licensed by ePM [89].

Most of the reviewed models simulate 'generic' product development teams and do not pose limitations on the domain of application. However, models [90]–[93], simulate mass and open collaborative product development (MCPD), thus introducing specific assumptions on the simulated processes and agents' behaviour. In recent years, MCPD has emerged as a new form of product development where large communities of loosely connected individuals work together to produce goods or services [90]. In contrast to traditional product development, participants of MCPD are self-interested, decentralised, loosely connected, and can choose when to work and what tasks to perform. There are no deadlines or tightly controlled processes, and product continuously evolves over time guided only by the interests of producers. One example of such a product is Wikipedia. Due to the differences from the traditional setting, results obtained by utilising MCPD team models often cannot be generalised to traditional product development teams.

The models [94] and [95] simulate space mission design teams. In [95], Bellamine-Ben Saoud and Mark are modelling a design session during which sidebar discussions are occurring and interrupting the workers. Sidebars are creating noise, but also act as a source of information, and the authors explore how sidebar conversations affect error detection and team performance. Olson et al. [94] developed an agent-based simulator which is used to simulate behaviour observed in the design group at NASA's Jet Propulsion Laboratory called Team X. Although these models can be used to simulate other design teams, their usage is hampered by the amount of input data necessary to tailor the simulations for other use cases.

The work in [96]–[98] forms another group of models which simulate teams developing a specific class of products: software. Blau et al. model [96] studies how incentive schemes, individual or group, in large-scale lean software development affect the teams' backlog dependency resolution behaviour. Farhangian et al. [97] model team member's competency and personality and study their impact on team performance by comparing task allocation strategies: minimising over-competency or minimising under-competency. Finally, the model developed by Xia et al. [99] is used to explore the effects task assignment strategies on the evolution of knowledge in software R&D teams.

Within construction and management engineering (CEM) domain, models [100], [101] and [102] have been proposed. However, due to a coarse-grained view of teamwork and project execution taken, these models are applicable to broader fields of research. Additionally, the problems these models are tackling are relevant to other product development domains. The model presented in [102] captures interdependencies and functional diversity among members and compares team member selection orientations in different (market) contexts. Building on the VDT [80], VOICE [100] enables modelling of details of the project workflow and organisational structure to estimate performance outcomes regarding project time, cost and quality. Son et al. [101], [103] develop a simple, network-based model of collaboration to study how interaction cost influences the evolution of temporary project teams.

The listed examples emphasise another dimension along which the reviewed models differ: the time frame and project phase simulated. While most of the models take one task or a project as standard simulation length (i.e. one simulation run represents one task or a project), others are specifically designed to simulate team dynamics over long periods. For example, [82] and [78] simulate a brainstorming session (thus simulating no more than several work-hours). On the other hand, in Hsu et al. [102] model, one simulation step represents one day, and a typical simulation run represents multiple years period. Similarly, the model in [103] simulates the evolution of collaboration networks over multiple projects. Regarding the design phase simulated, several models (e.g. [104], [105]) explicitly map different design phases to the modelled aspects. The model presented in [94] simulates conceptual design of a space mission plan, while [57] and [106] are targeted towards capturing the dynamics of early design processes in industrial organisations.

3.2 Representation of a task and an environment

Many of the developed models conceptualise design as a search over a rugged landscape. In these models, each point on the landscape represents one solution, and the height of the landscape determines its suitability for the problem. Thus, the agents roaming the space are - in essence - acting as optimisers in search of a global maxima.

One popular approach to creating a rugged landscape that represents a design space is by utilising Kauffman's NK model [107] In the NK model, *N* is a parameter representing the number of individual elements influencing the height of the landscape (i.e. the overall fitness value). Each element typically has two possible states: 0 or 1, thus yielding 2^N possible combinations. For every element, a fitness value is assigned to each of its states. Further, the model defines interconnections among the elements. Namely, each element (out of *N* elements) is related to *K* other elements. The contribution of one element to the overall fitness depends on the state of the element and states of each of its K related elements. If K equals to zero, the element's contribution to the overall fitness depends solely on the element's state (i.e. equals the fitness value assigned to that element-state pair). On the other extreme, if K equals N-1, dependencies form a fully connected graph. In other words, the element's contribution to the overall fitness depends on simultaneous states of every element of the system. Thus, as the K increases, the variance of the overall fitness values (i.e. ruggedness) also increases: maxima become higher but more difficult to find [70], [107].

In the design context, the NK model can be used to model design performance. For example, in [70] and [108], parameter N represents the number of design components (or decisions to be made). At the same time, K determines the level of interconnectedness of the components (or decisions). In [104], the author uses the NK model to represent a level at which possible solutions meet the requirements. Parameter N represents the number of requirements, while K guides the level of interdependencies among the requirements. Hsu et al. [102] use the NK model to represent the interdependences among individual team members' contributions to the overall design: there are N team members and the performance of each depends on the performances of K other team members. Finally, Jafari Songhori et al. [109] develop a modified NK model in which N represents the overall number of elements. These elements are divided into K subgroups (subsystems) of equal size. New parameter C was introduced to capture the interdependencies among subsystems. Thus, the model in [109] represents a system of interconnected subsystems. Teams of designers develop subsystems: one team develops one subsystem. Elements of each subsystem are themselves interconnected to capture intra-team dependencies.

While the NK model provides a simple way of generating theoretical design landscapes that include dependencies among components of the system, several of its limitations can be emphasised. For example, NK models assume constrained (albeit large) design spaces: there is a finite number of solutions (i.e. designs) which can be generated. The created landscape is static – fitness values do not change throughout the simulation, and N and K remain the same indicating that the number of components, requirements or team members (depending on the application) is fixed and that their interdependencies do not change. Additionally, by applying the fitness functions as in the NK model, an assumption about the availability of an objective performance measure is made. Ambler [104] notes that a rugged landscape usually represents only the physical nature of the underlying product – thus omitting market, economic and legislative aspects from the performance evaluation.

Mihm et al. [110] also note that in product development is not common that a small parameter shift (i.e. small shift in one component's performance) results in a radical change of the overall performance – which the NK model (with high values of K parameter) would suggest. Instead, the authors in [110] develop smooth performance functions arguing that correlated performance for neighbouring points suits the product development context. Additionally, rather than choosing one of the discrete states of their components, the agents in Mihm et al.'s model [110] move by following local smooth functions with a single optimum. Models in [106], [111]–[113] also develop agents' environment as a rugged landscape determining the team performance. While these solution spaces are continuous, much of the problems (e.g. assumption of the existence of an objective performance function and static landscape) present in the NK model apply to these landscapes as well.

All of the listed models that conceptualise design as a search problem assume the existence of an optimum or optimums. Although one can set up such models in a way that the optimum is unreachable to agents (albeit, known to the modeller), the model in [78] develops a fundamentally different solution space: the one in which agents can always generate an "even better" design. The agents in Sosa and Gero's [78] model are searching a space of all designs which can be generated by overlaying (and intersecting) geometrical shapes within a plane. The goal function is defined based on the number of shapes emerging from such overlays, and the number of sides of resulting shapes. This function is unbounded and defines a rugged landscape (a change in input can generate a significant change in performance). A reason for such function choice lays in the purpose of the model: while previously mentioned models are capturing design's technical performance, the model in [78] strives to capture the dynamics of design's creativity.

Rather than modelling team performance through the assessment of a design, some models focus on details of a product development project's workflow. Typically, in such models agents are presented with a set of tasks which they need to process (i.e. spend sufficient time working on each). Collaboration and communication among agents within such models is guided by prespecified task dependencies and exception-handling protocols. These models have the capability of estimating the project duration, problems in execution regarding iterations and amount of rework, anticipating communication issues or work overload. Thus – as described in the previous section – they provide managerial insights into the project performance.

In most of these models (e.g. [57], [80], [81], [83], [85], [114]–[116]), task characteristics, dependencies among tasks, as well as task assignment to agents are predefined and taken as a simulation input. As such, many details regarding project workflow have to be known in advance and passed to the simulation. For example, VDT [80] utilises Quality Function Deployment (QFD) [117] and Design Structure Matrix (DSM) [118] to transform data on the product, process and organisation into task dependencies and characteristics regarding complexity and uncertainty (details can be found in [119]). DSM has also been employed in models such as [102], [116] and [90] to represent the tasks based on interdependencies among product's functions ([116]), product's modules ([90]), or among workers ([102]).

Aside from task uncertainty and complexity, in order to set up the simulation in VDT, one has to specify knowledge or skill requirements and typical duration of each task. Describing tasks regarding dependencies, knowledge requirements, duration and/or difficulty has also been employed in models such as [83], [85], [97], [99], [116] and [115]. These task characteristics enable determining the order and approximate duration of tasks, and uncertainty, knowledge requirements and difficulty (or complexity) serve as a basis for the calculation of the probability of rework and failures. One can also specify the priority of each task (e.g. [83], [85], [97]), thus imposing an arrangement among independent tasks. Additionally, in [81], the author models a rate at which new information is produced during the task performance, while [81], [92], [97] include the number of participants and collaboration time for each task. Depending on the model's purpose, one may add multiple specific parameters characterising tasks. For example, the model in [97] introduces a parameter indicating the required creativity level of each task and estimates how different team compositions cope with such requirements.

Somewhat similar to the approach taken in discrete-event simulation models, the models simulating design team's workflow often aim to simulate rework and iterations due to conflicts in resource allocation and failures due to inadequate work quality. Some models capture rework and iterations by simply integrating a parameter guiding the probability of task failure and need for modification (e.g. [57], [92]). In other models (e.g. [80], [114], [116]), reworks are needed when there is a mismatch between agent's knowledge and task requirements, which causes the insufficiencies in work quality. However, current models rarely capture iterations introduced by changes in task requirements. Only two of the models take into account such effects: [57] and [97].

Specifying the project workflow (i.e. tasks, task duration, requirements and dependencies), and assigning each task to an agent requires a significant amount of input data. Several models thus employ different task delegation and task scheduling mechanisms to reduce the amount of input needed to set up the simulation. To simulate task delegation, models utilise manager agents (i.e. agents simulating managers). Different managerial strategies are implemented, and the simulation results serve as estimators of their suitability for the problem of interest. For example, in [99], tasks are dynamically and randomly created (following a Poisson process). The manager agent assigns tasks to idle agents following one of the strategies: choose the most knowledgeable agent, the one with 'just sufficient' knowledge, the one whose knowledge is just below the requirement (so they will learn by doing), or randomly. In VOICE [114], on the other hand, the manager optimises either time, quality or team's work pressure. Aside from modelling a manager agent which delegates tasks and handles errors in project execution, one can opt to implement specific task scheduling mechanisms in agents representing team members. In this manner, the design agents can re-arrange their tasks and avoid resource or collaboration conflicts. Work reported in [120] presents one algorithm for task scheduling which takes into account task's urgency, importance, recovery cost and the agent's preferences.

As already emphasised, the models detailing a design project's workflow usually focus on estimating timeliness and achieved quality for specific work arrangements. VDT [80], for example, determines the amount of rework and coordination in the overall work volume. Cost and time efficiency are estimated by calculating the ratio of planned and realised work volume/production time. Similarly, coordination quality is determined from the number of attended request for coordination, while verification quality is based on the percentage of fixed failures (of the overall number of detected failures). The model in [114] adds additional performance measure: it estimates work-related pressure.

The described groups of models, 'landscape search-based' (where a product is approximated) and 'workflow processing-based' (where a process is approximated), are neither exhaustive nor disjoint. A simple example covering both categories would be a model where the agent's workflow dictates its movements over a rugged landscape. Several models do not fit into any of the groups. For example, the work in [103] presents a simple, very general model of collaboration, in which details on a produced design's quality or utilised workflow are not modelled.

From the presented overview, one can conclude that - in general - current models represent agents' environments as abstract spaces. Two of the reviewed models, however, introduce a great level of realism in representing agents' workspace. Bellamine-Ben Saoud and Mark's [95] and Christian's [81] models enable users to simulate specific physical settings (e.g. room layouts, agents' location and possible routes). A modelling decision to include concrete environments stems from the models' purpose: these models aim to simulate information sharing among agents in great detail – in which physical location plays an important role.

Referring back to Salas et al. [38] work on elements of team performance models, one can discuss how work-assignment, task load, task type and task interdependencies are presented in reviewed agent-based models of product development teams. In 'landscape search-based' models, work is assigned to agents by specifying their initial position and search strategies, and task interdependencies are integrated into the landscape's definition. In 'workflow processing-based' models, both task assignment and interdependencies are explicated at the simulation start. Task load and difficulty are either included through ruggedness of the landscape or specified as a parameter assigned to each task. However, as further discussed in the next subsection, most of the models do not deal with the effect of workload and task difficulty on an agent's level of stress. Rare models such as [77] and [114] address the detrimental effects of work overload on team performance. Additionally, [120] takes into account the time required to 'recover' memories of interrupted and reworked tasks.

Time pressure is another element which has to be taken into account when modelling environmental and task influences on team member's stress level [38]. But most models do not explicitly include an increase in effort, stress or other changes to performance due to approaching deadlines. TEAKS [83] is an example of the model where delay in tasks causes an emotional response from agents. Some models implement changes in the agent's search strategies in order to match the human effort to converge to a solution after a certain period of exploration. For example, [112] implements agents' search as a simulated annealing process.

Finally, additional environmental factor influencing team performance is noise [38]. Models [80], [95] and [108] explicitly include effects of noise. Noise can, thus, consume time and attention of workers [80] and cause some messages to be lost [95]. Perhaps the most interesting effect is one where a message is wrongly interpreted or changed during transmission [108]. As further discussed in following subsections, although not explicitly including environmental noise, some models do include altercations in agent's communication effectiveness. For example, some include agents' parameters guiding their communication efficacy [81], [121], model differences in using various communication tools to exchange messages [80], [81], or take into account differences in knowledge backgrounds of communicating agents to determine the probability of successful communication [109], [111]. Nevertheless, the effect of environmental noise on the agent's stress level and development of faulty mental models, as well as the effect of information loss on team performance have rarely been studied in the current agent-based models of product development teams.

3.3 Representation of an individual team member

The level of detail introduced in modelling designers is necessarily related to the intended scope of the model. Thus, it is not surprising that the designer's characteristics included in the model greatly vary among the reviewed agent-based systems.

Many of the developed models simulate simple agents, guided by several rules. For example, agents in [105] transfer from one state to another following a 1^{st} order Markov process and the simulation is over when all of the agents reach an absorbing state. Similarly, many of the models where design problem is presented as a search over a rugged landscape implement agents that follow straightforward search heuristics. Brabazon et al. model [70], for instance, implements search as an adapted genetic algorithm: agents anchor in the best solution currently found and, through replication and trial-and-error mechanisms (i.e. mutation), try to produce better designs. Herrmann [122] models agents with two problem-solving styles and several search heuristics (hill-climbing, fixed effort and target values). Agents in Mihm et al. model [110] are characterised only by their progress rate that influences the speed at which they approach the local optimum. The agents' understanding of optimum may be imprecise as their perception of other members' progress may be obsolete. In these models, agents do not differ in their capabilities, personalities, emotions or motivation. Learning in terms of capability improvement is not implemented, and cultural factors are not taken into account. Thus, many of the elements important for characterising individual team members (as proposed by Salas et al. [38]) are simplified or seen as not influencing the problem of interest. Similar to the models [67] and [68] discussed in the previous chapter, such high levels of abstraction and simplicity of modelling assumptions facilitate analysis and promote understanding of the results. However, many of the agent-based models of product development teams tried to include more detail, provide greater veridicality and represent particular aspects of designer's behaviour.

To present an overview of approaches to modelling individual designers, each of the elements emphasised by Salas et al. [38] framework is briefly discussed. Thus, modelling of the designer's mental model and learning, expertise, cognitive ability, personality, motivation, attitude and emotions, and cultural background should be discussed. However, the literature review revealed that the cultural background of team participants has rarely been discussed in works presenting agent-based models of product development teams. Only two models – both extending the VDT [80] model - take into account some relevant issues. Power-to-the-Edge (POW-ER [123], [124]) model implements different hierarchies among agents to represent work structures characteristic for particular cultures. POW-ID [125] builds on POW-ER and implements a possibility of simulating agents in different time zones. Nevertheless, the agent's cultural background which may influence its social distance, decision-making or leadership behaviours [38] has not been included in the models. The current practices in modelling the remaining aspects of interest (mental models, expertise, cognitive ability, personality, motivation, and attitude and emotions) are discussed in the following subsections.

3.3.1 Representation of a designer's mental model and learning

Salas et al. [38] define a mental model as a human-generated description (or a view) of a system, its purpose, form, states and functioning. Mental models develop through experience and influence one's actions. The perceived system can be technological or human (e.g. a team). Thus, common practices in implementing the agent's mental models regarding team and team task are explored. Following the discussion on task representation presented in the previous subsection, one can separate two cases of agent's task processing.

In the case where agents are given a series of tasks which they have to process, most often no detail task representation is formed within the agent's mental model. Rather, parameters regarding the agent's competence and task difficulty determine the required processing time and the probability of rework. Christian [81] presents one model where agents store details on the tasks relevant for their role. For example, the agents in [81] remember the last time an information on a task's progress is updated and use it to predict the task's current status. Several models ([99], [116]) represent task processing with a system dynamics model where the rate of progress depends on the agent's work efficiency. In models where agents are scheduling their tasks based on their preferences, agents are equipped with knowledge on task requirements and resource capacity and availability. Such knowledge enables determining the preferable order of tasks and guides the dynamics of the agent's attention allocation. Additionally, one can discuss how the agent's behaviour or performance changes throughout the simulation. In 'workflow processing-based' models, learning – when included in the model - is most commonly represented through a reduction in time needed to process a task and an increase in agent's capability or knowledge parameter. When repeatedly faced with a task, the time needed for the agent to finish it changes based on the learning rate prescribed by the modeller. Such learning-by-doing is implemented in, for example, POW-ER [123], [124]. Similarly, models in [92], [126] and [57] employ this technique, where [57] also incorporates improvement in design quality as a result of repetitions. Aside from learning-by-doing, several models include learning through collaboration with others. Agents in Xia et al. [99] and Zhang and Thomson [116] models, thus, perform tasks quicker and increase their competence parameters as a result of either collaboration or repetition. Models [77], [85], [93], [115] include only an increase in competencies as a result of interaction with team members, leaders or resources. Interestingly, of mentioned models, only [124] and [99] implement competence decrease, i.e. forgetting. Rather than an increase in competence or reduction in task performance duration, in models [96] and [94] reinforcement learning is implemented in agents to enable determining a sequence of actions which leads to team performance increase.

In the case where agents roam a rugged landscape, agents' perception of a task refers to their understanding of the landscape. In the majority of these models, agents are capable of accurately assessing the fitness of their position (i.e. found solution) and memorising the best solution found. At the same time, their search strategies remain unchanged throughout the simulation. Some models, however, introduce biases and inaccuracies in the agent's view of a task. An example of a biased perception of the team progress due to obsolete information on other's position (implemented in [110]) has already been mentioned. Similarly, agents in [104] have a (not necessarily accurate) perception of their own and their peers' position on the landscape. Agents in [112] display self-biased behaviour: when choosing which designs to work on, they assign a slightly higher score to the solutions they have generated. Martynov and Abdelzaher [108] introduce biases in the evaluations of solution's performance which stems from differences in agent's domain of expertise. Agents in [71] continuously develop (and change) perception of the goal function and, through reinforcement learning, update their production rules. Learning about the goal function (i.e. task) based on received feedback is also implemented in [106], [112], [113] and [111]. Agents in CISAT [112] model develop strategies based on the utility of performed moves. Agents in [106] and [111] store memories on past solutions and build on them to orient their search. KABOOM framework [111], for example, simulates serial position effect where agents assign more weight to early and recent memories than to intermediate ones. The agents in mentioned models ([71], [106], [111]–[113]), thus, continuously learn and forget – consequently changing their behaviour. While learning through collaboration in 'workflow-processing based' models is represented with competence increase, in 'landscape-search based' models, agents can learn from others by replicating their position (along all or some of the coordinates). Such imitation-based learning is implemented in, for example, [71], [72], [104], [122] and [109].

A search over a rugged landscape is easily understood as a cognitive activity of exploration (broad search) and exploitation (local refinement). Thus, such models inform understanding of the effect of a designer's cognitive behaviour on team performance. As already mentioned, some models (e.g. [112], [113], [127]) employ simulated annealing to match the agent's cognitive behaviour with trends observed in studies of designers. The model presented in [105] takes a different approach: it builds on a well-known ontology of design processes (Function-Behaviour-Structure ontology, FBS [128]) to represent the agent's cognitive behaviour. As a result, the model's assumptions and/or results can be straightforwardly compared to the empirical studies utilising the same ontology. For example, in Bott and Mesmer's model [105], agents' mental models are calibrated to fit the transitions from one design process to another as observed in the real-world data (see [129] for details).

In addition to perceptions of tasks, the agent's mental model may contain perception of its team members or, more precisely, team members' competence and domain of expertise. However, in the majority of cases, the agent's view of others is predefined (i.e. given at the simulation start), static and accurate. These agents use their knowledge to, for example, direct help requests to more experienced team members. Models presented in [75], [91], [94], [95], [99], [126] and [116] all develop agents with static and accurate view of knowledge distribution within a team. In some models ([83], [93], [104]), past interactions determine the agent's willingness to repeat the collaboration. Agents in [93], for example, follow the preferential attachment principle where success in past collaboration increases the likelihood of its repetition. Similarly, TEAKS [83], [130] agents have a parameter denoted as 'trust', which increases with experience and guides the level of collaboration. Although in these models ([83], [93], [104]) view of others changes throughout the simulation, such view is represented with a single parameter. But in several models, knowledge and skill required for the task are not coupled in a single parameter, i.e. multiple skills or knowledge of various domains are differentiated. In such models, the agent's view of others' capabilities should be dependent on task requirements: a team member may be perceived as knowledgeable in one, but not in other domain. The models addressing this view can be found in a later version of the VDT model, POW-ER [123], and Singh et al. [76], [131]. Singh et al. [76], [131] implement a model specifically targeted at studies of the development of agents' perceptions of other's knowledge and expertise. In this model, each agent possesses a matrix where rows correspond to agents and columns to knowledge areas. The values within this matrix denote a perception of agents' familiarity with each knowledge area (higher value means perceived greater expertise). The agents update their matrices on who-knows-what through interactions and observations of other's actions. An interesting approach to modelling agent's perception of others was proposed in Gero and Kannenngiesser's [44] and Singh and Gero's [132] models where each agent represents others in terms of their perceived function, behaviour and structure. In other words, the works [44] and [132] use FBS ontology [128] to describe designers. These models are aimed at studying how team expertise develops in temporary design teams. However, no implementation details are presented for either of the models.

3.3.2 Representation of a designer's expertise and role

In addition to a mental model, agents can have a domain of expertise and a role within a team. The domain of expertise relates to a knowledge area a designer has mastered. Depending on the implementation, the role can indicate one's position within a hierarchy and/or can be associated with particular tasks one has to perform. While a role is a distinct concept from the domain (and level) of expertise, both have indications on the level of authority, availability and seniority. Thus, this subsection gives a brief overview of approaches to modelling both.

Modelling domain of expertise by using FBS ontology ([128]) has already been described on cases of [44] and [132] models. A popular approach to modelling expertise is to represent a set of all relevant knowledge areas and skills as a vector. If there are N relevant knowledge areas and skills, an N-dimensional vector is assigned to each agent, and the values of vector coordinates determine the agent's competences in particular knowledge areas. A task the agents have to perform is then associated with the N-dimensional knowledge vectors by, for example, imposing requirements on the vector coordinates' values. This approach to modelling agent's domain of expertise has been employed in models such as [72], [75], [76], [97], [102], [108], [116] and [109].

Models [57], [75], [104], [130] and [116] also distinguish agents based on seniority: novice and expert (and intermediate [116]) agents are modelled and differentiated based on their competences and approaches to performing a task. For example, Singh and Casakin [75] emphasise expert agents are capable of using analogies from between-domain visual sources, while novice use within-domain sources. Fernandes et al. [57] assign different tasks, authority level, and quality and speed parameters to 'junior' and 'senior' agents. Similarly, Zhou et al. [93] classify participants based on their productivity: ordinary developer, core developer and innovation leader types are modelled and characterised by their knowledge parameter and completion rate.

Assigning different tasks and authority level to agents based on their predefined roles is implemented in models such as [57], [80], [83], [84], [92], [94], [95], [99] and [114]. Models like VDT [80], Zhang et al. [84] and VOICE [114] assign each agent a position in hierarchy: subordinates report their work progress to agents up the hierarchy and wait for their instructions when errors and failures are encountered. TEAKS [83] makes a distinction between manager, coordinator, specialist, technician and assistant agents. Xia et al. [99] develop manager and worker agents. Similarly, the model in [57] includes project lead and design agents but also models customers. Zhang et al. [92] differentiates management, technical core and common development agents.

Finally, one can model differences in time agents are committing to particular tasks or projects. Due to the discussed tendency of design teams to have fluid boundaries and temporary membership [3], members of design teams often do not share equivalent time allocation. Additionally, differences in position and roles may cause team members to vary in time dedicated to particular tasks. To include such differences, several models ([80], [81], [85], [91], [102], [115]) assign each agent the availability parameter that guides the likelihood of agents responding to the collaboration or help requests.

3.3.3 Representation of a designer's cognitive ability

Related to modelling of the agent's mental model is a representation of its cognitive ability. Cognitive ability is defined as an individual's capacity to process information and learn [38]. Sosa and Gero [71], for example, characterise each agent with processing and synthetic abilities which control how many domain rules and hypotheses can the agent generate and utilise/test. Singh et al. [106] discuss the cognitive ability in context of mental inertia, which influences how knowledge is extracted from memories. This model [106] includes individual differences in learning and forgetting rates, thus capturing one's learning capacity. Similarly, [72] distinguishes 'slow' and 'fast' learners.

In several models, the agent's skill and knowledge level are fixed (i.e. remain constant throughout simulation). In these models, cognitive ability and experience are combined into a single parameter guiding the speed of task processing. For example, first versions of the VDT model [80] characterise agents by their 'information-processing capability' (derived from skill level, skill type and available time), Christian [81] uses 'work efficiency', Zhou et al. [93] have 'completion rate', and VOICE [114] 'competency' – all of which are implemented as a single parameter guiding the agent's processing speed. TEAKS [83] and Farhangian et al. [97] models similarly include a parameter defining the agent's level of knowledge and skill, but additionally, these models incorporate parameters representing agent's capability to generate creative solutions.

3.3.4 Representation of a designer's personality

A great majority of models developed to date do not specify any details regarding the agent's personality. Some models assign each agent a single parameter guiding its tendency to work with others: Crowder et al. [85] introduce 'response rate' which determines the likelihood that the agent will respond to a help (or collaboration) request. The similar parameter is implemented in Xia et al. [99] (denoted 'willingness to help') and Zhou et al. [93] (denoted 'collaboration probability') models. Levine and Prietula [91] model three cooperative types:

co-operator, which describes the agent that always contributes to others and responds to requests, free-rider, which rarely contributes to others, and reciprocator, which bases its response on others' behaviour. Son and Rojas [103] include parameters of sociability and familiarity. Sociability signals the level of the agent's extraversion and guides the frequency of social interactions. Familiarity describes the agent's tendency to form strong bonds with its peers, which may prevent the agent from forming new connections (i.e. collaborating with agents who are not its current contacts).

The agent's susceptibility to influences of the team is captured in models [106] and [111]. Both of these models incorporate 'group conformity' as a parameter determining if the agent is likely to alter its reasoning to (not) comply with the group's evaluations. These models describe the agent's cognitive style: a preference for solution space exploration. KABOOM builds on Kirton Adaptation-Innovation (KAI) inventory [133] and – in addition to conformity - specifies the 'sufficiency of originality' and 'efficiency' preferences for each agent. Similarly, Singh et al. [106] define cognitive style through the preferences for exploration or exploitation duration.

The agent's preferences for task processing are also incorporated in [77] and [120]. In [77], each agent has a workload preference guiding the number of parallel projects the agent's wishes to perform. In [120], the agents are characterised by their preference for short or long tasks.

Of the reviewed models, the most elaborate representations of the designer's personalities can be found in [83] and [97]. TEAKS [83] model separates agents based on their personality trends: expressive, analytical, driver and amiable. Personality traits included in this model relate to the designer's extraversion, preference for working in a group, and trust tendency. As noted in the next subsection, TEAKS implements fuzzy rules through which the agents' traits and trends are influencing the agents' emotions and performance. Farhangian et al. [97] develop a detail representation of the agent's personality by building on the work on the Myers Briggs Type Indicator [134] and Belbin's team roles [135]. In this model, the complementarity of agents' personalities is used to predict team performance.

3.3.5 Representation of a designer's attitudes, emotions and motivation

The designers' attitude, emotions, stress level and motivation influence their task performance on a daily basis. Perhaps the most straightforward approach to capturing day-to-day variability in performance due to these factors is presented in [57], where the agent's work quality parameters and progress rate coefficients for each task are drawn from triangular density functions.

The only model explicitly incorporating the agent's emotional states and their influence on performance is TEAKS [83]. This model includes two positive emotions: desire and interest, and two negative ones: disgust and anxiety. The fuzzy rules are used to update the agent's internal state and its effect on the agent's performance. TEAKS agents have variable emotional responses to a task. By introducing stochastic values, the modellers in [83] capture the differences in agent's emotional states due to non-modelled factors: the same task may evoke different emotions (if the situation in which it was obtained changes). Models in [77] and [115] deal with the agent's level of pressure by measuring its workload. The model in [115] implements a simple, threshold-based approach by defining a workload level above which the agent's performance starts to significantly decay. In addition, for each task, the agents in [115] have allegiance (commitment or priority) and motivation parameters which influence the time required to process the tasks. In [77], on the other hand, work overload influences the agent's motivation level, which in turn influences the likelihood of creativity.

The agent's motivation level is also modelled in [85]. In [85] model, motivation depends on the shared mental models developed within the team. In turn, motivation influences working time and quality of outcomes. Additionally, models [82], [132] and [106] deal with the agent's motivation. In [82] the agent's actions are guided by its motivation level which is calculated from its state variables regarding intellectual performance, activity, psychological fatigue, empathy for its team members, and trust in others. The authors in [132] describe agents in terms of their experience, beliefs, motivation and attitudes, from which competence, confidence, willingness and persistence are derived. In [106], the authors differentiate award-based and praise-based motivation and build on them to derive the agent's attitude regarding self-efficacy. However, since the implementation details of these models ([82], [106], [132]) are missing, no additional insights can be obtained.

3.4 Modelling of interactions and team properties and behaviours

Previous sub-sections introduced many details regarding interactions among agents in reviewed models. For example, noise influences, help-seeking behaviour, a communication directed at error handling, and transactive memory formation (as a result of interactions) have all been mentioned. However, this sub-section gives a structured overview of standard practices in interaction modelling. Particular attention is given to team properties, processes and behaviours emerging from simulated interactions.

Among the reviewed models, the details on and frequency of agents' interactions vary significantly due to the differences in the level of granularity and task type simulated. For example, models [82] and [108] simulate collaborative tasks in which agents interact in each simulation step, and many details on their communication (e.g. the contents of exchanged messages) are represented. Model in [81], for example, has such a detail representation of agent communication that, as the simulation output, one can extract the time traces of discussed topics. On the other side of the spectrum are models in which such a high view is taken that no agents' interactions are explicitly represented (e.g. [102]). Many models (e.g. [80]) fall somewhere in between: they incorporate collaborative tasks, calculate the percentage of time spent working in pairs or in group setting and model the impact of collaboration on work progress or learning.

In models where agents are processing multiple, dependent tasks, frequently-implemented communication triggers include task interdependencies, error handling protocols, prescheduled meetings or collaborative tasks, and collaboration requests due to insufficiencies in knowledge. In VDT [80], for instance, agents report the encountered difficulties (failures and errors) to their superiors and wait for the resolution decisions. Similarly, when a pair of VDT agents are performing dependent tasks, they coordinate their work by either attending prescheduled team meetings or by contacting each other over one of the simulated communication tools. Meetings are also implemented in the [81] and [116] models, while models [84], [95] and [94] describe interactions aimed at resolving problems and errors. In some models (e.g. [83], [84], [94]), tasks which require cooperation and discussion with others are predefined and assigned to agents. Once the agent selects a collaborative task, it sends a request to agents relevant for this task. When a required number of agents responds, the task is processed. As discussed previously, the parameters driving the agent's collaboration willingness and availability are sometimes introduced (e.g. [84], [85]), consequently influencing the interaction frequency. Interactions due to insufficiencies in one's knowledge are implemented in models such as [85], [93] and [131] where agents send help requests to their peers and, either learn through collaboration [85], perform the task jointly [93] or delegate their tasks to responders [131]. Similar communication triggers can be found in models focusing on the representation of agents' search for the solution (i.e. 'landscape-search based' models). In Mihm et al. [110] model, for example, the agents exchange information on their location either at pre-scheduled meetings or during random encounters. By employing Poisson process, the authors in [110] describe the unscheduled nature of updates, i.e. communication among agents. A probabilistic form of interaction is also implemented in models [112] and [111] where agents have parameters guiding their interaction rate, while [111] also incorporates prescheduled meetings. An equivalent of help-request sent in 'workflow-processing based' models, in 'landscape search-based' models occurs when an agent is stuck in a local optimum (i.e. insufficient knowledge). The agent then tries to improve its position on the landscape by contacting others and imitating (some or all) coordinates of their location or their search rules (e.g. [71], [109]). The agents in [71], thus, imitate others if their current search heuristics are insufficient to generate satisfactory results. As already stated, in some models, the agents interact in every simulation step due to the nature of the simulated task and the level of granularity taken in the model. The models reported in [72], [78], [82], [106], [108], [131] simulate agents performing a collaborative task during which agents are exchanging their solution proposals at every simulation step. Finally, some models enable choosing (as a simulation input) among several collaborative strategies employed by the agents, thus defining the collaboration patterns and frequency (e.g. [113], [122]).

Following the described communication triggers, one can conclude that interactions often occur between agents whose tasks are interdependent (to coordinate their work) and up-the-hierarchy (to resolve encountered issues and report progress). To enable such interactions, models [80], [81], [84] and [131] specify a hierarchy among agents and predefine task interdependencies. Models [95] and [94] assign each agent with an interdependency map at the simulation start. The model in [109] employs a 'Connected cavemen' [136] network model to describe inter- and intra-team interaction patterns. These models, thus, predefine which agents can interact. Some models, however, enable agents to choose their collaborators either randomly, or based on their perception of others. Random collaborator selection is implemented in, for example, [70], [71] and [85] models. In other models, the agents rely on their perception of others' knowledge (i.e. transactive memory) or performance to choose the best collaborator. For instance, agents in [131] constantly update their understanding of others' expertise area and delegate tasks to each other accordingly. Agents in [72] observe the performance of their potential collaborators and, if a more successful peer is detected, tries to improve its performance by learning observed skills.

From the presented overview, one can conclude that, within reviewed models, communication among agents frequently occurs in pairs (e.g. [80], [81], [84], [110]), but often also in the form of group tasks or meetings. Models such as [82], [94], [95] implement communication in a way that agents' messages are broadcasted to the whole team.

In many models, interaction duration is either prescribed (equating either one step [72], several steps [95], or occurs at every step throughout simulation [108]). In some models, however, the duration of interaction depends on the collaborating agents' characteristics, knowledge and task difficulty. In VDT [80], coordination work duration is specified based on task uncertainty, complexity and agents' skill match with regards to the performed task. Additionally, VDT [80] takes into account the time needed to prepare and process communication occurring over one of the communication tools (e.g. time required to write or read an e-mail is modelled). Similar calculation of interaction duration based on agents' work efficiency, communication efficiency, communication channel or task type is modelled in, for example, [81], while in [85] difference in collaborating agents' competences and learning time parameter are used.

While most models either do not differentiate the communication channel, or model solely face-to-face communication (e.g. [82], [85], [94], [95]), several models specify communication tools which can be employed by the agents. The usage of communication tools can manifest in the amount of information transmitted and collaboration duration, but it also enables asynchronous collaboration. Agents can collaborate asynchronously in models such as [80], [81], [84], [132] and [92]. Irrespective of the synchronous or asynchronous collaboration mode taken, the majority of models implements only direct interaction among agents. The models reported in [95] and [131] also include indirect interaction, through overhearing other agent's communication [95], or observation of their actions [131].

Regarding the contents of the exchanged messages, the most common message types include progress report, error-handling messages and help requests in simulations where the agents are processing sequences of tasks (e.g. [80], [85], [114], [115]). In cases where agents are roaming the landscape, the agents usually exchange the details on their location (e.g. [104], [108], [110], [112], [122]). The agents in [91] exchange resources, while in [131], agents delegate tasks. An interesting feature of the model reported in [81] is that the agents can engage in gossip. However, the gossip only takes time and signals the usefulness of conversation; it does not affect the overall team performance or climate.

Many of the biases and external factors influencing the communication outcomes have already been discussed. Models such as [81] and [121] characterise agents based on their communication efficacy. This parameter captures one's capability to transmit information clearly and comprehensively. Thus, the lower values of the communication efficacy parameter result in the longer time needed to transfer the required information. Gero and Kannengiesser [44] discuss the need for establishing a common ground for communication to be successful.

In other words, the authors [44] emphasise that the cognitive gap among agents should be sufficiently small for them to understand each other's messages. The models such as [111] and [109] take into account the effects of cognitive distance on communication success. As discussed, communication tools are another factor influencing the information transfer. Various communication tools can be implemented: telephone, voice mail, e-mail, memos, fax, as well as internet or cad tools [80], [81], [132]. These communication and information tools can be characterised by parameters such as synchronicity, cost, recordability, capacity, portability, agent retrieval time and maximum simultaneous usage [80], [81]. Usage of these tools can either limit or facilitate communication [132]. Lastly, environmental noise impacts the message reception [95] and interpretation [108], or consumes the agents' attention [80].

As a final aspect of agents' communication, one can discuss its outcomes. In many models, interactions among agents consume time, but result in improved quality of the product and reduced probability of failures (e.g. [80], [81], [83]–[85], [115]). In addition, several models implement interaction outcomes in the form of learning in terms of competence increase or generation of new search heuristics or solutions (e.g. [71], [72], [85], [94]). As a result of communication and learning, the agent's perception of its task and the team can change. For example, in [110], an interaction among agents modifies their understanding of the task. Models presented in [106], [111], [132] develop agents whose perception of the suitable solution changes as a result of the opinion of the team. Thus, the agents may alter their actions to either conform to the team or to defect from the team-proposed solutions.

But perhaps the most interesting outcomes of interactions among agents are the emergent team properties, behaviours and processes. Although many models could be extended to enable capturing the team-level properties emerging from the individuals' interactions, in most cases, studies of emergence were not the primary purpose of these models. Thus, only a fraction of the reviewed models readily captures emerging team-level phenomena (see Table 3.1).

As discussed in Section 3.3.1, several models equip agents with the (dynamic and not necessarily accurate) knowledge on 'who-knows-what', and such knowledge is refined through either direct interaction or observation [131]. This permits studies on how transactive memory develops within a team. Studies of transactive memory formation and its impact on team performance and coordination is the main purpose of the model presented in [131], but models [80], [132] and [44] can also offer some insights. As their primary purpose, models [44] and [132] are directed towards capturing the emergence of team expertise. Although the initial versions of VDT did not include any emerging team properties, its extensions [123]–[125]

detailed agent's learning mechanisms and subsequently enabled simulations of trust and transactive memory formation studies.

Table 3.1 Emerging team properties and processes simulated by the reviewed models

Model and references	Emergent team properties and processes
VDT and its extensions [80], [123]–[125]	Trust Team learning Transactive memory
TEAKS [130]	Trust Team collaboration
Gero and Kannengiesser [44] model	Team expertise Transactive memory
Singh and Gero [132] model	Team expertise Transactive memory
Crowder et al. model [137], [138]	Trust Shared mental models Team learning Team motivation
Singh et al. model [76], [139]	Transactive memory
Dehkordi et al. [77] model	Team motivation Team creativity
Sosa and Gero [78] model	Team creativity
Dutta et al. [115] model	Team motivation Team learning
Singh and Casakin [75] model	Team cohesion Team collaboration
AMPERE [57]s	Team adaptation
Xia et al. [99] model	Team learning
Zhang and Thomson [116] model	Team learning
Singh et al. [106] model	Team leadership

The model developed in [130] deals with the emergence of trust and its influence on team collaboration. Similarly, Crowder et al. [85] capture trust and team learning, but also add a measurement of team motivation and shared mental models. However, within this model, the changes in these team properties are modelled through simple equations. First, trust increases due to interactions among agents. As a result shared mental model linearly increases, which - in turn – causes the rise in team motivation and competence. A similar approach is taken in the model reported in [115]. Dynamics of team motivation, as well as its influence on team creativity, are captured in [77]. Sosa and Gero [78] explore how individuals are influenced by the ideas of other team members during brainstorming sessions. Singh and Casakin [75] are studying how the use of analogies affects the emergence of team cohesion and collaboration. A model aimed at studying a team's capability to accommodate changes in task requirements is presented in [57]. By modelling improvement in quality and reduction in time required to perform a repeated task, AMPERE [57] estimates the team's adaptation to changes occurring during early design phase. Xia et al. [99] and Zhang and Thomson [116] model one's increase in competences as a result of task performance or collaboration. The model in [99] studies how task assignment strategies and communication affect the knowledge evolution within a team. However, these models ([57], [99], [116]) do not implement details on the agent's cognitive processes. Instead, knowledge is represented with a single parameter or a knowledge vector, and learning is modelled as an increase in competence value upon task completion. Finally, a recent model proposed in [106] is directed towards capturing the emergence of team leaders within flat teams. The authors [106] hypothesise that, as the team members interact and gain experience, their self-efficacy will change and influencers will naturally emerge (although no hierarchies were specified in the model).

3.5 Research implications

The review of the models presented herein revealed that the majority of models developed to date are directed towards studies of certain team properties and processes, or towards providing a general overview of the process and potential problems. In both cases, it is convenient to develop simple agents that implement only the specific aspects of a team member's behaviour and performance. In the former case, the omitted processes, properties and behaviours are regarded as having little or no direct influence (as posited by the existing theories) on the aspects of interest. They would, thus, introduce needless complexity to the system. In the latter case, the coarse-granular perspective renders detail modelling of most of the detailed processes as unnecessary. As a result, it is not surprising that none of the models addresses all of the team performance model elements proposed in [38]. For example, VDT and its extensions ([80], [123]–[125]), TEAKS [83], [130]; Singh and Gero [132], and Singh et al. [106] models cover multiple aspects of individual designer's behaviour and present some of the most comprehensive representations of humans among reviewed models. However, none of them includes every factor from the list provided in [38]: mental models, expertise, cognitive ability, personality, motivation, cultural factors, and attitude and emotions.

One can also note that of the models enabling studies of team properties emerging from interactions among agents (Table 3.1), models [44], [75], [132] and [106] provide only a description of a conceptual model, while implementation details are not yet reported. The relative scarcity of models focussing on emergent team properties emphasises a great research opportunity for computational modelling and simulation (and, particularly, agent-based) studies within the design field.

The work presented herein strives to enable cognitive studies of teams, thus supporting one of the most important directions in team research [140]. Therefore, works modelling agents' cognitive behaviour and learning are of special importance for informing the model developed within this work. Various useful ideas on the task representation and agent's mental model implementation can be derived from reviewed models. For example, the model presented in [105] introduced a mental model tailored to represent the fundamental design processes (proposed in [128]), which facilitates comparisons with real-world studies of designers (e.g. calibration to protocol studies data). Similarly, the models which conceptualise a design task as a search over a rugged landscape demonstrate an elegant way of representing designer's cognitive processes of exploration and exploitation. The models such as [111]–[113] and [106] discuss how different patterns in the agent's exploration and exploitation serve to represent one's effort to converge to a solution and can be linked to a designer's cognitive style. The reviewed models also enable insights on possible implementations of important aspects regarding the agent's learning behaviour: learning about task, learning-by-doing, forgetting, and learning about and from others. Finally, although none of the models integrates all of the aspects emphasised in [38], several valuable inputs regarding modelling of designer's details (e.g. personality), diversity and interactions can be obtained.

Of the reviewed models, only AMPERE [57] captures how learning affects team adaptation to changes in design requirements. However, in this model, details on agents' cognitive behaviour are omitted, and learning is implemented simply as a reduction in time required to perform a repetitive task. In developing a model for studies on the effect of learning on team performance, a recent model presented in [116] can offer several useful ideas. In [116] and [99], the agents improve their (technical and general) knowledge by repeatedly performing a task, as well as by communicating with others.

Building on these models, the work presented herein develops a model enabling cognitive studies of teams, team learning, and the resulting change in team performance over multiple tasks.

4 MODEL SPECIFICATION AND THEORETICAL FOUNDATION

This chapter gives a theoretical foundation of the desired model of a design team. To provide a modelling starting point, the literature was searched in order to answer the “what should be modelled - and how?” question. Research on the modelling of human behaviour in a social setting directed the identification of relevant components to constitute the agent’s architecture and enabled the development of the architecture of the overall system. Then, each element of the agent’s architecture is described through an overview of related theories and empirical studies, thus providing a basis upon which the model is developed. The results presented in this chapter build on the work published in [141].

4.1 What to model? – Identifying architectural components

Following the research aims laid out in the first chapter, and building on the presented research background (Chapter 2), and existing models of product development teams (Chapter 3), the goal of this thesis can be summarised as ‘*the development of an agent-based model of a design team which enables (cognitive) studies of the emergence (and change) of various team properties and behaviours*. When identifying the architectural components of the desired agent-based model, a good starting point can be found in the previously described work by Salas et al. [38]. However, the work [38] gives a general overview of relevant aspects, providing a little insight into how identified model elements should be realised and interconnected. The choice of agent-based modelling as a modelling approach sets a bottom-up perspective in which agents are designed to mimic real-world entities. Within this work, each agent represents a designer. Thus, studies on - and models of - human behaviour in a social setting in general, and designer’s behaviour in a team setting in particular, are used to identify the agent’s architecture components and describe the model structure.

In a review of approaches to modelling human behaviour within agent-based models, Kennedy [142] emphasises several fundamental principles of human nature which should be kept in mind when modelling humans. Particularly, humans are described as information processors. But the author emphasises boundedness (e.g. limited knowledge, cognitive ability and time to make a decision), diversity (e.g. in terms of personality, preferences, capabilities or motivation), and biases (e.g. due to emotional or social factors) of human cognitive behaviour. Listed features imply a requirement of accounting for the impact of *personality and capabilities* on human cognition, but also highlight the influences of *affective* and *social* aspects of human behaviour.

Inline are findings reported in [143], where the authors identified *cognition, affect, social aspects, norm consideration* and *learning* as key dimensions in modelling human decision-making.

Several conceptual frameworks aimed at human behaviour modelling have been developed and extensively used as frameworks in developing agent's architectures. One of the best known is the Belief-Desire-Intention (BDI) framework [144]. As evidenced by its name, BDI framework represents human behaviour by modelling one's beliefs – internalised information about the world, desires – all possible states that the agent may be motivated to achieve, and intentions - deliberative states arising from a commitment to a particular desire. BDI agent perceives the world, which can trigger an update of the agent's beliefs. Throughout the simulation, the agent is actively searching for a plan (i.e. a sequence of actions) which would lead to the achievement of intentions organised in an intention stack. The plan is selected based on its relevance to the agent's beliefs and intentions. Due to its generality, BDI framework can be used in many application areas. It does not pose any strict rules on the implementation architecture, thus permitting multiple implementations. On the other hand, its generality can be seen as a limitation: Kennedy [142] argues that BDI framework provides a little more than a conceptual framework for designing a model of human cognitive behaviour. No learning mechanisms incorporated, as well as the absence of social aspects (e.g. communication), are clear limitations of BDI framework in the context of the work presented herein. Additionally, the BDI framework has been criticised for lack of affective mechanisms influencing cognitive processes [143]. One BDI extension that mitigates such critiques is eBDI (emotional BDI) [145], [146] in which emotions are taken as one of the decision criteria. However, the eBDI framework has not yet received the popularity of BDI. An overview of eBDI proposals can be found in [147], which shows how recent work on emotional BDI agents – aside from modelling emotional responses - strives to incorporate influences of moods and personality on cognitive behaviour.

Another frequently referenced framework is PECS (Physical conditions, Emotional state, Cognitive capability and Social status) [148]. It builds on BDI, intending to mitigate its limitations. PECS architecture can be separated into three layers. The first, input layer, is concerned with sensing and perceiving the environment. The second, internal layer, models the agent's internal state in terms of physical, emotional, cognitive and social aspects. Each of the included aspects is separated in a corresponding component and characterised by an internal state and functions describing the change of the internal state. The final, output layer, deals with calculating the behaviours and determining actions. Authors argue that separation of different aspects into components permits the definition of a wide range of inter-component dependencies, consequently realising various behaviours. PECS is intended to provide a generic framework for simulating humans, but such generality came at the expense of the lack of detail. PECS is not based on any particular social or affective theory, and presents little to no detail on how each component should be implemented [143], [149].

In an effort to identify a set of behaviours that "must be integrated into a model to simulate individual behaviours in human-centred systems", Elkosantini [149] reviews several behaviour models and proposes an architecture comprising *personality, emotion, cognitive capacity, social environment, psychological capacity* and *physical capacity* components. Although this model emphasises several frequently neglected factors – such as the impact of psychological factors (motivation, self-efficacy and stress) and physical capacity (e.g. fatigue), the details on the implementations are missing. As such, learning, social interactions, emotions, and cognitive processing are not sufficiently described or are entirely omitted. An overview of several additional frameworks for human behaviour modelling can be found in [143].

Summarising the findings of listed frameworks, one can identify the following dimensions in modelling human behaviour: *cognition, affect, social status, personality traits and capabilities,* and *physical capacity*. Additional dimensions mentioned, such as learning, can be seen as an integral part of the agent's cognition [150]. Similarly, factors related to psychological capacity mentioned in [149] (e.g. motivation and stress) are closely intervened with the modelling of agent's affective states [151] and, thus, can be incorporated within the affective component. It is important to note that in the context of the study at hand, physical capacity is considered as of lesser importance since designers are (usually) not exposed to dangerous or physically demanding situations.

The presented discussion brings to the conclusion that the desired agent should be cognitive, social and affective, and possess differences in personality traits and capabilities. An interesting discussion on the relation among the three listed aspects (cognitive, social and emotional) can be found in [150]. Carley and Newell [150] seek an answer to the question "What is necessary to build an artificial social agent?" and posit that both, emotions and cognition, are prerequisites to realistic social behaviour. To provide an analytic framework, the authors study the agent's capability to display human-like behaviour along two dimensions: the agent's cognitive limitations and its knowledge of the socio-cultural context. Authors [150] argue that, if appropriate limitations are placed on agent's cognition and agent's knowledge of the socio-cultural context is enriched to enable recognising and responding to all classes of knowledge associated with the context, an agent becomes a highly veridical representation of human behaviour in all situations, i.e. the Model Social Agent.

The work [150] stresses the necessity of modelling cognitive limitations and mechanisms which impact processing capabilities (one of which is affect). But it also emphasises the agent's situational knowledge. The highest level of situational knowledge (i.e. the most realistic) is the one in which current agent's situation is seen as a result of the particular historical process, i.e. in which the agent is shaped by a specific culture and is accustomed to specific norms and values. Although such a high level of realism is unattainable within the scope of this work, the framework presented in [150] can serve as a guideline in creating an increasingly veridical system.

All of the works mentioned ([144], [148]–[150]) oppose modelling of humans as omnipotent, rational or even boundedly-rational agents (as classified in [150]). Instead, aside from being affective and constrained regarding the time to process information, agents should have limited, biased and possibly out-dated views of the world. Such a view is developed through a series of experiences in which past events are influencing the understanding of the current situation, while current experiences are, in turn, reshaping the understanding of past events. Many of the agent-based models of design teams (reviewed in the previous chapter) do not introduce that level of complexity when modelling designers, as details on agent's cognition are not necessary for the models' purposes. However, several notable artificial models of designer's cognition can be found within the field of design support systems and human-computer interaction systems (e.g. [152]–[157]). Namely, listed works introduce the *situated design agent*. Similar to the [144], [148] frameworks, the situated design agent senses the environment, perceives the input information, categorises it based on previous experiences to assign meaning to it (i.e. performs conceptualisation), and uses the obtained concepts to determine most appropriate actions. The cognitive theories underpinning the work in [152]–[156] are centred on the idea of *situatedness* - a notion that 'where you are when you do what you do matters' [158]. Researchers like [152], [159] consider situatedness to be "an essential aspect of intelligent behaviour". In contrast to a static worldview, situatedness captures the idea that the same thing can be understood differently if the perspectives, from which one had approached it, are different. Related is the concept of *constructive memory*. The theory of constructive memory [160], [161] posits that past experiences shape how the world is perceived, while simultaneously, the newly acquired experiences give meaning to past experiences. In other words, agent's memory is seen as both, a knowledge construct and a learning process that enables situated design agents to change (i.e. adapt) over time [152], [153]. Studies in neuroscience and cognition [161], [162] corroborate such a dynamic view of memory. Constructive memory can be simulated using, for example, constructive interactive activation and completion neural networks [152], [155], thus enabling the implementation of situated, cognitive agents [153]. A computational framework formally describing the relationship between (individual-level) adaptation and situated design agents' constructive memory can be found in [163].

Although the research on situated design agents models agent's learning and reasoning in detail, affective, social and personality-related aspects of an agent's cognitive behaviour are less (or not at all) represented. One model developed within the design field that can be used as inspiration to incorporate all of the identified aspects is [164]. Building on the notions of situated cognition, Thomas and Gero [164] developed a cognitive, situated, social and affective agent whose perception and conception processes are implemented as self-organising maps. This model includes many of the agent's desired properties, but it is tailored to simulate consumer's (rather than the designer's) behaviour.

To complete an overview of important aspects to be included in the desired model, one must examine the question of the team-level properties that need to be captured by the model. Since this work is concerned with cognitive behaviour of design teams over time, team learning (and, consequently, adaptation) is of particular interest. Team learning is regarded as a consequence of interactions among members during which members acquire, share and combine knowledge [165], [166]. Kozlowski and Bell [167] emphasise how team learning emerges from the individual-level behaviour: team members' knowledge and skill intersect and combine ultimately manifesting as team-level knowledge and skill. An important issue is that of appropriate metrics to capture the processes of interest [61]. The authors in [31] emphasise that team learning is typically observed through the lenses of team performance change. Still, it also manifests in the development of team mental models and transactive memory systems, changes in behavioural capabilities (cooperation, coordination and communication), and a change in affective and motivational states [167]. Thus, the model developed herein should be able to capture (at least some of) the listed elements.

4.2 Agent architecture

Following the theoretical background laid out in the previous subsection, the requirements on the agent's cognitive behaviour can be summarised as:

- Agents should be cognitive – i.e. able to reason and choose their actions, but their reasoning mechanisms should not be modelled as perfectly rational. Rather, the agent should have biases and imperfect knowledge of its environment;
- Agents should have dynamic mental models – i.e. capable of learning which, consequently, allows a change in behaviour;
- Agents should be situated – agent's behaviour should depend on its interpretation of the world and should change in accordance with situation perception. Coupled with the requirement of a dynamic mental model, a requirement of constructive memory can be formulated;
- Agents should be affective – the affective mechanisms should bias agent's cognitive behaviour and impact processing capability;
- Agents' cognitive ability and personality should be modelled to introduce differences in learning and preferences;
- Agents should have a perception of others – somewhat related to the requirement of situatedness, the agents should have a dynamic perspective of its social context. This property enables formation and studies of the transactive memory and trust.

In addition, following the discussion in Chapter 3, the agent's mental model should be implemented in a manner which permits establishing a straightforward relation to the results of empirical design studies (e.g. by enabling extraction of the output comparable to data obtained via protocol studies of designers). Although team learning and behavioural change can be studied without relating the agent's mental processes to design, the inclusion of a design-related mental model provides a point of comparison among simulated data and real-world findings. Coupled with the requirements emphasised by Salas et al. [38] (see Chapter 2), the provided list summarises the desired agent's properties and behaviours.

Building on the research reported in [154], [164], [168], the agent's cognitive architecture can be envisioned as presented in Figure 4.1. In the figure, one can observe several major 'building blocks'. The cognitive mechanisms constitute the most important aspect of the agent's architecture as they handle the obtained knowledge, shape the agent's perception and direct its behaviour. The implementation of these mechanisms should accord with the prominent cognitive theories and enable the functionality of constructive memory. Additionally, the cognitive mechanisms should be in charge of goal maintenance and expectation generation and update. Affective states should influence the manner in which perceived information is processed. Finally, cognitive ability and personality traits should be introduced to simulate differences in processing capabilities and coping mechanisms.

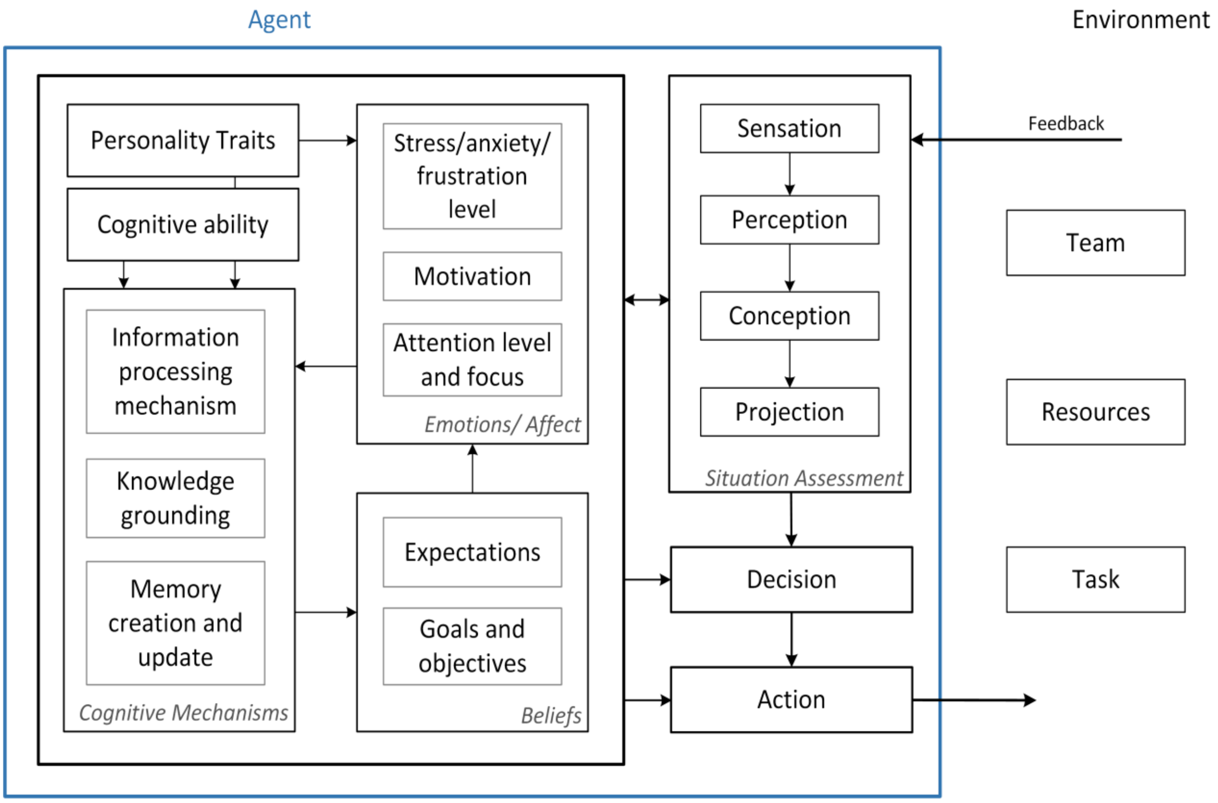


Figure 4.1 Architecture of an agent representing a designer (i.e. one team member)

4.3 System architecture

Similar to the case of the agent's architecture, the presented theoretical background can serve to derive a set of requirements for the overall system's architecture. The most important aspects for the studies of emergence are interactions among systems parts. In particular:

- the agents should be able to interact with each other and exchange knowledge in order for team learning to emerge [61]. Additionally, team members should be able to interact with the task and - if required – with resources.

Kozlowski and Bell [167] emphasise the importance of beliefs team members hold for each other by noting that they enable and shape team learning. Transactive memory systems and trust among team members depend on a context, and have a significant impact on interactions among team members. Thus, team members should be able to learn about each other, i.e.

- team members should form transactive memory and trust, and act upon them.

Following the work laid out in [167], to enable measurement of team learning, the desired system should capture the changes in team performance, transactive memory system development, improvements in communication and coordination, but also the fluctuations of the members' affective states. Thus:

- the system should enable an analysis of at least one measure providing insights in team learning (i.e. team performance improvement, communication, coordination, change in the transactive memory system, and state of the members' affective states)

Finally, similar to the requirement of design-relatedness of the agent's mental model, the simulated task should be design related. Following the discussion in Chapter 3, the modelled design space should be unbounded to enable studies of the solution space exploration and expansion.

The system architecture compliant with the listed requirements is presented in Figure 4.2. Each agent, represented by a circle, possess a perception of the current situation, i.e. about itself, other agents, tasks, resources (if any) and relations between these elements. The perceptions are dynamic and are influenced by interactions and past agent's experiences. Such representation enables extraction of properties such as transactive memory systems or trusts among agents. In addition, one should be able to extract various aspects regarding agent's interactions.

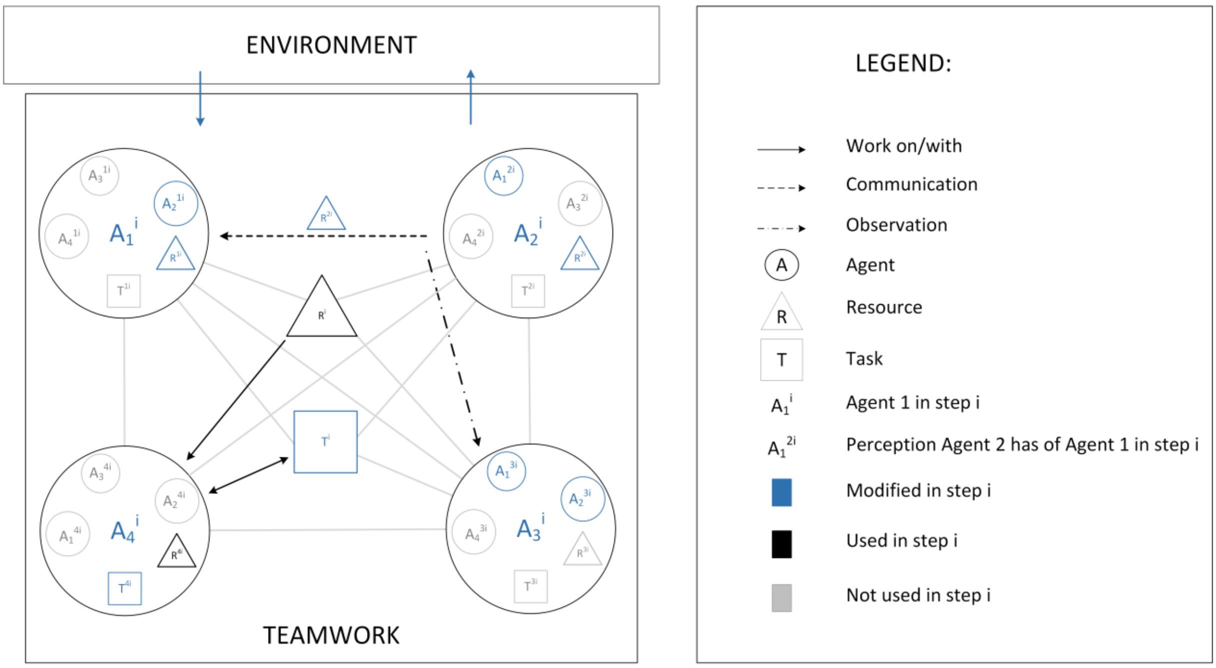


Figure 4.2 Architecture of the system

4.4 Theoretical background on the agent's architecture

Upon determining the agent's architecture, relevant theories, empirical studies and models need to be examined to inform the implementation of the identified model's components. Thus, the following subsections review the related work on cognition, affect, personality traits and cognitive ability, and trust and transactive memory system, and relate it to designers' behaviour.

4.4.1 Cognition

This subsection explicates the relevant theoretical backgrounds, as well as existing computational models that serve as a basis for the development of the agent's mental model. Although interlinked, in order to ease reading, the subsection is divided into segments related to cognitive architectures, theories of human thinking and design ontologies (in particular, FBS ontology [128]).

4.4.1.1 Cognitive architectures

Of particular interest for the goal of this work are existing computational systems directed towards modelling the human mind: *cognitive architectures*. Recently, Kotseruba and Tsotsos [169] published a review of cognitive architectures developed over the past 40 years. The authors have identified more than 300 cognitive architectures, of which approximately one third is still actively developed. Most prominent among these, ACT-R [170] and SOAR [171], share many commonalities and were, therefore, (along with another well-known cognitive architecture, Sigma [172]) used to define the so-called Standard Model of the Mind [173]. It is important to note that the Standard Model of the Mind is not a comprehensive model of human cognition, (omitting, for example, emotions, personality traits, drives and motivation which architectures such as CLARION [174] successfully relate to cognitive processes). The authors [173] emphasise that such omission is not due to the unimportance of these parts, but rather stems from the lack of well-established consensus. Mentioned cognitive architectures [170], [171] are the most comprehensive, validated architectures developed to date [173], and the model presented herein cannot match their level of detail and precision. Nevertheless, many of the implemented elements draw inspiration from these architectures.

The central aspect of cognitive architectures [170], [171] is the *working memory*, a temporary mental "global space" [173], which serves to combine perceived input information and long-term memories, in order to assess the situation and determine desired actions. Despite what its name may imply, working memory is not restricted to storing information. Instead, working memory is a term covering all of the mechanisms used in control, regulation and active maintenance of information relevant for the task at hand [175]. The notion of working memory was developed from the idea of short-term memory [176], and it contains the part of the long-term memory (particularly, part of declarative knowledge [177]) that is attended to in a given moment [178]. In that sense, working memory is tightly connected to attention [175]. The constructive memory and concepts of working memory, short-term memory and long-term memory represent different sides of the same phenomenon (i.e. cognition): while constructive memory concerns the way interactions between the agent and its environment give rise to a memory [154], working memory, short-term memory and long-term memory present the functional aspects of human cognition.

An important question related to modelling cognitive processes is whether working memory is limited and, if so, how. The nature of working memory limitations is discussed by each of the ten models of working memory in [179], as well as studied in a number of studies (e.g. [180], [181]). The consensus among researchers is that working memory is constrained [181] and that its capacity varies among individuals [182]. Additionally, Cowan [183] argues that the capacity varies with age, peaking at young adulthood. A frequently cited theory proposed by Miller [184] suggests the "magical capacity" to be seven (plus or minus two) elements held in the working memory in parallel. Cowan [183], on the other hand, argues that while the number of elements that can be active in a certain time is unlimited, the number of *focused* (i.e. attended to) elements is four (plus or minus one). However, the elements held in working memory are not necessarily of the same (i.e. unitary) size. Instead, researchers [185], [186] argue that by gaining experience, individuals organise knowledge elements in a meaningful way, thus allowing them to memorise multiple elements as a single unit called *chunk*. In the recent years, alternative theories emerged postulating that (rather than containing a constrained, discrete number of elements), working memory capacity is resource-limited in a sense that a certain amount of attention can be divided unequally among many units [181]. In this view, the amount of resources obtained by a unit in working memory influences the precision of its processing. Researchers [181] advocating this view note that the theory of limited resources can explain the relation between increased working memory load and decline in performance. Nevertheless, more research is needed to determine if these theories are indeed incompatible, and which (if any) captures the nature of working memory capacity.

To computationally model relevant processes related to working memory, ACT-R and SOAR architectures utilise *spreading activation* [187], [188]. Spreading activation is an algorithm in which a sufficiently active node (i.e. node whose activation is above a certain threshold) "fires", thus triggering propagation of activation to the connected nodes, with some of the activation being lost in the propagation process. In these architectures, the concepts stored in the long-term memory are represented as a graph of symbolic relations, and spreading activation algorithm explains the attention-shifting from concept to concept over the graph. To start activation spreading process one must specify the task's goal. The goal then serves as an activation source, thus causing the working memory to contain nodes related to the current task. Links connecting concepts are weighted to indicate an associative relation between them. Namely, the strength of the relations is updated based on the *Hebbian learning* [189] meaning that nodes that are active together form tight connections (or, as stated in [190], "neurons that fire together, wire together"). As the strength of the relation increases, the amount of activation that can pass between two nodes increases, therefore indicating that the link became more grounded and easier to process.

Aside from spreading activation, reasoning processes implemented in [170] are guided by *base-level activation* specified for each concept (i.e. declarative knowledge unit). Base-level activation indicates the concept's accessibility: each time a node is retrieved from the memory, its base-level activation increases which in turn means that less activation will be needed to retrieve the node in the future (thus serving as a learning process). Conversely, as the time passes without the node being attended to, the base-level activation of the node decreases (i.e. forgetting process). The base-level activation and activation received from spreading source activation in sum dictate the ease of the node's retrieval. The authors [177] argue that base-level activation accounts for the context-independent aspect of node's accessibility based on frequency and recency, while spreading activation deals with attentional activation guided by the current context.

Spreading activation and base-level activation guide the dynamics in the SOAR [188] and ACT-R [170] by controlling the amount of activation (by limiting the amount of source activation) and its duration (through the decay of base-level activation). It is important to note, however, that humans have the capability of attention control, intentionally inhibiting some nodes and directing their attention towards specific sub-goals [169], [181]. Following the taxonomy proposed in [191], there are three groups of mechanisms used for information reduction and attention control: *selection*, which involves extracting a single element of interest (among many), *reduction*, which filters out some of the elements, and *suppression*, which suppresses some of the elements. As noted in [177] and [169], some of these processes are not implemented in the ACT-R. Nevertheless, the past studies demonstrated the capability of the ACT-R framework [170] to match human performance on multiple cognitive tasks (e.g. [192], [193]).

4.4.1.2 Human thinking theories

The described processes of node activation nicely align with prominent theories of human thinking. For example, one can note how these processes can give rise to the two thinking modes introduced in [194]. The dual-system theory [194] distinguishes fast, automatic, effortless reasoning (so-called *System 1 thinking*) and rule-based, controlled, conscious reasoning (called *System 2 thinking*). System 1 thinking arises as a consequence of the past experiences that enabled the creation of strong associative relations among concepts, and does not consume working memory resources [195]. System 2 thinking, on the other hand, relies on deliberate processes of attention control, regulation and active information maintenance. Theories complementary to the dual-system theory were developed within the field of design computing [196], [197]. Maher and Gero [196] differentiate *reflexive*, *reactive* and *reflective* modes of reasoning. Reflexive reasoning denotes an automated response to stimuli, a reflex. Reactive reasoning includes decision-making and reasoning over several alternative responses to the perceived stimuli. The most effortful, reflective, reasoning involves an even higher amount of processing, filtering and hypothesising on the outcomes of each of the possible responses.

4.4.1.3 Design ontologies

Building on the theory and past research efforts described herein, this work aims to develop a system capable of simulating human thinking [194]. More so, the goal is to develop an agent whose mental model and cognitive processes are design-related, thus facilitating a comparison of simulation results and findings of empirical studies of designers. Therefore, there is a need for development of a design-related representation of the agent’s knowledge that is compatible with described cognitive processes and mechanisms. A convenient way to represent design knowledge is by utilising design ontologies (e.g. [128], [198]).

In this work, the *Function-Behaviour-Structure (FBS) ontology* [128] is used. The FBS was initially developed as an ontology for describing designs but was later shown to be suitable for describing design processes as well [199]. Numerous research efforts across multiple domains (e.g. mechanical and industrial design, or architecture) offered empirical support and demonstrated the applicability of FBS in providing tools to understand both, designing and designs [200].

The FBS framework is presented in Figure 4.3. The *Function (F)* ontological category denotes the purpose of a design object. Components of a design object and relations among these components form the design object's *Structure (S).* Attributes derived from the design object's structure are called *Behaviours (B).* FBS framework makes a distinction between expected behaviours – Be, and actual structure's behaviours (i.e. behaviours derived from the structure) - Bs. Finally, *Requirements (R)* posed on the design and a representation of a design object, *Description (D)*, are added to the framework. The fundamental design processes are denoted with arrows among the design issues (i.e. requirements, functions, behaviours, structures and descriptions). Namely, Gero [128] differentiates eight fundamental processes:

1. *Formulation* - a process during which a set of requirements is transformed into functions, which in turn serve to identify the expected behaviours (B_e). In short, the Formulation process defines the problem space;
2. *Synthesis* – a process of solution generation based on the expected behaviours;
3. *Analysis* – a process of determining the behaviours of the generated structures;
4. *Evaluation* – a process of comparing expected and obtained (i.e. derived from the generated structure) behaviours;
5. *Documentation* – a process of creating the design's description based on the generated structure;
6. *Reformulation type I* – a process of structure space modification driven by solution re-interpretation;
7. *Reformulation type II* – a process of behaviour space modification driven by solution re-interpretation;
8. *Reformulation type III* – a process of function space modification driven by solution re-interpretation and subsequent reformulation of the expectations posed on the behaviour.

The work of Jiang [201] employed FBS ontology in a study of design cognition of industrial design and mechanical engineering teams performing conceptual design activities, and found that FBS design issues captured more than 70% of the observations of a design session. The remaining time was spent on social communication, coordination of activities, and design process management. Similar findings were reported in [202]. The activities not captured by FBS ontology fall into categories of "extra-design activities" or activities not directly related to the current task [201]. Therefore, FBS is found suitable for capturing relevant design processes and is proven to enable studies of design cognition.

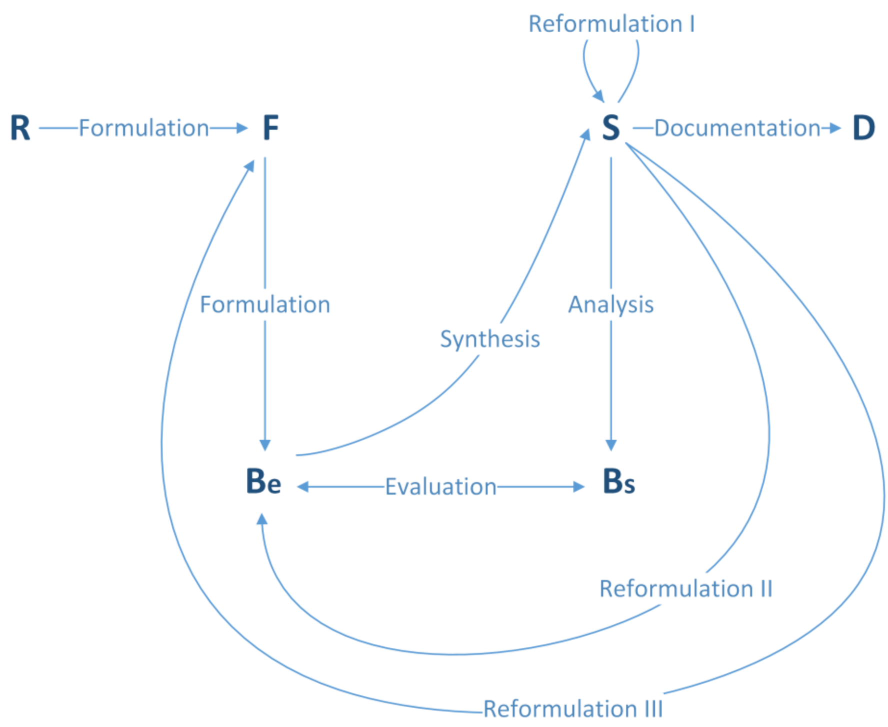


Figure 4.3 Function – Behaviour – Structure (FBS) framework

4.4.2 Affect

As already emphasised, the influence of emotions on decision-making, learning and behaviour regulation has long been recognised [203], [204]. Likewise, many researchers advocate the inclusion of affective elements in cognitive architectures as a necessity for realistic behaviour simulation. Thus, although they were initially developed without taking emotions into account, the prominent cognitive architectures ACT-R and SOAR over time incorporated affective modules: ACT-R's extension was proposed in [205], and SOAR's in [206]. Computational Models of Emotions (CME) are directed towards capturing the interplay among emotions, personality, memory and cognitive processes, and a review of those can be found in [207] and [208]. However, the well-established model of emotions is yet to be developed. Namely, as emphasised in [207], there is no consensus on which emotions are to be considered fundamental, there is a lack of established methodologies to guide the development of CME, and CMEs developed to date build on different theoretical approaches. The majority of CMEs are built on appraisal theories of emotion, such as OCC appraisal model [209] used in WASABI [210] model. Additionally, some CMEs adopt discrete (i.e. hierarchical) [211] or dimensional [212] view of emotions [213]. Most of the CMEs develop detail models of motivation and drives, and how they give rise to emotions experienced.

However, it is not clear if such explicit and detail model of emotions is needed to model design teams, and if so, how it should be related to the design context. The studies of emotions designers experience while designing are seldom. In fact, in their work, Hutchinson and Tracy [214] found that less than 2% of design papers dealing with emotions are referring to designer's emotions (as opposed to the user's emotions). Nevertheless, research [215] indicates that the designer's affective state plays an essential role in many design activities, and in particular, in ideation. Hutchinson [215] found that most of the designers were unaware of their emotions experienced during design. But once their attention was brought to them, the study subjects concluded that affective states are acting as a driving force for their designs. The study [215] reported that designers copped with emotions of anxiety and excitement by seeking feedback, imposing a structure to the process or relying on supervisors help. Sas and Zhang [216] studied the designer's emotions throughout the design process and found different emotions present in various stages of the design. Notably, the authors [216] emphasised the feelings of fear and frustration in periods of disagreement and no progress, i.e. when the deadlock is encountered. Similarly, Chulvi and González-Cruz [217] studied frustration as an emotion experienced during design tasks. Frustration is an emotion experienced as "irritable distress", and results from disappointment caused by inability or uncertainty of goal achievement [218]. Frustration may be caused internally – due to perceived deficiencies in knowledge or capabilities, or externally – as a result of environmental or social barriers to goal attainment. It reduces inhibitions and can result in anger and irrational behaviour. On the other hand, the feeling of frustration can also elicit more effort, which can consequently lead to an improvement [218].

Negative emotional states, such as frustration, nervousness and anxiety, influence the reasoning process by restricting the thought process and narrowing the focus [203]. On the other hand, positive emotions are linked to broader thought processes and greater tolerance for novelty [219]. Researchers note that positive mood is associated with divergent thinking, increasing the chances of obtaining sudden insights - the so called *Aha!* Moments [220]. Hutchinson and Tracey [214], however, found the mixed results regarding emotions and creativity. Namely, in comparison to an indifferent state, both, positive and negative emotions resulted in increased creativity. One explanation for a link among higher levels of creativity and negative affective state may stem from previously emphasised increase in cognitive persistence and effort due to negative emotions [214].

Building on the findings presented, one may conclude that at the task start the designers are motivated and positive, thus exploring the wide space of ideas. As time passes, the anxiety and a sense of urgency force the designers to focus on solution finding and convergence. This behaviour is compliant with the solution space exploration behaviour implemented in agent-based models of design teams such as [221], [222] by using, for example, simulated annealing. The studies of Gersick [223] suggest that at about half-time of the project people start to noticeably change their behaviour as a result of concerns due to deadlines. Although not studying the effect of emotions on designing, Stampfle and Badke-Schaub [202] observed the effect of approaching deadlines on the designer's effort to converge to a solution, noting that if no acceptable solutions were proposed, the designers tend to re-examine the discarded solutions.

4.4.3 Personality traits and cognitive ability

To introduce diversity in the team model, one has to model individual differences such as cognitive ability and personality traits [38]. The *cognitive ability*, or intelligence [224], is often defined as a general mental capacity [225], [226], and it relates to aspects of information processing speed [227], learning [228] and sense-making [226]. Studies [180] suggest that a single construct of *Working Memory Capacity* (WMC) influences an individual's ability to develop, maintain and update bindings, thus correlating with fluid intelligence. Further, research suggests that such working memory capacity differs among individuals [182], [183]. Preceding subsections discussed the 'controversial' nature of such WMC, where, for example, Cowan [183] argues that it ranges from 3 to 5 knowledge chunks focused at once. Similarly, Engle [229] explains that individual differences in working memory span influence one's speed of focus shifting.

Cognitive ability has been related to personality [230] with high cognitive ability often being linked to higher levels of *openness to experience* and *emotional stability* (i.e. lower levels of *neuroticism*), and negatively associated with *conscientiousness*. Listed personality traits are elements of one of the most established personality models: the *Big Five* or *Five-Factor Model* (FFM) of personality [231]. Aside from already listed, the factors comprising the FFM are *extraversion* and *agreeableness*. Based on the findings from several studies, the FFM personality traits are regarded as stable over time [232]. In their studies of impact personality traits have on job performance, Barrick and Mount [232] group personality traits as those contributing to "getting ahead": conscientiousness, emotional stability/neuroticism and openness to experience, and those related to "getting along": extraversion and agreeableness. The latter group of personality traits proves especially important in team settings.

Numerous studies have explored the interplay among cognitive ability, personality traits and individual and team performance (e.g. [232], [233]). Within the design field, an early study was reported in [234], generating several insights and directions for future work. For example, the authors found a significant negative relationship between heterogeneity of conscientiousness and team performance. Later studies [235] explored the influence of average team conscientiousness on design team performance, but no significant relations were found. Others [236], [237] studied the interplay among team personality composition and creativity. However, more research is needed to determine the relations among different aspects of personality and design team performance.

As a starting point in modelling agent's personality, the developed model (see Chapter 5) includes aspects of extraversion and agreeableness, i.e. the personality traits contributing to "getting along". Extraverts are often described as sociable, assertive and talkative individuals [231], who enjoy interacting with others and are not afraid of criticism [237], [238]. Introverts, on the other hand, tend to be reserved in interactions with others and reflective. Baer et al. [237] argue that the level of extraversion influences the likelihood of a designer sharing ideas with the team.

Agreeableness is a trait related to altruism and cooperativeness. Agreeable people are trusting, modest, friendly, and are viewed as supportive and tender-minded [231]. Conversely, disagreeable people can be hostile, argumentative, and seek other's faults [239]. In a design team setting, the disagreeable personality may be manifested as criticism and dismissal of other's ideas [237]

4.4.4 Trust and transactive memory system

Learning through interaction, observation and imitation of others, or what Tomasello [240] refers to as learning *from* and *through* others, is regarded as innate to humans. It is necessarily accompanied by learning *about* others. The perception each team member holds of others' knowledge and beliefs, i.e. the *transactive memory* [241], [242] is shown to have a significant influence on the team performance and behaviours [243]. Therefore, a large number of computational models of teamwork implement a transactive memory system (e.g. [244]–[246]), arguing that it contributes to the bounded-rationality - and thus increases the behaviour realism - of the simulated actors [241].

The perception of others is a basis for *trust* formation [247]. Trust has been defined as "the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party” ([248], p. 712). Salas et al. [32] note that a lack of trust among members of a team leads to putting extra effort into inspecting other’s work, which in turn prolongs the time needed to complete the task and may result in disagreements and conflicts.

Various forms of trust have been studied in the literature. For example, Webber [249] distinguishes *early trust*, *affective trust* and *cognitive trust*. Early trust is a one-dimensional construct build on reputation, prior knowledge of one’s competence, and likability. As familiarity among individuals’ increases, trust can be separated into the cognitive and affective component. The former is based on reliability and competence, while the latter is related to the level of concern and formed emotional bonds. Marsh [250] categorised trust into: *basic trust* – a trust disposition not directed towards anything or anyone specific, general notion of the agent’s propensity to trust; *general trust* – an overall trust the agent holds in a particular team member; and *situational trust* – a trust towards other agent with the respect to the current situation (as perceived by the trustor agent). Listed categorisations clearly emphasise the dynamic and context-dependent nature of trust [247].

Trust is viewed to arise from the trustee’s perceived trustworthiness – which is a function of ability, benevolence and integrity [248], [251], and trustor’s propensity to trust [252]. Mayer et al. argued that people have an inherent propensity to trust [248], rooted in one’s personality. Similarly, researchers [253], [254] have found traits such as agreeableness to be a significant predictor of trust propensity. Future experiences and perceived team performance, shape the further development of both, transactive memory and trust [243], [247]. Literature review on the transactive memory system research [243] revealed discrepancies in how researchers view the relationship between trust and transactive memory system. Namely, while cognitive-based trust is sometimes regarded as an important dimension of transactive memory [255], others have described trust as a transactive memory system’s antecedent [256] or a mediator of the influence of transactive memory system on team performance [257]. The research presented herein, however, does not aim to capture a detailed interplay among these elements. As detailed in the next chapter, the developed model includes only the perceptions agents have on “who-knows-what”, which is based on the task-related competencies and expertise (thus relating only to cognitive-based trust and transactive memory).

5 MODEL IMPLEMENTATION

This chapter presents details of the developed model: first on the agents representing design team members and then concerning their interactions and tasks performed. Similar to Section 4.4, to ease reading and support understanding, the description of modelled team member agent is organised into sections describing the agent's mental model, affect, personality and trust. Each of the specified sections consists of an implementation description and a short overview discussing the limitations of the implementation. The chapter then details communication among agents and describes technical aspects regarding task representation, generation, sequencing and completion. Finally, an outline of a simulation run is presented.

The conceptual model presented in this chapter is implemented using MASON [258], and the adequacy of both, conceptual and computational models, is assessed in experiments reported in the next chapter.

5.1 Team member agent

The team member agent, as indicated by its name, is a computational representation of a single member of a team and represents the most elaborate element of the developed model. In contrast to the majority of agent-based models reviewed in Chapter 3, the developed agent is cognitively rich, affective, and its actions are guided by its previous experiences. Inclusion of multiple aspects of human behaviour in agents serves to achieve the goal of obtaining a multi-purpose tool for team studies. However, models that constitute such detailed agents (as opposed to models where a few simple rules guide agent's actions) have several limitations, most notably difficult verification and validation [259]. In order to ease verification and validation, as well as to enhance flexibility in designing simulation experiments, the agent is implemented in a modular fashion: with the exception of the core elements and processes of the agent's mental model, the effects of the personality, memory and affective aspects can be switched off, therefore enabling isolated studies of a particular element's impact on the overall behaviour. These implementation details are explained in the following sections.

5.1.1 Team member agent's mental model

Following a theoretical background and relevant work presented in the previous chapter, this subsection gives the details on the developed model. First, the agent's mental model is described in light of the design research by using FBS design ontology to model the agent's knowledge. FBS framework is further used to describe the agent's reasoning processes modelled as spreading activation over the knowledge network. Additionally, mechanisms of inhibition, learning and chunk formation are described. Taken altogether, these mechanisms explain the agent's cognitive behaviour on a single task.

Further subsections are dedicated to describing the team member agent's mental model update in between two tasks, and mental model initialisation at the simulation start. Finally, an overview is presented and limitations of the current implementation are discussed.

5.1.1.1 Development of a design-related mental model: Relation to FBS ontology

Based on the FBS framework, the design knowledge can be represented as a network of functions, behaviours and structures, while design processes can be modelled as a traversal over such network. This idea inspired the team member agent's mental model depicted in Figure 5.1. As shown in the figure, the mental model is represented as a multi-layered network. Layers correspond to ontological categories distinguished by FBS framework (i.e. function layer, behaviour layer and structure layer), each of which contains nodes of their corresponding type. Similar to the FBS framework, there are links from function nodes to behaviour nodes, and from behaviour nodes to structure nodes. There are no links among nodes of the same type, nor direct links from functions to structures [260].

Agent's mental model does not include description nodes as the *Documentation* process is omitted from this study (thus forming one direction for future work). On the other hand, requirements had to be included to provide a source of activation. Requirements define the task to be solved and are modelled as a combination of behaviours which a structure has to meet (i.e. the structure has to be linked to the required behaviours) to be accepted as a solution. A particular function node can, therefore, be seen as relevant for the requirement if (and only if) it is connected to one or more relevant behaviour nodes. Although not illustrated in Figure 5.1, links from requirements to function nodes can be established based on the relevance of a function node for the given requirement. In this manner, an activation starting at the requirement can be spread to function nodes, and further passed to behaviour nodes. This activation passing corresponds to *Formulation* process of FBS ontology. The activation can then be forwarded to structure nodes, representing the process of *Synthesis*.

The *Analysis* process could have been modelled as an activation spreading from the "sufficiently active" (i.e. firing) structure node back to behaviour nodes, i.e. the structure node acting as a source. In this manner, the behaviour nodes active due to activation spread from the requirements would be regarded as a set of expected behaviours (Be), while behaviour nodes active due to activation spread from the structure node would generate the Bs set, thus permitting Evaluation process.

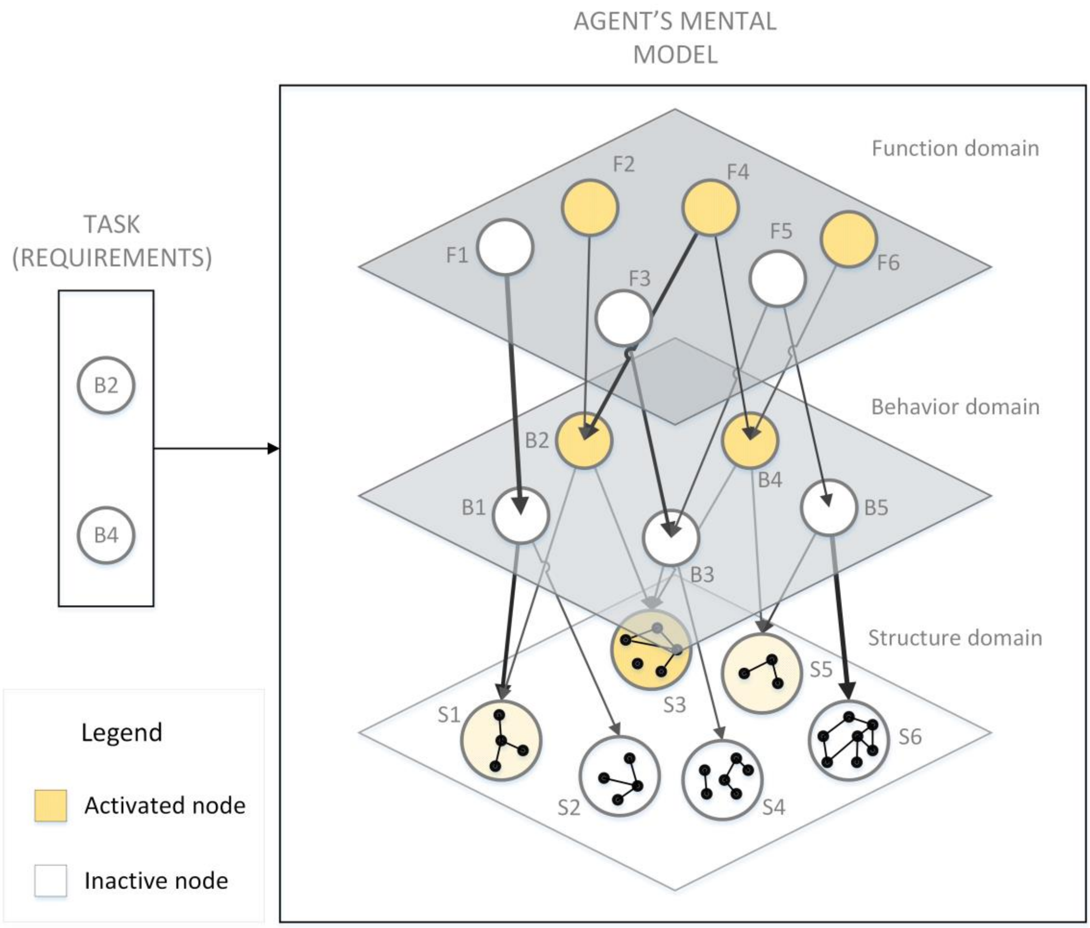


Figure 5.1 Team member agent's mental model [261]

However, such a simple model does not instil a manner in which new structure nodes (and their relations to behaviour nodes) can be created. A straightforward solution is to predefine all of the possible structure nodes and their connections to behaviour nodes, or to implement a random assignment of behaviour-structure links during simulation runtime. But implementing these mechanisms would limit the capability to study solution space expansion and properties. Another approach is to endow each structure node with a structure representation that enables calculation of various properties. For example, associating each structure with a number allows observation of the features of being positive/negative, odd/even, prime/composite or integer/rational. The calculation of such properties simulates the process of the behaviour determination (i.e. Analysis). Further, incorporating representations of structures can provide the means for new structure node generation. For example, one can represent a structure as a vector in an *n*-dimensional space and associate behaviour nodes with a specific orientation, length or direction. Creation of new structures can then be modelled as an application of vector operations (e.g. vector addition or scalar multiplication). Or, one can reuse the idea presented in [78] by associating each structure with a geometric shape, enabling shape overlap (to create new shapes/structures), and regarding each behaviour node as representing the property of – for example – having a specific number of sides or covering a particular area.

A design structure is often regarded as a network of parts (components) [262]–[264], which inspired the structure representation implemented herein. As shown in Figure 5.1, in this work each structure node is associated with a (binary and undirected) network, and behaviour nodes are associated with certain network properties. More precisely, each behaviour node corresponds to a certain range of a particular network property (e.g. the number of components being within range $\langle 3, 5]$). An example of a structure node and its associated behaviour nodes is presented in Figure 5.2.

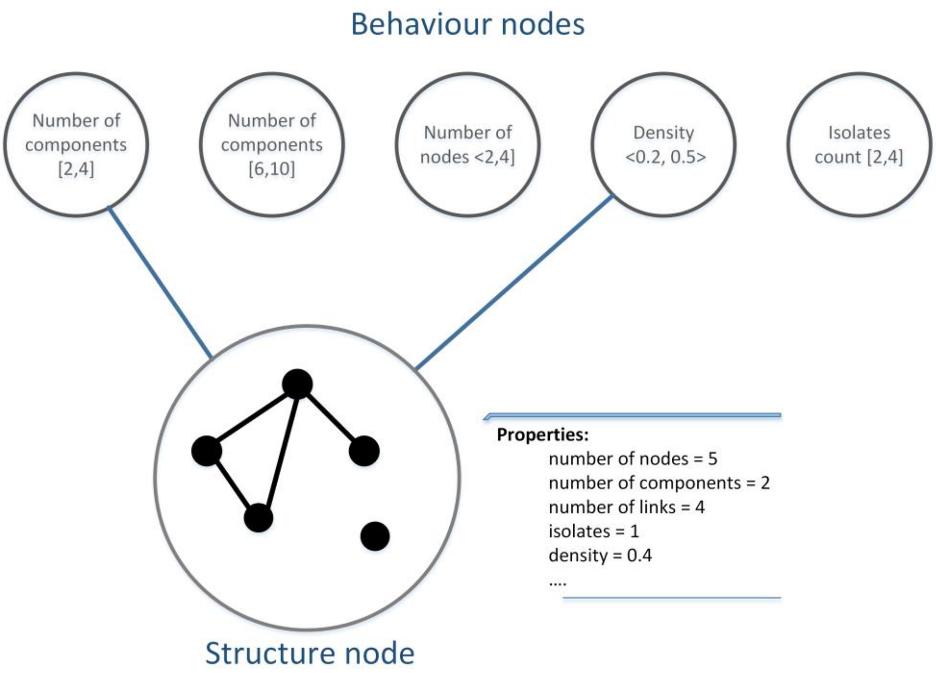


Figure 5.2 An example of a structure node and associated behaviour nodes

Once the structure node's activation exceeds a threshold, an agent can calculate the important network properties (i.e. those corresponding to behaviour nodes that the requirements are posed upon) and determine if the structure node qualifies as a solution. This simulates the processes of *Analysis* and *Evaluation*. If an unmet requirement is detected during the Evaluation process, an agent (re)directs the activation to the relevant function nodes (i.e. function nodes related to the relevant behaviour nodes), thus simulating the *Reformulation III* process. The *Reformulation II* process is not explicitly modelled. Instead, its functionality is achieved by subsequent passing of the activation (redirected through Reformulation III) over the function-behaviour links.

Finally, the *Reformulation I* process occurs through the process of new structure generation. Currently, two structure generation mechanisms are implemented: *union* and *concatenation*. First is triggered each time two structure nodes are simultaneously active beyond a predefined threshold, and it creates a structure node whose respective network is obtained by overlaying networks associated with the two activated structure nodes. This process simulates the act of combining two distinct structures into one, and is depicted in Figure 5.3a. The concatenation process, on the other hand, requires just one sufficiently active structure node and consists of collapsing two of structure node network's nodes (i.e. "parts") into one. It is a process inspired by the act of combining two parts of a structure into a single part (e.g. coupling a mobile phone's screen and a keyboard into a touchscreen). When a structure node is active, its associated network's links are scanned to determine if any can be replaced. This process is stochastic, and it is guided by the frequency of the link's occurrence in networks of all of the structure nodes known to the agent. More precisely, the precondition for concatenation is that a sufficient number of distinct structure nodes contain the link to be replaced. If this precondition is met, a ratio of the number of known structures in which the link exists, and the number of known structures in which the parts to be collapsed are not linked determines the likelihood of a replacement occurrence. The concatenation process is shown in Figure 5.3b. Only one combination and/or concatenation can be performed in a single step.

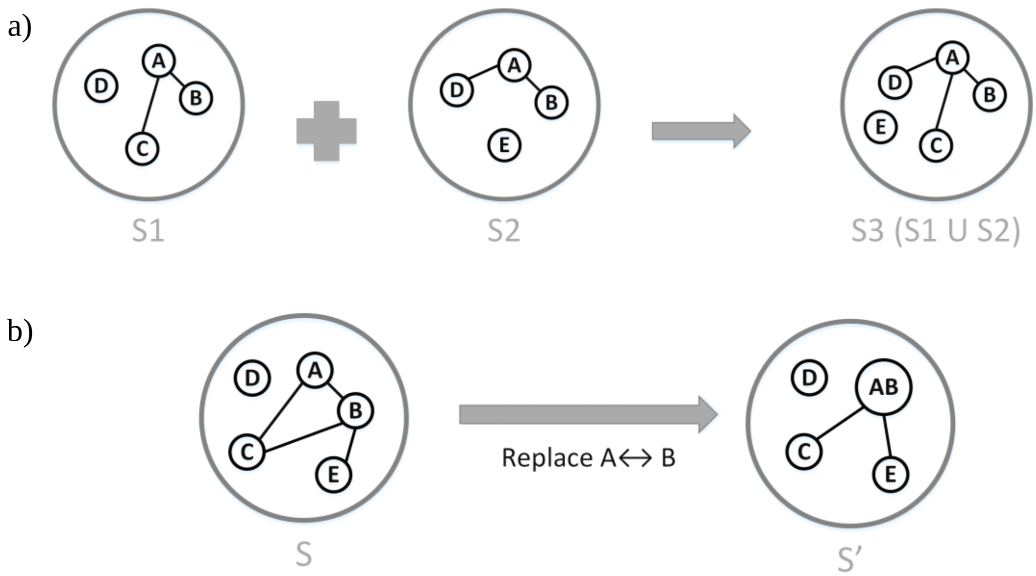


Figure 5.3 Generation of new structures a) The union process b) The concatenation process

Irrespective of the generation mechanism applied, the properties of the resulting network often differ from the original network's properties - which suggests that a new network can display a previously unseen combination of properties. On the FBS node level, this means that a new structure node can be connected to a set of behaviour nodes different (or even disjoint) from the set(s) of behaviour nodes related to original structure node(s). Thus, the defined generation processes equip team member agents with the mechanism to find a solution even if the structure with required behaviours was not previously known to any of them.

It is important to note the effect of the interplay between the two generation processes. While union results in a network whose size (i.e. the number of nodes and links) is larger than or equal to the original networks, concatenation reduces both, the number of links and the number of nodes. This enables agents to generate structures of a wide range of behaviours. More importantly, the inclusion of both, union and concatenation generation processes renders the solution space as unbounded in the sense that agents can always generate new structures. Since the structure node networks are binary, the union of a network S1 and its subnetwork S2 results in S1. Thus, if the only available mechanism were the union, the agents would eventually exhaust the set of possible combinations. Similarly, applying only the concatenation process would subsequently lead to networks with no links, and no new structures could be generated. However, if an initial space is sufficiently rich, one can generate an infinite number of structures by utilising the processes of concatenation and union. It can easily be shown (see Appendix B) that an initial space containing just one structure node whose network consists of two links with an intersecting node is sufficient to generate – through union and concatenation - an unbounded solution space. The unboundedness of the solution space was ensured in every simulation study performed by utilising the model presented herein.

5.1.1.2 Additional processes of activation regulation

The described processes of activation spread over the links and activation redirection due to encountered discrepancies are in line with the FBS design processes and serve to model the working memory mechanisms. As noted earlier, researchers [179], [229] regard working memory as containing those long-term memory elements that are active above a certain threshold. Similar to the stance taken in ACT-R [170] architecture, in this work, the accessibility of nodes is correlated with their activation. Highly activated nodes are accessible to conscious awareness, meaning that they can be communicated and reasoned upon. In this work, two activation level thresholds are introduced: the first, lower one, guides which nodes enter the consciousness and can be verbalised. This threshold is thus called the *activation* *threshold*. However, not every node that an agent is aware of will be reasoned upon in detail. Thus, the second, higher threshold guides which nodes can be analysed (i.e. *analysis threshold*). For example, a structure node network's properties can be analysed only when the structure node's level of activation exceeds the second threshold. Similarly, reasoning upon which function nodes are relevant for the task at hand can be initiated only when the requirement's level of activation meets the analysis threshold. As described, initial activation impulse is generated in the requirement node (i.e. the task), and is then spread over the links in the agent's mental model. If, at any time, the activation disappears (in a sense that activation of every node is below the activation threshold), the activation of the relevant requirement node is reinitiated. The (re)activation of requirement nodes simulates the prefrontal cortex's function of active goal maintenance by (re)directing the attention towards goals [265].

Working memory has a crucial role in *attention switching* and (intentional) *inhibition* of irrelevant or unsuitable information [169]. The attention switching is included as previously described and follows FBS processes. But additional mechanisms had to be included to simulate intentional inhibition of unsuitable nodes. Namely, if a structure is deemed as non-satisfactory, although it might be the best structure found so far (i.e. the one satisfying the most requirements), the agent can intentionally inhibit it for a certain period to allow further search for structures which would meet all of the requirements. The inhibition is implemented as a gradual increase in a negative impulse (i.e. the impulse subtracted from the activation impulse), at the rate proportional to the number of requirements that the structure fails to satisfy. Since inhibition simulates a voluntary process, it can stop the agent from analysing a particular structure (i.e. inhibition caused the activation to drop below the level of activation needed for the analysis). However, inhibition cannot decrease the level of activation enough to cause the agent not to be aware of the node. In other words, the maximal level of inhibition still permits the node to have an activation level above the activation threshold.

Finally, to equip team member agents with the capability to change their behaviour based on the experiences, the *learning* mechanisms had to be implemented in agent's mental model. Building on the ideas presented in [153], [170], the weight of the links in agent's mental model increases by the links' use. The link weight influences the amount of activation that can be passed along the link in a single step, thus representing the ease of the links' processing. This is indicated by the differences in links' thickness in Figure 5.1. As the nodes' activation exceeds the analysis threshold, the links associated with the node can be processed and increased in weight. Following the proposal presented in [266], the link weight increase follows a sigmoid curve. Namely, the weight is updated following the logistic function of the form:

$$w(x) = \frac{1}{1 + e^{-(x-5)}}, \tag{5.1}$$

where $x \in [0, 10]$, meaning that weight of the link does not exceed the value of 1 (i.e. $w(x) \in \langle 0,1 \rangle$). Each weight update is guided by the formula $w(x_i) \rightarrow w(x_{i+1})$, where $x_{i+1} = x_i + c$, and c is a predefined constant. If the link does not exist within the agent's mental model, its weight is regarded as being equal to zero.

To align agent cognitive behaviour with the prominent theories of human thinking, a link weight threshold (that separates "grounded" from "ungrounded" links) is introduced. The link whose weight is above the *reflex threshold* is regarded as grounded and can be processed at almost no cost - thus simulating System 1 thinking [194], or reflexive reasoning [196]. More precisely, links whose weight does not exceed the threshold require (at least) one step to transfer enough activation for receiving node to become active. On the other hand, grounded links can pass activation immediately (i.e. as a reflex), thus allowing the receiving node to forward the activation in the same step. For example, if a link from function node $f \in F$ to behaviour node $b \in B$ is grounded, and node f is active, the activation can pass from f to b and then be further passed towards structure nodes linked to b in just one step. The presented interpretation of System 1 and System 2 thinking in the FBS context nicely aligns with Kannengiesser's and Gero's studies on design thinking through the lenses of Kahneman's dual-system theory [195], [267].

The learning (i.e. link grounding) is closely related to expertise gain and knowledge *chunk* formation [186], [268]. Namely, nodes connected with well-grounded links can be regarded as a chunk that can be processed as a single knowledge unit. As stated in [195], System 1 thinking (i.e. processing of links within a chunk) does not use working memory resources. In other words, when modelling working memory capacity, elements forming a chunk should be regarded as occupying a single "slot". This work builds on Cowan's [183] view of working memory, thus assuming the working memory capacity to range between 3 and 5 knowledge chunks that can be focused (or "fire") in one simulation step. Within this work, the size of one chunk is not limited so chunks can grow with experience. However, the assumption of unboundedness of chunk size may create unrealistic situations in which large number of nodes is active. This aspect of the implemented model is further explored in the next chapter, but additional refinement is needed to implement theoretical implications in more detail.

Although the view in which well-connected entities form a chunk has been frequently utilised (e.g. [268]), it is not evident how this notion should be translated to the model at hand. For example, if a structure node S is the most active node, and it is tightly connected to a node B, which in turn shares well-grounded links to several other structures - do all of them form a chunk? As currently implemented, the process of extracting the agent's focus (in one simulation step) follows a simple algorithm (Figure 5.4). First, nodes active above the activation threshold are collected and ordered by the amount of activation. Starting from the most active node, the immediate grounded surrounding (i.e. one grounded link apart) of the node is extracted and organised to form a chunk. If two chunks intersect, they are combined into a single chunk. This process is continued until the working memory capacity is reached, or no active nodes are left.

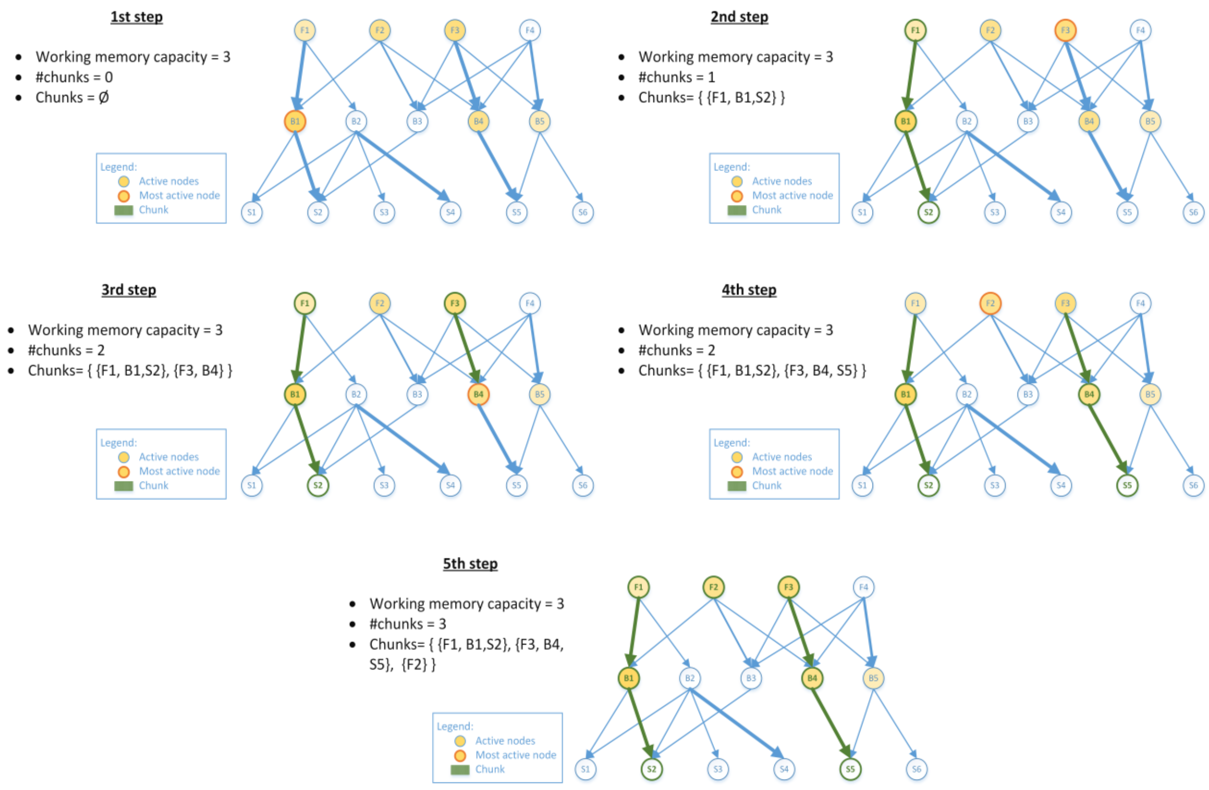


Figure 5.4 An example of extraction of nodes focused in a single step

Once the focused nodes are extracted, they fire, thus spreading their activation through the network. The firing nodes on the receiving end of the grounded links can pass on their own activation as well as the activation from the focused nodes on the other end of the grounded links. The spreading activation follows the formula [153]:

$$A_i(t) = A_i(t-1) * (1 - decay) + \left(\sum_{\substack{j=0, \\ A_j \in focused}}^{k} A_j(t-1) * w_{j,i} \right) \big(1 - A_i(t-1)\big), \quad (5.2)$$

where k is the overall number of nodes, $A_i(t)$ denotes the activation of the i-th node in the step t, *decay* is a predefined factor, and $w_{j,i}$ is the weight of the link from j-th node to the i-th (and zero if such link does not exist). Note, only focused nodes pass their activation to other nodes.

The described processes guide team member agent's behaviour on a single task. A summary of the agent's cognitive behaviour in each simulation step is presented in Figure 5.5.

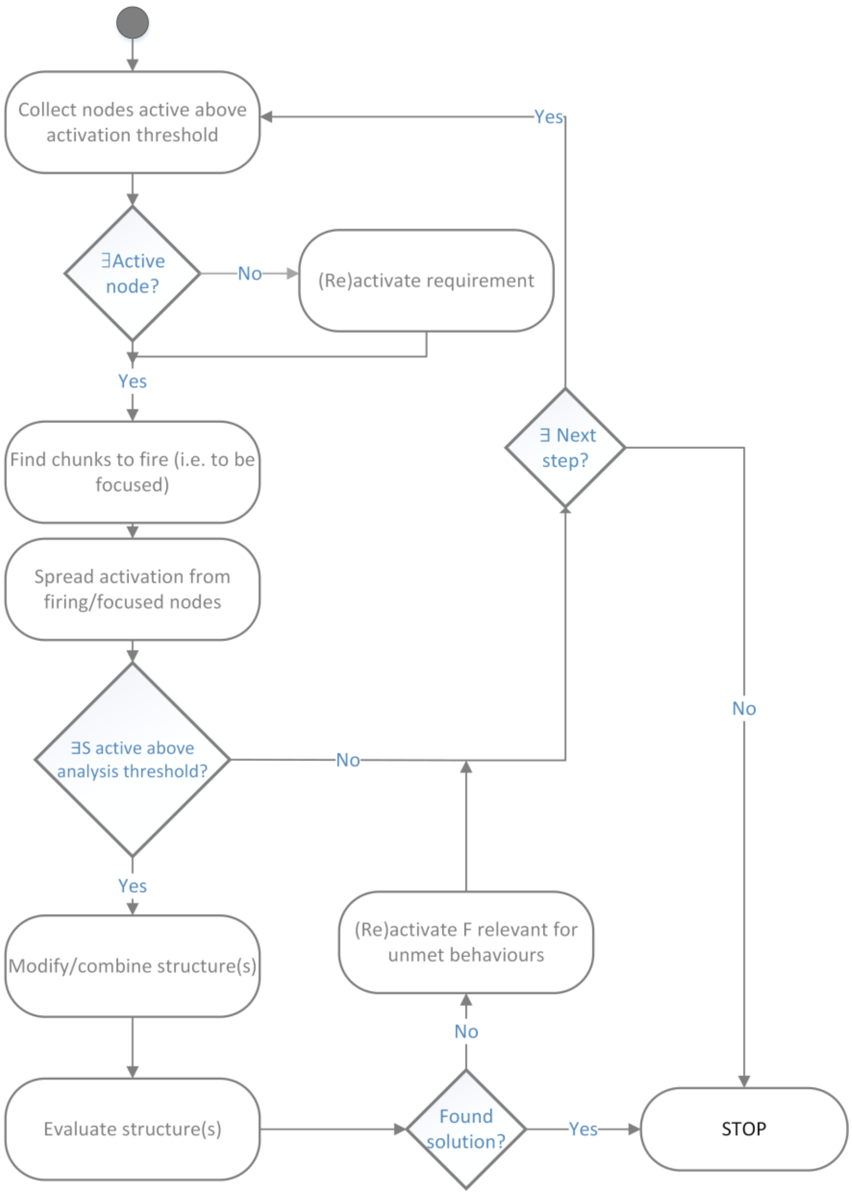


Figure 5.5 Agent's cognitive behaviour (on one task)

5.1.1.3 *Mental model update over multiple tasks*

The previous subsection describes agent's cognitive behaviour on a single task. To model behaviour over multiple tasks additional mechanisms had to be included. Namely, a one-task simulation is intended to simulate several work hours, during which no significant forgetting is expected to occur. But if one wishes to study the change of the mental models over several tasks, it is necessary to implement knowledge decline, as well.

Therefore, two levels of link weight update are implemented. The first one – temporary change occurring during a single task - is implemented as defined previously. To model learning and forgetting occurring between tasks, the new weight of each link in the agent's mental model is based on the difference between the weight at the start of the previous task and the weight at task end. Namely, if there is no change, the link has not been used in the task. Therefore, its weight declines following the formula:

$$w_{start}(n) = forgettingRate \ * w_{start}(n-1), \tag{5.3}$$

where *n* denotes the ordinal number of the task in a task sequence, and $w_{start}(n)$ initial weight of the link (i.e. weight of the link at the start of n-th task). If, on the other hand, the weight has increased during the past task, new weight is calculated as:

$$w_{start}(n) = w_{start}(n-1) + learningRate * \left(w_{end}(n-1) - w_{start}(n-1)\right). \tag{5.4}$$

where $w_{end}(n)$ denotes the final weight of the link *w*, i.e. the weight at the end of the *n*-th task.

The described processes are sufficient to model reasoning and knowledge change, and thus they form the core mechanisms of the agent's mental model. Additional methods are included to provide a richer set of possibilities when designing simulation experiments and, as noted earlier, can be easily added to or excluded from simulation studies. One such method is intended to account for the effect of memories of the past tasks. Namely, similarities of the preceding tasks and current situation may cause specific nodes to be easier to evoke, or may tie negative affect to the node (e.g. due to failure of the past task), thus causing one to avoid verbalising or considering it in the current task. The implementation of this mechanism is inspired by the *base-level activation* in ACT-R architecture. However, in this model, it serves a different purpose. In ACT-R each node's base-level activation accounts for the context-independent ease of activation (due to recency and frequency of the node's use). Here, the effect of recency and frequency of use is captured by the weight of the links. Base-level activation, on the other hand, serves to take into account the "context" formed as a result of memories. In short, base-level activation serves to capture the tendency of a node to be activated due to similarities between past situations (in which the node was deemed as important and thus may have emotional tone tied to it) and the current task.

To implement the described mechanism, each agent memorises the past tasks in the following manner: for each task, a set of nodes that were focused (and, therefore, deemed as relevant) during the task is remembered. However, not each node is equally important/easy to remember. The importance of a node for the task is approximated by the number of times it was focused, while the node that was chosen as a final solution is deemed as the most important. The number of times the nodes were focused (i.e. the nodes' importance) is normalised so that the nodes' importance is within range $[0.2, 0.8]$, while the most important node – final solution – gains the importance 1. One consequence of such implementation is that different agents may have different recollections of the previous experiences. At the start of the task, current requirements are compared to the past, remembered tasks. The similarity of the past tasks and the task at hand, as well as the importance of the nodes for the past tasks, guides the base-level activation of each node. Similar to the SOAR's implementation [171], the default number of tasks an agent can remember - *maximal number of past tasks remembered* - is set to ten.

Formally, a task can be defined as a set of required behaviour nodes:

$$T = \{b_k \mid b_k \in B\},$$

where *B* is a set of all behaviour nodes. Then, a similarity between past task T_i and current task T_j can be calculated by – for example – the Sørensen–Dice coefficient [269], [270], Jaccard index [271] or overlap coefficient [272]. In the current implementation, Sørensen–Dice (or Dice for short) coefficient is used. Then, a base-level activation (BLA) for a particular node $node_k$ in the task T_j is obtained following the equation:

$$BLA(node_k) = \frac{\sum_{i=1}^{j-1} Dice(T_i, T_j) * (j-i)^{-1} * \ importance_i\,(node_k)}{\sum_{i=1}^{j-1} (j-i)^{-1}}, \tag{5.5}$$

where $importance_i(node_k)$ denotes the perceived importance of the node $node_k$ for the task T_i. The value obtained following the proposed formula can further be scaled by a parameter (*maximumBLA*) to set the maximum value (and, thus the effect) that base-level activation can have. At each simulation step, a node's activation equals its base-level activation, plus the activation obtained by spreading (or activation redirecting, as described earlier).

Further, since this process is aimed to account for the affect tied to a node, it also serves to set the initial inhibition value for each node. Particularly, if the node is a structure node proposed as a final solution in a task T_i, or if it is a behaviour node linked to the final solution, then its initial inhibition is set based on the success of the task T_i:

$$inhibition(node_k) = \left| minimum\left(\frac{\sum_{i=1}^{j-1} Dice(T_i, T_j) * (j-i)^{-1} * \; success(T_i)}{\sum_{i=1}^{j-1}(j-i)^{-1}}, 0\right)\right|, \quad (5.6)$$

where $success(T_i)$ equals -1 if the task T_i was unsuccessful, and 1 otherwise. Similar to base-level activation, inhibition can be scaled with a factor ($maximumInitialInhibition$) guiding the maximal value that initial inhibition can take. While base-level activation remains unchanged during the task, inhibition value can be changed (both, increased or decreased) due to subsequent reasoning on the node's utility for the task.

5.1.1.4 Mental model initialisation

The remaining aspect of the team member agent's mental model, which needs to be explicated is its initialisation – how to create a mental model if no previous experiences are obtained (i.e. no tasks performed).

For this purpose, each agent is assigned a *domain of expertise* (or expertise, for short). First, the set of behaviour nodes is created by selecting several properties of interest (i.e. network measures), dividing the possible values of each property into a finite set of disjoint intervals, and, finally, assigning each property range to a distinct behaviour node. Then, several of the created behaviour nodes are selected to form the agent's domain of expertise and are included in the agent's initial mental model. Function nodes are created and randomly connected to 3-5 behaviour nodes. Function nodes that are connected to the behaviour nodes within agent's expertise are added to the agent's initial mental model, and the weight of function-behaviour links is set to be higher than mean link weight value (i.e. above 0.5). Finally, structure nodes whose network properties fit within the range of agent's expertise are created and added to the initial mental model (again, with well-established links to behaviour nodes). In this manner, a portion of knowledge related to a specific set of behaviours is created – thus forming the agent's expertise. Knowledge (i.e. nodes and links) outside agent's expertise area may also be added to the agent's initial mental model. However, the weight of non-expertise links is set to be below 0.5.

The domain of expertise may serve to set base-level activations if the agent has no other experiences (no past tasks). If no tasks were yet performed, the similarity between the agent's domain of expertise and the current task is calculated, and base-level activations of expertise-relevant nodes are set accordingly. In this case, the importance of nodes (for the expertise-related nodes) is calculated based on the average strength of their connections to behaviour nodes within the expertise domain.

Agent's domain of expertise may serve as an indication of its role in a task. Therefore, additional, optional, aspect of the implemented model modifies the agent's priorities based on the parts of the task that are closest to its expertise. In other words, it is assumed that each agent will be preoccupied with those aspects of the task which are the most similar to its domain of expertise. The agent's *task-based profile* serves to calculate agent's priorities (i.e. tendency towards resolving certain requirements). Namely, if the set of network measures upon which the current task's requirements are posed is denoted with *requiredMeasures*, and for each measure at least one interval (which corresponds to one behaviour node) *rB* is required, then an agent's profile is defined as:

$$profile \in \mathbb{R}^m \bullet m = |requiredMeasures|$$

$$profile^i = \frac{Dice\left(expertSet^i, requiredSet^i\right)}{\sum_{j=1}^{m} profile^j} \; where \tag{5.7}$$

$$expertSet^i = \left\{b \in expertise \,\middle|\, measure(b) = requiredMeasure^i\right\} \; and \tag{5.8}$$

$$requiredSet^i = \left\{rB \in B \,\middle|\, measure(rB) = requiredMeasure^i\right\} \tag{5.9}$$

In other words, the profile is calculated by determining the similarity between a set of required behaviour nodes and a set of expertise behaviour nodes, for each of the network measures upon which the requirements are posed. Again, Sørensen–Dice coefficient is used to calculate set similarity, although alternative coefficients can be used as well. The priority of a particular measure i (i.e. $profile^i$) is proportional to the similarity of required and expert node-sets, and the profile is normalised so that $\sum_{i=1}^{m} profile^i = 1$.

The profile, thus, indicates the amount of interest the agent is committing to each of the required properties (i.e. network measures). This is manifested through the amount of activation spread from the requirement node to function nodes (where each function node gains the amount of activation proportional to the priority of the behaviour nodes connected to it), and through the structure evaluation process (where fulfilment of a requirement contributes to the overall structure's score in proportion to the perceived relevance of the requirement). The agent's profile guides the order of resolving the requirements, as it is more likely that the agent will focus on the prioritised requirements first. Note, however, once a certain requirement is satisfied, the activation is sent only to the unmet requirements (Reformulation III process). Such activation is again divided based on the agent's profile, but - once the most important requirements are resolved - the requirements of minor importance will also gain attention.

5.1.1.5 Overview and limitations

The developed team member agent's cognitive model builds on the work on cognitive architectures by incorporating several important aspects in a design-related mental model, and manages to capture different modes of human thinking. In other words, the developed mental model brings together previous work and theoretical research presented in Section 4.4.1.

The key aspects of the implemented model can be summarised as follows:

1. In contrast to the majority of existing agent-based models of teamwork in product development, **the developed agent mental model is design-related** – thus providing a means for comparison of the empirical studies' results and agent's behaviour. On the other hand, the level of abstraction in an agent's knowledge representation permits studies of artificial designers or design teams without overburdening the simulation with technical details of a particular design task.
2. The team member agent's mental model builds on the existing cognitive architectures and theories of human thinking to simulate reasoning processes, knowledge acquisition, grounding and forgetting. These processes render the mental model as dynamic, consequently enabling **the change of agent's behaviour based on past experiences**.
3. **The solution space the agents search is unbounded** (i.e. agents can always create new structure nodes). Past experiences influence the search process, and the agent's actions dictate which structures can be generated in subsequent steps. The developed implementation permits studies on the structure space's expansion and properties across tasks.

Although the listed aspects match the characteristics of the desired model (as laid out in Chapter 4), the described agent's mental model has several limitations that should be mitigated in the future model refinements. Several shortcomings were already emphasised, such as the fact that the developed model does not capture all of the relevant aspects and processes of the FBS framework. Namely, Reformulation II is not included directly, and the Documentation process is omitted. Inclusion of these processes would require further research and model extension. For example, if one wishes to include sketching process [273], studies on what triggers sketching, how do designers chose what is to be sketched, how does observing a sketch affect the observers' mental models, how does producing (and observing) a sketch influence the mental model of the person sketching, or how long does the process take would have to be considered. Relatedly, as emphasised in [175], processing of visual input may be different in terms of speed and capacity than the processing of, for example, audio input or a (nonverbalised) thought. Manual actions are also likely to demand certain processing effort. How these actions would be best integrated into the current model remains a direction for future work.

Further shortcomings of the current model are related to the representation of the requirements. As implemented, the task is defined as requirements posed on the behaviours of the solution. However, a task may consist of functional requirements or requirements posed on a structure, as well [274]. Thus, future studies will explore, for example, how to model functional requirement fulfilment.

Limitations of the developed model may also be regarded from the perspective of cognitive science. Following the discussion on the nature of the working memory capacity, a refinement of the current model may consist of limiting the amount of overall activation present in the network at any given time. In the developed model, the activation of each node is limited, but no limitations on the number of active nodes (and, thus, on the overall amount of activation) are posed. Prominent cognitive architecture, SOAR [171] and ACT-R [170], limit the source activation and model activation spreading as a division (in proportion to the link weight) of the source activation among connected knowledge units. The activation obtained by each receiving node gets smaller as the number of connections the source has increases, which is known under the name *fan effect*. However, the inclusion of fan effect in the current model was not convenient due to the disproportionally large number of nodes in the structure layer in comparison to function and behaviour layers of the mental model. Such disproportions emerge because agents are capable of creating new structure nodes, while the number of known behaviour and function nodes increases only through communication - as will be further discussed in following subsections. Dividing activation from one behaviour node to dozens of connected structure nodes would result in an insignificant amount of activation obtained by each structure. On the other hand, the implemented mechanism permits a single, well-connected note to transmit a sufficient amount of activation to activate a large number of nodes. Although the resulting situation is potentially unrealistic (in a sense that an agent is "aware" of a large number of nodes), the subsequent process of focus selection limits the agent's reasoning processes to a specific number of chunks. Nevertheless, future studies should be targeted at the refinement of the mechanisms related to working memory capacity modelling.

Another cognitive aspect not included in the present model is *habituation* – a phenomenon of a decrease in a response to stimulus as a consequence of repeated exposure to it [275]. In the model, node activation indicates the level of attention directed towards the node (i.e. the response is always proportional to the amount of activation), irrespective of the frequency and/or magnitude of past activations. Similarly, it may be discussed how well the false-memories (i.e. *Deese-Roediger-McDermott*[276]) effect is captured. A limitation of the model is the fact that many of the cognitive phenomena rely on the associative links and similarities among concepts that are currently not modelled. For example, reasoning upon one structure may evoke another one due to similarities among their parts (rather than through spreading activation from nodes corresponding to network properties, i.e. activation due to similarities among network properties). In ACT-R [189] this is modelled by inclusion of activation obtained from *partial matching* (along spreading activation and base-level activation) which accounts for similarities among concepts. It is unclear, however, how such a notion should be applied to similarities among design issues (i.e. functions, behaviours and structures).

Finally, each of the implementation decisions presented in this subsection (i.e. Section 5.1.1) has its shortcomings. The agent's solution-grading mechanism, for example, is oversimplified in a sense that a score assigned to each structure depends solely on whether the structure meets the requirements or not, and how confident the agent is in relevant behaviour-structure links. If the agent identifies two structure nodes as solutions, and is certain (i.e. the links are grounded) in its assessment, no additional comparison mechanisms are provided. In reality, it would be possible to compare the structures based on, for example, their optimality and creativity. Thus, future model refinements will necessarily be directed at equipping agents with novelty assessment mechanisms that will – together with already implemented utility assessment – form the basis for agents to determine the creativity level of each solution.

5.1.2 Team member agent's emotional state and affect

It can be argued that in the developed model, the motivational aspect is implicitly included. Namely, if the motivation is assumed to be correlated with the level of attention, the amount of activation present in the agent's mental model may act as motivation proxy. However, the impact of negative emotions such as fear due to uncertainty of goal achievement, or frustration due to lack of progress is not included in the core elements of the agent's mental model. Thus, the affective component described in this subsection adds the impact of negative emotions. For simplicity, the negative emotional states of anxiety, uncertainty, fear, anger and frustration are unified under the name *frustration*.

Following the theoretical background laid out in the previous chapter, at the simulation start, the behaviour of agents is unaffected by negative emotions. This period relates to what Norman

[203] calls *creative stage*. However, as the simulation unfolds and no progress is made, the frustration factor starts to rise and causes the agents to try to converge to a solution. The amount of negative affect experienced is assumed to be correlated with the probability of failure. Thus, the equation from [57], originally used to model the probability of not meeting the customer's demands, can be utilised:

$$frustration(t) = 1 - e^{(c_1 * t_{left} - c_2) \times (1 - score_{best})}, \tag{5.10}$$

where $frustration(t)$ marks the level of frustration in a time (i.e. step) t, t_{left} is the amount of time left, i.e. $t_{left} = \frac{T_{deadline} - t}{T_{deadline}}$ where $T_{deadline}$ represents the maximal time – or number of steps – the agents have to solve the task (i.e. *task duration* - number of simulation steps allocated for the task), $score_{best}$ is the score of the best-performing (as rated by the agent) structure proposed so far, and c_1 and c_2 are constants.

To simulate an increased effort to reach the solution, the amount of frustration obtained by the presented equation is added to the overall activation of the currently active or remembered structure nodes. In other words, if the affective component is enabled, at each time step past the pre-specified time-point, the agent calculates its level of frustration. If no structures are currently active above the analysis threshold, the impulse equal to the frustration level is added to each of the structures that were either focused within the current step, or were proposed during the task. If at a particular step, a structure becomes sufficiently active to be analysed, the frustration effect is temporarily suspended to simulate agent's focus on the structure's analysis. If the structure is deemed as unsuitable, the frustration level is again taken into account (see Figure 5.6).

Since the findings reported in [223] suggest that irrespective of the task duration, the 50% of the time allocated marks the change in the team behaviour, $\frac{T_{deadline}}{2}$ was taken as a default time at which frustration level can rise above zero. In other words, the default value for the parameter $t_{left\_\min}$ is 0.5 (see Appendix C). When this requirement is added to the equation, it easily follows that $c_1 = 2c_2$.

To specify both constants c_1 and c_2, additional requirement has to be set. The default setting is obtained by modelling that when 90% of the simulation is done ($t_{left\_\max} = 0.1$) and none of the requirements is satisfied (i.e. $score_{best} = 0$), the agent acts by randomly analysing structures (i.e. $frustration(0.1) = analysisThreshold$). However, other configurations may also be chosen, which enables simulation of variations in agent's capability to cope with stress, i.e. its level of neuroticism. The general equation is calculated as:

$$frustration(t) = 1 - e^{\frac{t_{left} - t_{left\_min}}{t_{left\_max} - t_{left\_min}} \times (1 - score_{best}) \times \log(1 - analysisThreshold)}, \tag{5.11}$$

(where *log* stands for natural logarithm) for cases where $t > t_{left\_min}$, while equals zero otherwise.

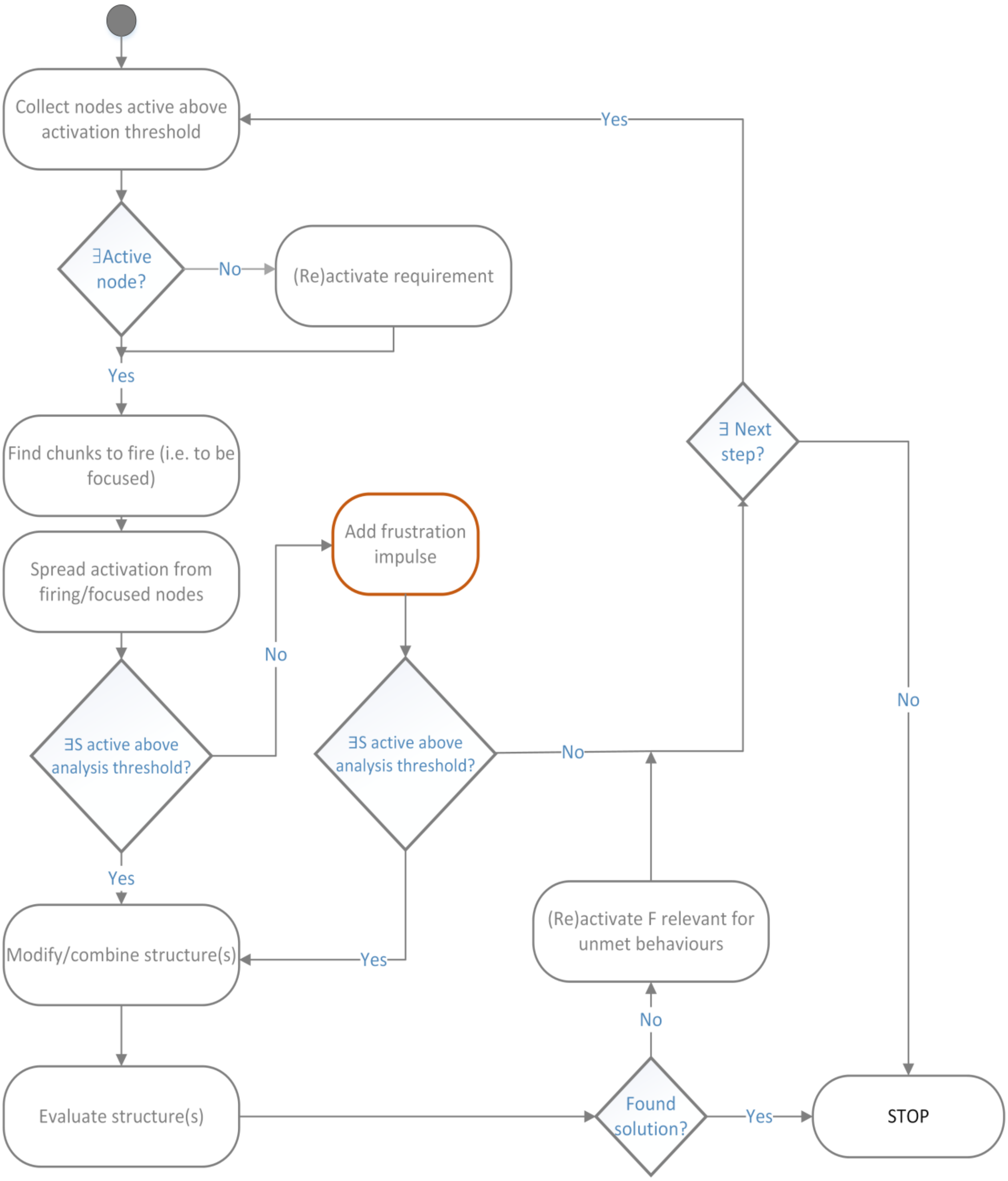


Figure 5.6 The inclusion of affect in agent's mental model

5.1.2.1 Overview and limitations

As described in Section 5.1.1.3, the memories of the past failures may cause the agent to (perhaps unreasonably) avoid building upon a certain structure or behaviour nodes. Similarly, the frustration introduced in the previous subsection modifies the agent's reasoning processes. The implemented model of frustration:

1. relates the level of frustration to the perceived team performance by modelling an increase in frustration when deadlocks are encountered [216];
2. **biases the agent's cognitive behaviour** by directing the agent's focus towards (possibly unsuitable) structures – thus simulating an increase in cognitive persistence and effort [218], but also potentially preventing the agent from sufficiently analysing the problem and exploring the solution space.

The simple mechanism implemented in the model captures the notion of designers' tendency to re-examine discarded solutions when pressured by deadlines and no suitable alternatives have been found [202]. Namely, the frustration impulse is taken into account only after a pre-specified period of "undisturbed" reasoning has finished, and it simulates the designer's effort to focus on the structures in order to find a suitable one. It can be noted that the presented implementation does not pose any further constrains on the problem and solution space exploration: e.g. agents can start to explore the solution space immediately if the problem is well-understood, can generate either large number of alternative solutions or focus on the single one, and can shift between the problem and solution space in all stages of the task.

Since the same amount of activation is added to all of the structures that were either discussed by the team or recently focused by the agent, those most active in agent's mental model will be brought to the agent's attention first. This process may cause the agents to re-evaluate the previously discarded structures and determine that they were, in fact, suitable as solutions: during the evaluation, the relevant behaviour-structure links may be further grounded which influences the agent's assessment of the structure. On the other hand, if the structure is rated as insufficiently good, the activation will be passed on to the unmet requirements, thus focusing the agent's reasoning towards the relevant behaviour nodes. In certain situations, this may cause agents to completely focus on a particular set of (unmet) requirements, repeatedly discussing the relevant knowledge links, or fixating a set of insufficient structures, thus generating what Norman [203] refers to as *tunnel vision*.

Additionally, a possible extreme situation is the one in which many unsuitable structures were proposed and/or focused, but all of their activation levels are similar. In this case, all of these structures will become active at once, causing the agent to select among them randomly. This behaviour, however, is in contrast with the claims that negative emotions narrow the focus and reduce the working memory capacity [277]. Another way of implementing the effect of negative emotions is by modifying the spreading activation algorithm. Particularly, a frustration level may serve as a variable indicating which links should be dismissed: i.e. all of the links whose weight is below a certain value can be (temporarily) discarded and thus omitted from the spreading activation process. This mechanism would simulate the agent's effort to inhibit unrelated thoughts and concentrate on the well-grounded knowledge, rather than exploring the loose connections and broad range of ideas. Future studies will study the differences, advantages and limitations of the two (i.e. implemented and proposed) mechanisms.

The future work should also address the individual differences in coping with negative emotions. Since it depends on the individual's perception of the team progress, the current mechanism enables the agents to differ in the level of frustration. However, the studies from psychology note that individuals cope with stress differently, with some being more prone to anxiety and neurotic thoughts [231]. Chulvi and González-Cruz [217] found that – irrespective of the design task solved or method used – personality has a significant impact on the level of frustration experienced during designing. Such differences may be included by varying the amount of time needed for the agent to start experiencing the frustration impact, and by varying the time at which the frustration level reaches the maximum (equating 1.0 activation impulse).

The described model of emotions is rather simple: the agent's affective state is summarised in one variable and arises as a response to a situation. Such model of emotions is frequently implemented and, due to its simplicity, has the advantages of being easy to develop, use and understand [278]. But additional work is needed to achieve a more realistic affective behaviour. For example, researchers [214], [216] found various emotions dominating different phases of the design process. How to include them in the current model, and simulate the interplay among these emotions [279] remains an open question. Similarly, the effect of other team member's emotions on the agent's affective state should be considered.

Finally, more thought should be dedicated to modelling the agent's initial motivational state. Hutchinson and Tracey [214] note the importance of the initial emotional sub-tone tied to the design task. Currently, the developed model only includes negative emotions related to specific nodes (as indicated by the inhibition value) due to past failures. However, one may be more or less motivated to solve a task at hand. For example, one may be bored by the repetition of similar tasks, or anxious due to perceived task difficulty and knowledge insufficiencies. Such a lack of perceived fit among capabilities and task requirements may influence the initial levels of motivation and attention, which could be modelled by varying the degree of initial activation.

5.1.3 Team member agent's personality and cognitive ability

The reasoning processes and mechanisms described in previous subsections are universal among agents. However, in many applications, it may be convenient to introduce differences in the agent's knowledge, skills or personality. The expertise area, as discussed previously, can vary among agents, thus introducing the differences in their knowledge and permitting the studies of the effect of knowledge overlap on the team performance. In this subsection, another way of adding variety is discussed: through modelling of different personality traits and cognitive abilities.

In the developed model, cognitive ability and personality traits are modelled as parameters and remain constant (for an agent) during the simulation. As a default setting, each of the aspects is presumed to follow a normal distribution and is, thus, set by drawing a random number from 0 to 1, with a mean set to 0.5 and a standard deviation of 0.166. The 0.5 of each dimension marks the average/balanced ability/personality; and smaller amounts indicate individuals with less cognitive skills, and more introverted or disagreeable personality.

As emphasised previously, in this work, Cowan's [183] view of working memory is used. Thus, once the cognitive ability parameter (ranging from 0 to 1) is set, it dictates the value of focus capacity parameter as follows: if the cognitive ability parameter is within one standard deviation from the average value, the focus capacity is set to 4; otherwise the value is set to 3 or 5 depending on the cognitive ability value. Formally:

$$focusCapacity = \begin{cases} 3, & cognitiveAbility < \mu - \vartheta \\ 4, & cognitiveAbility \in [\mu - \vartheta, \mu + \vartheta] \\ 5, & cognitiveAbility > \mu + \vartheta \end{cases} \tag{5.12}$$

where $\mu = 0.5$, and $\vartheta = 0.166$.

As its name implies, the focus capacity parameter specifies the number of chunks that can be processed (i.e. focused) in a single simulation step. The cognitive ability parameter also guides learning. Namely, at each update of knowledge link weight, the increment (c parameter) is scaled with the cognitive ability parameter, thus capturing the notion that individuals with higher levels of cognitive skills acquire knowledge easier.

The modelled personality traits influence the agent's interactions and communication with others (as Section 5.2 further details). For an agent to decide that certain knowledge element (i.e. a knowledge link or a structure) is worth sharing with others, activation or a score of the knowledge element has to exceed a *sharing threshold*. The extraversion parameter guides the value of the sharing threshold, thus indicating the higher or lower levels of reservation towards others. Formally, the sharing threshold is calculated as:

$$sharingThreshold = activationThreshold + 2\,\delta\,(1 - extraversion), \tag{5.13}$$

where $\delta = analysisThreshold - activationThreshold$. Therefore, if the agent has average (i.e. 0.5) level of extraversion, it is comfortable with sharing the knowledge elements it has analysed. In the extremes, if the agent is overly extraverted (i.e. extraversion = 1), it will share anything active sufficiently to enter its 'consciousness'. On the other hand, as a default, an extremely introverted agent (i.e. extraversion = 0) shares only the knowledge elements active to their maximal level.

Agreeableness influences how agents process messages (i.e. knowledge elements) expressed by others. The agreeableness parameter dictates the level of initial trust the agent holds in others (see Section 5.1.4). Each time a team member communicates a knowledge link, the amount of activation sent to relevant knowledge nodes (in the listener agent's mental model) is proportional to the agent's trust in the speaker. The amount of activation governs whether the relevant link will be created in the listener's mental model. In other words, the more agreeable the agent is, the more likely it is to hold high trust in others, and thus to pay more attention to other's messages: the agreeable agent is more likely to learn from others. However, even if the agent loses trust in others, it may still feel the urge to comply with the group solution. Therefore, the level of agreeableness (irrespective of the trust level) influences the amount of activation the agent will send to the structures proposed by its team members. The level of structure's activation affects the thoroughness of its analysis. Thus, the more agreeable the agent is, the more likely it is to adopt team member's structures and build on them. Finally, the effect of perceived warmth/hostility is included in agent's evaluations: the team members will perceive agent's evaluation of the proposed structure as more positive or more negative than actual, depending on the agent's agreeableness level. In the extreme cases where agreeableness of the rater is 1 (or 0), the perceived score is 0.05 higher (or lower) than the actual score the agent assigned to the structure.

5.1.3.1 Overview and limitations

The previous subsection presented a simple model of the effects of cognitive ability, extraversion and agreeableness parameters on the agent's behaviour. In particular:

4. Cognitive ability parameter governs the learning speed and captures the individual differences in the working memory capacity by regulating the number of elements that can be processed in a single step.
5. Extraversion parameter sets the threshold the knowledge element's activation has to surpass in order to be eligible for communicating to others.
6. Agreeableness parameter guides the level of attention the agent will dedicate to other's ideas and can influence the team members' perception of the agent's remarks.

It should be noted, however, that more research is needed to determine if labels "agreeableness" and "extraversion" are ontologically suitable (or there are more specific terms) for the effects captured by the implemented mechanisms. For example, the implemented agreeableness influence has many similarities to what [106] denotes as *group conformity*.

The absence of influences of the "getting ahead" group of personality traits (neuroticism, conscientiousness and openness) is a clear limitation of the proposed model. The possible effect of anxiety (a facet of neuroticism) on the frustration modelling is discussed previously. Further, a neurotic person is often described as prone to guilt, emotional reactions and insecurity [238] and thus may react to the dismissal of their ideas more pronouncedly than their emotionally stable colleagues. For example, in response to criticism, their sharing threshold may change or their focus can be directed towards well-grounded knowledge.

Conscientiousness is a trait ascribed to cautious, careful and responsible people who like to follow well-established rules [231]. Thus, conscientious agents may, for example, be more motivated (i.e. have higher levels of activation present in their mental model) and have a higher threshold for solution acceptance (i.e. do not accept solutions which do not fit all of the requirements). Further, they may dismiss knowledge links which are not well-grounded, and be more reluctant to abandon solutions which were successful in the past (e.g. by modelling a slower increase in inhibition of past solutions).

Openness is associated with preference for new and unexplored, and it signals that a person is more apt to change [280]. Once the current model is extended to include the requirement of novelty (i.e. that a structure agents propose as solution necessarily has to be new in a context of a task or situation), agents may be modelled to place different emphasis on solution's utility (i.e. meeting the requirements) and solution's novelty. Then, a conscientious agent may be modelled to place a great emphasis on structure's utility [236], while agent high on openness would have a higher tolerance for novelty and place a greater emphasis on that component of the solution's score. These ideas build on the work of Norman [203] who suggested that an average person may be willing to accept a structure not meeting all of the requirements if it is viewed as fun and creative. To implement a desired "preference for novelty" aspect, the future model development will follow the work on curiosity in computational agents [281].

The developed model of cognitive ability's and personality factors' influence on reasoning does not come close to capturing the rich interplay among these and other factors (e.g. gender or age) observed in real-world settings (e.g. see [282]). For example, cognitive ability is related to social learning, memory duration, habituation, generalisation and many other aspects [283]. The model does not explicitly include the correlations among personality traits [284], nor relations between cognitive ability and personality [230]. Influence of personality on the motivational factors is also omitted [232]. Further, a large number of studies noted the effect personality traits have on the cognitive ability of older adults [285]. Aside from age, the current model neglects the gender [286] or cultural [239] differences of participants.

The current model views personality traits as stable (which is consistent with the literature [287]). However, researchers note that various situational aspects may dictate which aspects of personality will be manifested [288]. This stream of research relates to the necessity of a more detail modelling of emotions, moods and states. The current model, however, takes a straightforward approach to modelling the interplay of situation, affect and personality. For example, the implemented agents are more susceptible to frustration than to (dis)agreeableness, so the frustration impulse is equally distributed to both, structures which were proposed by others, and to those which were proposed by the agent. This may cause the agent to suggest structures which it rejected several steps prior. The future work directed towards mitigating these limitations may build on the ideas from computational models of emotions which utilise FFM (e.g. Mamid [289] and Alma [290]).

Personal differences can also be expressed in respect to cognitive style variations [111]. Another point of interest may be modelling differences in learning and forgetting. Namely, research suggests that forgetting rate (and perhaps the learning rate), although stable, varies across domains [291]. The many examples of possible sources of variety among people (and agents) are too broad to be captured by this model. However, the prioritisation of the future improvements will be led by the research on adaptation and creativity.

5.1.4 Team member agent's reputation, trust and perception of others

A final aspect of agent's architecture deals with the way the agent perceives - and is perceived by – other agents. Particularly, this subsection covers the elements related to agent's learning about and from its colleagues, a perquisite for the group of agents to simulate a team (rather than an isolated work of multiple agents).

The agent's (cognitive) trust in others is tied to their domain of expertise. Since the current implementation defines agent's domain of expertise as a set of behaviour nodes (see Section 5.1.1.4), the notion of transactive memory and trust are similarly associated only with the behaviour layer knowledge elements. For every behaviour node b (known to agent a_i) and a team member a_j, agent a_i determines the value $trust_i(b, a_j, n)$ representing the current (i.e. in the *n*-th task in a task sequence) perceived level of a_j's expertise in the subject b. Following the research presented in the Section 4.4.4, the initial values of *trust* are based on the agent a_i's agreeableness and the domain of expertise of the agent a_j. More precisely, for every behaviour node $b \in B_i$ known to the agent a_i (where B_i represents a set of behaviour nodes known to the agent a_i), and every agent $a_j \in A \setminus \{i\}$ (where A is the set of all agents), the initial value of $trust_i : B \times A \setminus \{i\} \times \mathbb{N}_0 \rightarrow \mathbb{R}$ function is set to a sum of agent a_i's agreeableness and a randomly generated number representing the perceived agent a_j's competence $tm_i(b, a_j, n)$:

$$\begin{aligned} trust_i(b, a_j, 0) &= \Delta Trust\left(agreeableness_i + tm_i(b, a_j, 0)\right) + minTrust, \\ \Delta Trust &= maxTrust - minTrust, \end{aligned} \tag{5.14}$$

$$tm_i(b, a_j, 0) = \begin{cases} 0.5 + random(0,0.5), & b \in expertise_j \\ 0.5 - random(0,0.5), & b \notin expertise_j \end{cases}. \tag{5.15}$$

This reveals several of the developed model's limitations: the current implementation neglects other aspects of trustworthiness (benevolence and integrity), presumes an equal influence of agreeableness and competence, and assumes that agent's domain of expertise is known to others even if they have no prior experience in working together.

The modelling decision to tie a specific trust value to each of the behaviour nodes enables straightforward definitions of *general trust* (held in a particular agent) and *situational trust*. If the agent a_i perceives the current situation as consisting of a set of behaviour nodes $B_i' \subseteq B_i$, the situational trust in agent a_j is formed as an average of trust values for the relevant nodes:

$$situationalTrust_i(a_j, n) = \frac{1}{|B_i'|} \sum_{b \in B_i'} trust_i(b, a_j, n)\,. \tag{5.16}$$

The perception of the set of required behaviour nodes may change during the task, as the agent can learn new behaviour nodes and realise they are required. Along with the changes in the perceived set of required behaviour nodes B_i', the situational trust in the agent a_j changes as well.

If the agent does not know any of the required behaviour nodes, its level of trust in each of its team members equals the *general trust* value. Formally, *general trust* is defined as:

$$generalTrust_i(a_j, n) = \frac{1}{|B_i|} \sum_{b \in B_i} trust_i(b, a_j, n). \tag{5.17}$$

Trust the agent holds in a trustee influences the agent's perception of the messages shared by the trustee. To simulate the amount of attention dedicated to the communicated knowledge link, the activation of relevant (i.e. communicated) nodes is increased by the value of (situational) trust held in a speaker. If the activation surpasses the node activation threshold, the listener agent will learn, or further ground, the communicated link. Additionally, the activation received as a result of listening can cause relevant nodes to enter the focus of attention.

Trust also serves to determine one's influence on the acceptance of the final solution. A value named *reputation* (or a *team trust*) denotes the average level of trust that team members hold in the agent a_i and it guides the level of importance the team assigns to the agent's solution evaluation. Thus:

$$reputation_i(n) = \frac{1}{|A| - 1} \sum_{\substack{j=1 \\ j \neq i}}^{|A|} situationalTrust_j\,(a_i, n). \tag{5.18}$$

When determining if the team will accept a structure as a solution (see Section 5.2. for details), the agent's reputation defines its 'share' in the structure's score formation. Thus, if every agent in the team is equally (dis)respected, their assessments of the proposed structure are averaged. Otherwise, the more respected members get more influence over the structure's acceptance or dismissal.

Trust among agents changes as a result of team successes and failures. At the end of each task, the team receives an evaluation of their solution, which states which requirements were or were not satisfied. Then, for each of the relevant behaviour nodes, agent a_i updates values of trust in others. As before, let $requiredMeasure^i$ denote a particular property (i.e. network measure) upon which the requirements are posed, and let *n* denote the ordinal number of the task in a task sequence. Further, let $requiredSet^i$ denote a set of allowed behaviour nodes (i.e. intervals allowed for the particular property):

$$requiredSet^i = \{rB \in B \mid measure(rB) = requiredMeasure^i\}. \tag{5.19}$$

The structure meets the requirement posed upon the $requiredMeasure^i$ property, if it is connected to one of the behaviour nodes in $requiredSet^i$ set. Then, the agent increases the trust in its team members (only) for the particular behaviour node obtained. In other words, upon completion of the *n*-th task during which a behaviour node $rB \in requiredSet^i$ is obtained, the trust in an agent a_j (to be used in the next task) is updated as:

$$\begin{aligned} trust_i(rB, a_j, n+1) \\ &= trust_i(rB, a_j, n) + trustChangeRate \\ &* \left(maxTrust - trust_i(rB, a_j, n)\right). \end{aligned} \tag{5.20}$$

If, on the other hand, for a certain $requiredMeasure^k$ property, the obtained behaviour node does not correspond to any of the acceptable intervals, the trust value is decreased for each of the $rB \in requiredSet^k$ nodes. The trust is lowered for every agent a_j:

$$\begin{aligned} trust_i(rB, a_j, n+1) \\ &= trust_i(rB, a_j, n) + trustChangeRate \\ &* \left(minTrust - trust_i(rB, a_j, n)\right). \end{aligned} \tag{5.21}$$

5.1.4.1 Overview and limitations

The included model of transactive memory, trust and reputation is very simple and models only several aspects of the agent's perception of its team members. In particular:

1. The agent forms perception of every of its team members based on their (perceived) capability to resolve particular requirements (i.e. capability to produce structures with particular behaviours).
2. Trust and reputation are each approximated by a single value calculated based on the formed perceptions.

3. Perceptions the agents have about each other are dynamic and influence communication and learning within the team.

As described, trust value guides the level of attention the agent pays to messages (i.e. knowledge links) proposed by a trustee. This may seem contrary to the proposition of Salas et al. [32], who noted that a lack of trust causes people to spend more time checking the other's input. However, in the context of the simulated task, the "input" is strictly in the form of messages transmitting knowledge. As modelled, the trust level signals the perceived amount of other's expertise, and a lack of trust may cause agents to dismiss others' messages due to perceived incompetence. In line with the implementation are findings of [292], which argued that trust influences communication and a lack of trust might result in aggressive communication and distortion or dismissal of facts. As currently implemented, the lack of trust undermines the effects of communication, thus inhibiting the dissemination of knowledge. The lack of trust, therefore, may – in line with Salas et al. [32]'s findings - result in the prolonged time needed to complete the task. The experiments reported in the next chapter test whether the current implementation indeed has such an effect.

The initial trust formation neglects the influence of a trustee's characteristics: one's likability, enthusiasm and benevolence are not taken into account. Further, the developed model assumes that each team member can recognise which requirements are falling within other's expertise area. This may sometimes be the case, but future model versions will include the ability of simulating teams without any presumptions of others' knowledge. Additional shortcomings are related to the lack of affective trust [249] and trust-generating structures' [293] modelling. The agents are currently not affected by homophily [241], and there is no transitivity or reciprocity of trust. As the most negative influence of (the lack of) trust, the current model includes the dismissal of messages. There are no other negative influences: e.g. distrust [294], workspace gossip [295] or conflicts [296], and, thus, their influences on reputation are not modelled.

Another limitation of the current implementation is the lack of more thorough reasoning on team performance. For example, agents faced with failure will reduce trust in all of the team members, irrespective of the task dynamics: agent a_i's trust in agent a_j will decrease even if agent a_j had good ideas (which were dismissed by the team).

The current model assumes agent-to-team communication, and thus no learning form observation is included. Further, all of the communication is regarded as face-to-face (i.e. no influence of communication technology is modelled). Future model versions will likely need to allow additional modes of communication, learning from observation and their impact on trust formation. For example, research [252] demonstrates the differences in trust formation and influence in teams operating in virtual settings.

Finally, the agent's perception of others can be detailed to enable inferences about other's knowledge, preferences or motivation. For example, an agent can have a perception of others' priorities. Similar to the agent's profile definition (see Section 5.1.1.4), the agent's model can be extended to enable a_i to calculate other agent's (a_j) profile and use it to determine which requirements (i.e. properties) the agent a_j prioritises. The profile perception can influence communication among agents: agent a_i may give more thought to those message aspects which are perceived as relevant to a_j, thus increasing the likelihood of learning the respective behaviour-structure links.

5.2 Communication and interactions among team member agents

Section 5.1 introduced much of the details regarding the communication among agents. This section serves to clarify the interaction rules further and to provide a comprehensive overview of the communication protocols.

In order to enable communication among agents, the agent's cognitive behaviour diagram (Figure 5.5 and Figure 5.6) had to be extended. The agent working in a team setting has to be able to influence and be influenced by its team members. A *sharing threshold* is introduced (see Section 5.1.3) to serve as guidance on whether the agent has a potential solution to propose to the team. If the structure's score is perceived as above sharing threshold, the agent will try to propose it to the team. The structure's score depends on the weight of the relevant behaviour-structure links and – if enabled – on the agent's profile (i.e. priorities). More precisely, for every required property ($requiredMeasure^i$), the agent assesses whether the structure's network fits within a desired range (i.e. obtained behaviour node rB is within $requiredSet^i$). If the structure meets the requirement, the weight of the link from rB to the structure, scaled by the perceived priority of the requirement (denoted $profile^i$) is added to the structure's score. Thus, if the structure is denoted with $s \in S$, one can write:

$$score(s) = \sum_{\substack{i=1, \\ rB \in requiredSet^i}}^{m} w_{rB,s} * profile^i, \tag{5.22}$$

where *m* denotes the number of properties the requirements are posed upon. Note, if the profile is not calculated, each agent is modelled to regard requirements as equally important (i.e. then $profile^i = \frac{1}{m}$ for every required property *i*).

If the agent has no structures to propose, it checks whether any of the links is sufficiently active to be communicated. The "link activation" depends on the activation of its nodes, and the weight of the link. Each of the three elements (two nodes' activation and link weight) has to surpass the sharing threshold to be eligible for communicating. If more than one link meets this requirement, the most active link is shared.

The agents that 'wish' to share a knowledge element with others apply for communication by sending a message to a *control agent*. Control agent is not meant to represent any real-world entity. Instead, it serves as simulation support by directing the communication among agents, taking care of the task sequencing, and collecting the statistics on the performed simulations.

At each simulation step, the control agent receives all of the team member agents' applications and randomly selects a *speaker*. Speaker's message is then transmitted to every team member agent. If the communicated message contains a structure proposed as a solution, the agents are prompted to evaluate it against their mental models and to send back (to the control agent) their evaluations. Control agent receives the evaluations, weights each with the respect to the reputation of the sender, and presents the overall score (i.e. *team score*) to the team member agents. If the team score exceeds a predefined *acceptance threshold*, the structure is deemed as accepted by the team and the task is finished. Note, however, if the acceptance threshold is set to be below 1.0 value (i.e. maximal score), the team may accept an unsuitable structure as a solution. Such situation, for example, occurs if every agent in a team assigns a very low priority to a certain requirement.

If the communicated message contains a knowledge link, or if the acceptance threshold is not met, the agents continue their search for the solution. Communication affects agents' mental models. If the link is communicated, listener agents will activate the respective nodes and possibly (i.e. if the nodes become sufficiently active) learn or further ground the communicated link. As discussed previously, the amount of activation depends on the trust the listener holds in the speaker. Similarly, the explication of the scoring team assigned to the structure has an impact on the activation of the structure in the listener's mental model. The listener agent compares its score with the team score. If a discrepancy is detected, the listener agent adjusts the level of inhibition of the structure. The change in inhibition depends on the scores' difference (*team score – agent's score*), and the agreeableness of the listener. The team member agent's cognitive behaviour on a single task performed in a team setting is presented in Figure 5.7.

Figure 5.7 Agent's cognitive behaviour in a task performed in a team setting

The presented communication model has many limitations. A modelling decision that only knowledge links present in agent's mental model can be communicated, results in only five message types which team member agent can transmit:

1. *Requirement node – Function node* link,
2. *Function node – Behaviour node* link,

3. *Behaviour node –Structure node* link,
4. *Solution proposal*,
5. *Proposed solution's evaluation.*

Studies of designers demonstrated that all of the FBS processes occur - and are verbalised – during teamwork [201]. Additionally, designers are shown to externalise thoughts implying the occurrence of System 1 thinking, i.e. expressing *Function node – Structure node* links [267]. Although the developed agents are capable of performing almost every of the FBS processes internally, they have no means of explicating all of them. Thus, the current model has deficiencies in its capability to replicate the real-world design communication. Consequently, the communication patterns extracted from simulation results cannot (yet, i.e. from the current implementation) be directly compared to the real-world data. Similarly, the developed model neglects important aspects of team communication such as asking for clarification. The simulated communication follows the so-called "Process 2" of the processes of thinking in design teams identified in [202], but many details remain to be added to increase the realism of agent communication.

A related issue is that of *noise* modelling. Currently, agents may dismiss others' messages due to trust deficiencies. Further, even when trust is sufficient, the agent may 'choose' to focus on unrelated knowledge elements. However, there are no errors such as mistakenly activating nodes that resemble (based on some criteria) communicated ones. Similarly, proposed structures are always correctly encoded in agent's mental model: the structure's network, no matter how complex, is fully understood by the listeners – and the listeners make no mistakes in analysing its properties. The newly learnt structures are likely to receive low scores at first evaluations, as the behaviour-structure links take time forming. Thus, several simulation steps are needed for agents to get a realistic view of the structure, which may account for some of the time needed to 'present and explain' the structure (to others) in the real world. Nevertheless, future implementations should include faulty reasoning, information loss and a more realistic model of structure explication.

The detailing of an information loss and noise model will enable improvements regarding several communication aspects. For example, each time the agent is interested in the message but does not manage to capture it in its entirety or is not familiar with the content, it should be able to send a request for clarification. The agents could answer the requests for clarification by explicating links whose processing led to the communicated message. Thus, noise modelling can be coupled with enabling "Process 1" thinking in design teams [202].

Another limitation of the current implementation is a predefined order of the agent's actions. For example, agents always give priority to solution proposal: if the agent has found a structure whose score is perceived to be sufficiently high, it will propose the structure, irrespective of the activity level of processed links. It is plausible that one would like first to introduce relevant links, and only then propose a structure as a solution. Similarly, if the solution is proposed, agents (as currently modelled) are required to evaluate it immediately. Instead, agents should be given an option to choose the desired action (e.g. evaluate, ignore the proposal, request clarification, offer a solution modification) by utilising, for example, Luce's choice axiom [297] where trust in speaker and activation of the relevant nodes may be used to determine the choice weights. Note, if one wishes to simulate brainstorming (where no solution evaluations should be shared [298]), then a step consisting of evaluation sharing should be completely omitted.

Finally, as already emphasised, all agent communication is restricted to messages that can be formed using FBS ontology. Thus, agents currently cannot coordinate task assignments, discuss managerial issues, joke, or engage in personal and other non-design communication. Jiang [201] notes that such communication is related to extra-design activities or is not directly relevant to the task. However, such communication may be important for studies of social ties' impact on the team, or conflict modelling. Thus, future efforts may be directed toward capturing these aspects of communication.

5.3 Task representation

A task is defined as a set of requirements. Currently, requirements are posed only on a structure network's properties and are thus tied to behaviour nodes. The first step in defining a task is the selection of one or more network properties. Let $G = (V, E)$ denote a graph G defined by a set of vertices V and a set of edges E. The properties implemented to date include:

- **Diameter** [299]: defined as a maximum - over all node pairs – of the shortest path lengths. To set an upper boundary of possible values, the obtained diameter value is further divided by the number of nodes in the network (i.e. scaled to fit [0, 1] interval). Formally, let $D(i, j)$ denote a shortest path between nodes i and j, if such path exist (otherwise $D(i, j) = 0$). Then diameter is defined as:

$$diameter(\mathrm{G}) = \max_{i,j \in V}(D(i, j)) \tag{5.23}$$

  If the graph is disconnected then the value is calculated on the set of all pairs of connected vertices [300]. Then, the final diameter property value (i.e. used in simulations) is obtained as: $diameter(\mathrm{G})/v$ where v denotes the number of nodes in a graph $v = |V|$.

- **Network degree centralisation** [301], [302] defined as follows: let $degree_{max} = \max_{i \in V}(degree(i))$ where $degree(i)$ denotes the out-degree of node *i*. Then, network degree centralisation is calculated as:

$$degreeCentralisation(\mathrm{G}) = \frac{\sum_{i \in V}(degree_{max} - degree(i))}{(v-1)(v-2)}. \tag{5.24}$$

- **Network closeness centralisation** [301], [302]: Let $closeness_{max} = \max_{i \in G}\big(closeness(i)\big)$ where $closeness(i)$ is the closeness centrality of node *i* calculated using [303]. Then, network closeness centralisation is calculated as

$$closenessCentralisation(\mathrm{G}) = \frac{\sum_{i \in V}(closeness_{max} - closeness(i))}{(v-1)(v-2)\big/_{2v-3}}, \tag{5.25}$$

if the graph is connected. Otherwise, the value is calculated on the network's largest connected component [304].

- **Network betweenness centralisation** [301], [302]: Let $betweeneess(i)$ denote betweenness centrality of the node $i \in V$ calculated using [305]. Then, network betweennes centralisation value is calculated as:

$$betweennessCentralisation(\mathrm{G}) = \frac{\sum_{i \in V}(betweenness_{max} - betweenness(i))}{(v-2)(v-1)^2}, \tag{5.26}$$

where $betweenness_{max} = \max_{i \in V}(betweenness(i))$.

- **Average clustering coefficient** [306]: Clustering coefficient for node *i* is often calculated as a fraction of node *i*'s neighbours that are neighbours themselves. Thus, the average clustering coefficient is:

$$averageClusteringCoefficient(\mathrm{G}) = \frac{1}{v}\sum_{i \in V} clusteringCoefficient(i). \tag{5.27}$$

- **Hierarchy measure** [307]: Let $D(i, j)$ again denote the shortest path from node *i* to node *j* [308]. The hierarchy value of the network is calculated as:

$$hierarchy(\mathrm{G}) = \frac{1}{v-1}\sum_{i \in V}(h_{max} - h(i)), \tag{5.28}$$

where $h(i) = \frac{1}{v-1} \sum_{j \in G} \frac{1}{d(i,j)}$ and $h_{max} = \max_{i \in V}(h(i))$.

- **Density** [299], [302] is defined as a portion of the network links which are realised. Formally:

$$density(\mathrm{G}) = \frac{2 * |E|}{v * (v-1)}. \tag{5.29}$$

Once a set of properties is selected, for each property, an interval of permitted values (i.e. an interval within which structure's properties will have to fall to satisfy task requirements) is chosen. The behaviour nodes tied to the permitted property values are, thus, regarded as required. In the studies presented herein, every task was designed to have a feasible solution: i.e. it was reassured that a network satisfying all of the requirements can be constructed.

When creating a space of all possible behaviour nodes, for each of the properties, a random number from 3 to 10 is selected, indicating the number of subintervals the property's range is divided into. Each of the created subintervals is assigned to one behaviour node. Thus, the number of behaviour nodes in a simulation ranges from 21 to 70.

Note, each of the properties currently modelled has a codomain (i.e. a range of plausible values) of [0, 1]. When dividing this range into subintervals, the subintervals' boundaries are not chosen as equidistant. There are two reasons for such modelling decision. First is the fact that irrespective of the property taken, not all values within the property's [0, 1] range are equally probable. Secondly, the mechanisms used for structure creation (i.e. union and contraction mechanisms) shape the solution space in such a way that certain values become much easier to achieve. For example, employing union creates larger and larger networks for which low betweenness centrality values are more probable than high betweenness centrality values. Thus, if the subintervals were chosen to be of the same length, the (randomly) created tasks would significantly differ in their difficulty – even if they posed requirements on the same properties, and the same number of intervals were considered admissible. With the aim of enabling the (partial) control over task difficulty, a set of 100,000 simulations of default (i.e. 1,000 steps) length were performed, and the properties of every created structure's network were collected. Then, an analysis of each of the properties was conducted and the intervals of approximately equal probability (of a structure's property values falling into it) were determined. It should be noted, however, that the conducted analysis did not study correlations among properties, thus preventing the user from gaining complete control over the task difficulty. The required analysis of network properties will be performed in the future.

A structure is considered feasible (i.e. a task solution) if each of its network's properties fits within the property's allowed range. Thus, a structure either is, or is not, a solution for a given task. Such binary nature of structure's viability is overly simplistic. A more realistic approach would, for example, include a 'desired value' and an 'acceptable range' (which contains the desired value) of a property, so that several structures meeting the requirements could be ranked based on how close they are to the optimum (i.e. desired solution). But real-world situation is much more complex: conflicting, soft or ill-defined requirements are common in design [202]. Often the exact requirements are unknown at a task start [57], new requirements are introduced during task execution [309], or the existing requirements are changed or dismissed [310]. The future efforts will necessarily be directed towards enabling the simulation of such cases.

Several other limitations of the current task implementation have already been discussed. For example, the requirements are currently posed only on the behaviour nodes and the prerequisite of solution's novelty is not by default included in the task requirements. Additional refinements of the current model may be aimed at tailoring the simulation to fit different task types (e.g. brainstorming), model various task environments (e.g. virtual teams) or permit usage of external resources such as databases or documentation of past projects during the task performance. Finally, future studies may take a closer look into the nature of the simulated network properties, and possibly match them to real-world behaviours. Additional network properties, as well as new agent's structure-creating mechanisms, should also be modelled.

5.4 Simulation outline and scenarios

Figure 5.8 shows the agent's behaviour over a sequence of tasks, i.e. over one simulation run. At the simulation start, agent's mental model (known function, behaviour and structure nodes and the links among them) and expertise, personality, cognitive ability and initial trust in others are set. After each task, the agent's knowledge is updated (as described in Section 5.1.1.3): weights of the links used in the previous task are increased, while links not used are decreased in weight. Further, the agent's trust in others is modified based on the success of the completed task. At the start of the next task, (if relevant modules are enabled), the agent uses past experiences to assign base-level activation and initial inhibition values to knowledge nodes. Further, its perception of the priority of the next task's requirements is formed. Throughout the simulation, the agent learns and consequently changes its behaviour.

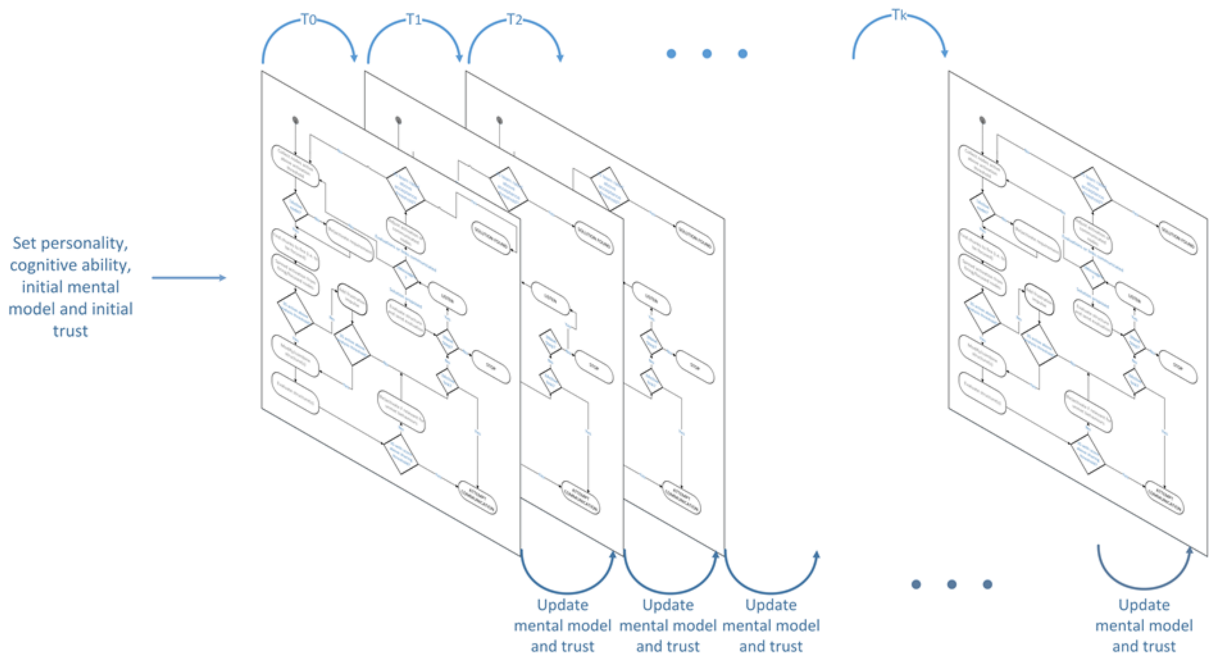


Figure 5.8 Agent's behaviour over a sequence of tasks

The implemented model enables various simulation configurations. To define one task, the model's user selects:

- Number of agents involved
- Number of steps allocated (i.e. simulation length)
- Whether to include optional aspects of agent's architecture (base level activation and initial inhibition computation on the past tasks' basis, profile calculation, personality and cognitive ability, affect, trust)

Additionally, one can specify agents' personalities and cognitive ability, choose the level of overlap among agents' expertise area, or modify agent's communication to allow only solution-proposing message exchange (and no knowledge link sharing). Naturally, if needed, one can modify any of the parameters listed in Appendix C.

To specify a task sequence, the user can select:

- The number of tasks (and for which the statistics is collected),
- The similarities among tasks: do the tasks share some of the requirements or are they unrelated (or if the level of overlap is irrelevant),
- The pattern among tasks: do some tasks repeat,
- Whether the tasks need to be progressively more difficult (pose requirements on larger number of properties and specify a stricter range of acceptable values),

- If it is necessary to check that agents do (not) know the solution in advance,
- After which tasks the agents' mental models need to be restarted or a property changed.

The listed options enable great flexibility in experiments design. One should, however, keep in mind the limitations of the task generating mechanisms (as described in Section 5.4). For example, the only manner in which task difficulty is controlled is by extending or reducing the intervals of permitted property values. Due to the deficiency in modelling and control of correlations among properties, it may happen that, for example, extending the permitted range for one property does not reduce the overall task difficulty (if satisfying other requirements necessarily implies the values for the property to be outside of the 'added' subinterval).

6 VERIFICATION AND VALIDATION OF THE DEVELOPED MODEL

This chapter presents a series of tests conducted to study the behaviour of the developed model. In particular, the tests are aimed to ensure that implemented mechanisms are bug and error-free (i.e. verification) [311] and to assess the level of alignment between simulated behaviour and prominent theories, existing models or empirical findings of real-world studies (i.e. validation) [312]. Model's validity is its most important property [313], but the validation of agent-based models is often a cumbersome and lengthy process [40], [314]. In fact, in his book on modelling, Sterman [315] wrote:

"Instead of seeking a single test of validity models either pass or fail, good modellers seek multiple points of contact between the model and reality by drawing on many sources of data and a wide range of tests. Instead of viewing validation as a testing step after a model is completed, they recognise that theory building and theory testing are intimately intertwined in an iterative loop. Instead of presenting evidence that the model is valid, good modellers focus the client on the limitations of the model so it can be improved and so clients will not misuse it." (p.850).

Following the cited thought, multiple aspects of the model and its implementation were tested, and the obtained results are reported in this chapter. Through the presentation of model behaviour in various simulation settings, this chapter aims to demonstrate the model's capabilities and point out the model's limitations. The previous chapter discussed several limitations of the developed system. The tests reported herein serve to further the studies of the impact of model assumptions. Obtained results will guide the future model refinements, but could also provide valuable insights into the effects of the interplay of the implemented theories – inspiring, in turn, new empirical studies. Thus, as noted by [315], the developed model should not be regarded as 'completed'. Instead, the model, as well as theories it is based upon, will be further developed, increasing the level of validity of simulation output.

The validity of a model cannot be separated from the model's purpose [316], as a model is always validated in a certain context. As described in Chapter 1, the developed model is envisioned as a computational laboratory that will provide a means of conducting 'what-if' analysis and serve as a theory-testing tool. As a result, the model should increase the understanding of teams: the interplays of team processes, as well as the emergence of team properties and behaviours. Klügl [313] notes that such objectives do not require thorough statistical validation (in terms of calibration of the model to real-world data). In this chapter, the model's internal working and outputs are compared to the patterns observed in the real world. Nevertheless, future work will necessarily be aimed at achieving higher validity levels [314].

To conduct systematic testing of the implemented model, proposals reported in [259] and [316] are followed. The hierarchical testing was conducted: first by studying each of the agent's internal mechanisms in isolation, then by integrating them and simulating the behaviour of a single agent on one task, and afterwards by studying the agent's behaviour over multiple tasks. Once testing of the agent's work in isolation was completed, a team of agents was simulated in various scenarios to study the impact of different aspects (e.g. personality) on the team behaviour. To detect bugs, techniques such as animation, traces, code walkthroughs and internal consistency checks were used in each testing stage [317]. Further, model parameter values were varied to study their impact on the simulation outcomes. This chapter presents some of the most important results of such analysis.

6.1 Model testing in individual-agent setting

To test the individual agent's performance, a 'team' of one agent was created, and its performance on sequences of tasks was observed. Each of the conducted tests was aimed at studying a specific aspect of the agent's architecture that has an impact on the individual work. In particular, this subsection presents the impact of learning, inhibition, base-level activation, cognitive ability, affect and expertise area on agent's behaviour. The remaining aspects introduced in the previous chapters (e.g. personality) do not influence agents working in isolation – and were, thus, tested only in the team setting - the results of which are presented in Section 6.2. In each of the experiments comprising a single agent, the settings regarding communication were similar to the case where multiple agents are working collaboratively on a task. In other words, although it is performing alone, the agent simulates 'think aloud' studies by explicating structures and knowledge links that meet the sharing threshold (as described in the previous chapter). Such modelling decision was made to enable comparisons of the communicated content in individual and team settings.

6.1.1 Learning, forgetting and mental model update

The first step in testing the agent's learning (and forgetting) performance consisted of checking whether the agent's knowledge gets updated as expected during and in between tasks. Aside from code walkthrough and knowledge link weights' tracking, two experiments were conducted: one aimed to inspect if the agent's behaviour (in terms of differences in solution space exploration and explicated content) changes when it is repeatedly exposed to a task, and second aimed at studying agent's performance over a sequence of multiple (different) tasks.

6.1.1.1 Learning over a repeated task

The first experiment consisted of exposing the agent to a single task 10 times. In other words, a task sequence corresponding to one simulation run was of the form $T_1T_1T_1T_1T_1T_1T_1T_1T_1T_1$ where T_1, is a randomly generated task. The agent's performance on each task instance was tracked. To enable detail analysis, a distinction was made based on the agent's success on the performed task sequence. A task sequence (i.e. one simulation run) was deemed as successful if the agent managed to find - and settle for - a structure satisfying the requirements in at least one of the task performances. First, learning from successful task performances was studied. Then a separate analysis was conducted on a set of tasks for which the agent did not manage to find a solution within a specified number of simulation steps. The data were collected from 600 simulation runs performed (300 successful and 300 unsuccessful task sequences). Similar experiments of agent's learning over a sequence of identical tasks were conducted in, for example, for example, [268] and [177].

The average number of steps (and standard deviations) for each task in a 'successful' task sequence is presented in Figure 6.1. Further details on the number of steps needed to finish the task are shown in Table 6.1.

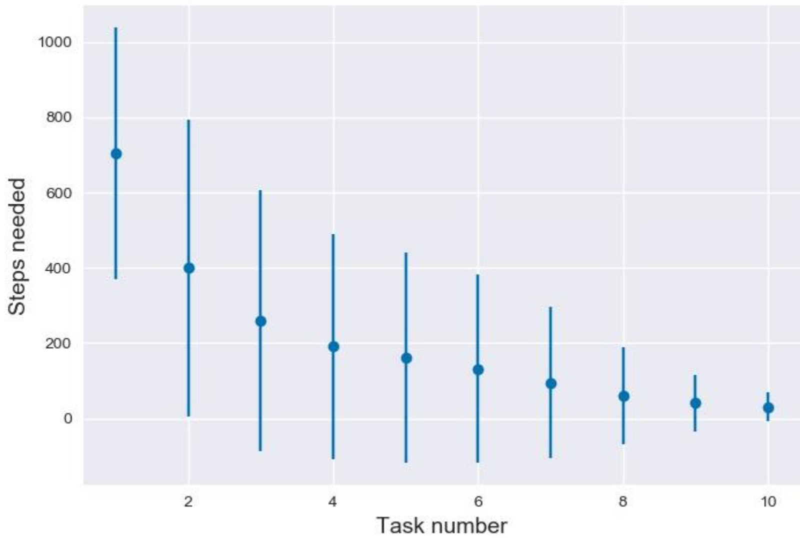


Figure 6.1 Average number (and standard deviation) of steps needed to perform a task over multiple repetitions

As evident from Figure 6.1 and Table 6.1, when repeatedly faced with a task it is capable of solving, the agent improves its performance in terms of the number of steps needed to find a solution. Even if the first several trials were unsuccessful (i.e. agent did not manage to find a solution within a default timeframe of 1,000 steps), once the agent manages to find a solution, it succeeds in every subsequent attempt. Statistical significance of the improvement in between tasks was tested using pairwise Wilcoxon tests, which confirmed the differences among every pair of task runs ($p < 0.01$).

Table 6.1 The statistics on the number of steps needed to perform a task over multiple repetitions (in successful runs)

Task	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Min	99	63	51	43	38	32	27	20	16	13
1st Q	335.5	96.25	72	60	51	42	35	28	23	18
Median	806.5	138	90.5	70	58.5	50.5	41	34	28	22
Mean	704.3	399.7	258.9	190.6	160.8	131.9	95.1	59.3	41.2	30.4
3rd Q	1000	957.75	129.8	91	74.75	60.75	51.75	42	34	28
Max	1000	1000	1000	1000	1000	1000	1000	1000	902	461

To gain deeper insights into the changes in the agent's performance in successful simulation runs, the number of new links learnt, the number of new structures created, and the number of explicated knowledge elements (i.e. links and structures) were counted. The boxplots for these values are presented in Figure 6.2. Further, as presented in Figure 6.3, the number of focused links and nodes was collected.

The presented figures (Figure 6.2 and Figure 6.3) demonstrate how the agent is adapting its search due to the experience gained in the previous task performances. The number of structures explicated decreases with each repetition, reaching the median of two structure considered (one of which was accepted) in the tenth task. Similarly, as the agent learns, it reduces the number of links explicated, thus indicating that it narrowed its search by grounding several links required to reach the solution. Similar findings can be obtained by observing the number of links and nodes focused on each task (Figure 6.2): with each task repetition, the number of knowledge elements focused decreases, and converges indicating that agent has found and grounded a set of knowledge elements tightly tied to the current task which it uses to reach the solution. The task repetition also has an impact on agent's learning of new links and structures: once the task is well understood, and the solution is found, the agent stops exploring the knowledge space and quickly converges to the solution. Therefore, as the agent performs the same task repeatedly, it stops learning new knowledge elements: by the 8th task, the median of new links learnt and new structures created is zero.

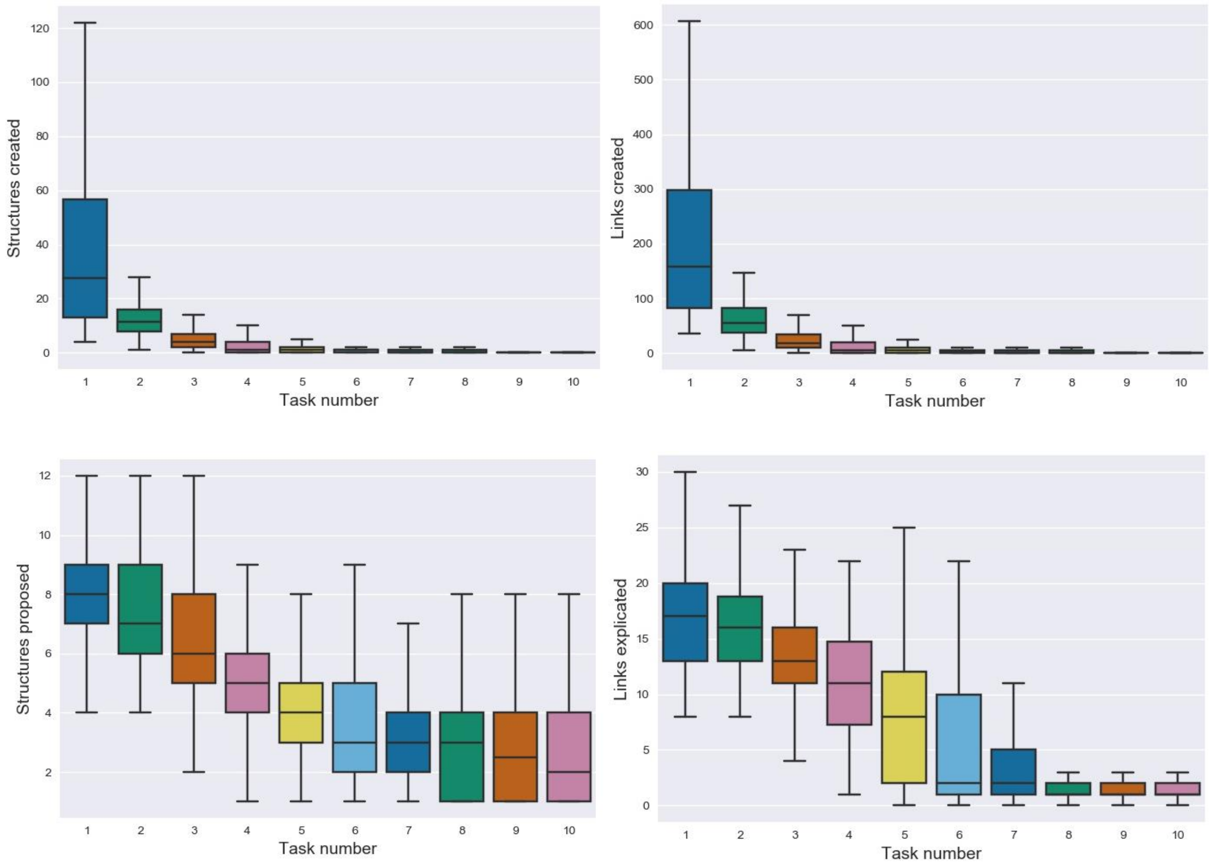


Figure 6.2 The number of new knowledge elements learnt and the number of knowledge elements explicated during each of the tasks (in a successful sequence)

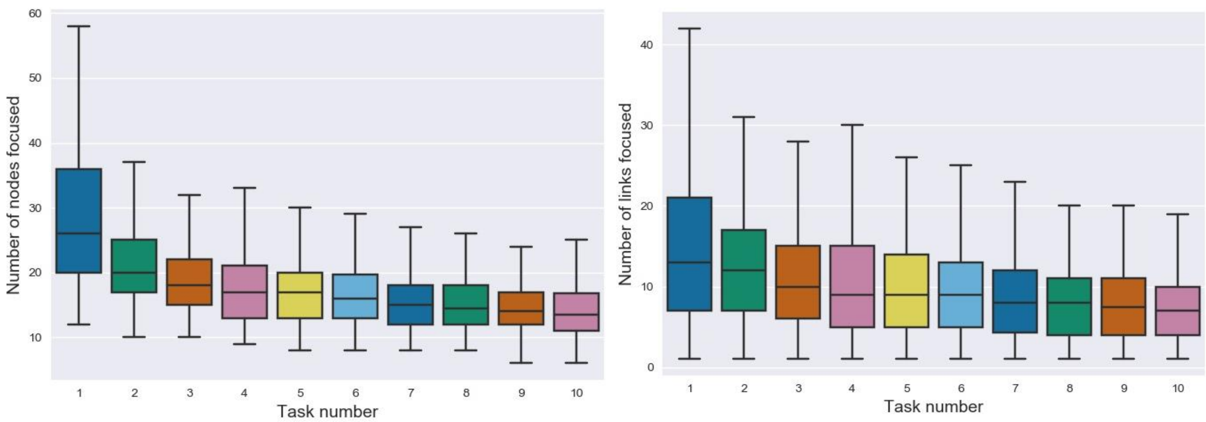


Figure 6.3 The number of focused nodes and links over the course of the task (in a successful sequence)

The similar trends in communicating the knowledge elements cannot be observed in case of tasks where the agent is unable to find a solution. Namely, although the number of new links and structures does decrease as the task gets repeated, the number of communicated links does not change significantly (as tested by pairwise Wilcoxon tests). The number of proposed structures stabilises after several task repetitions. The data shown in Figure 6.4 indicates that after several task repetitions the agent has explored the knowledge space but - as it is unable to find a solution - it is 'stuck' creating a small number of new structures and learning a few new links in each new attempt to solve the task.

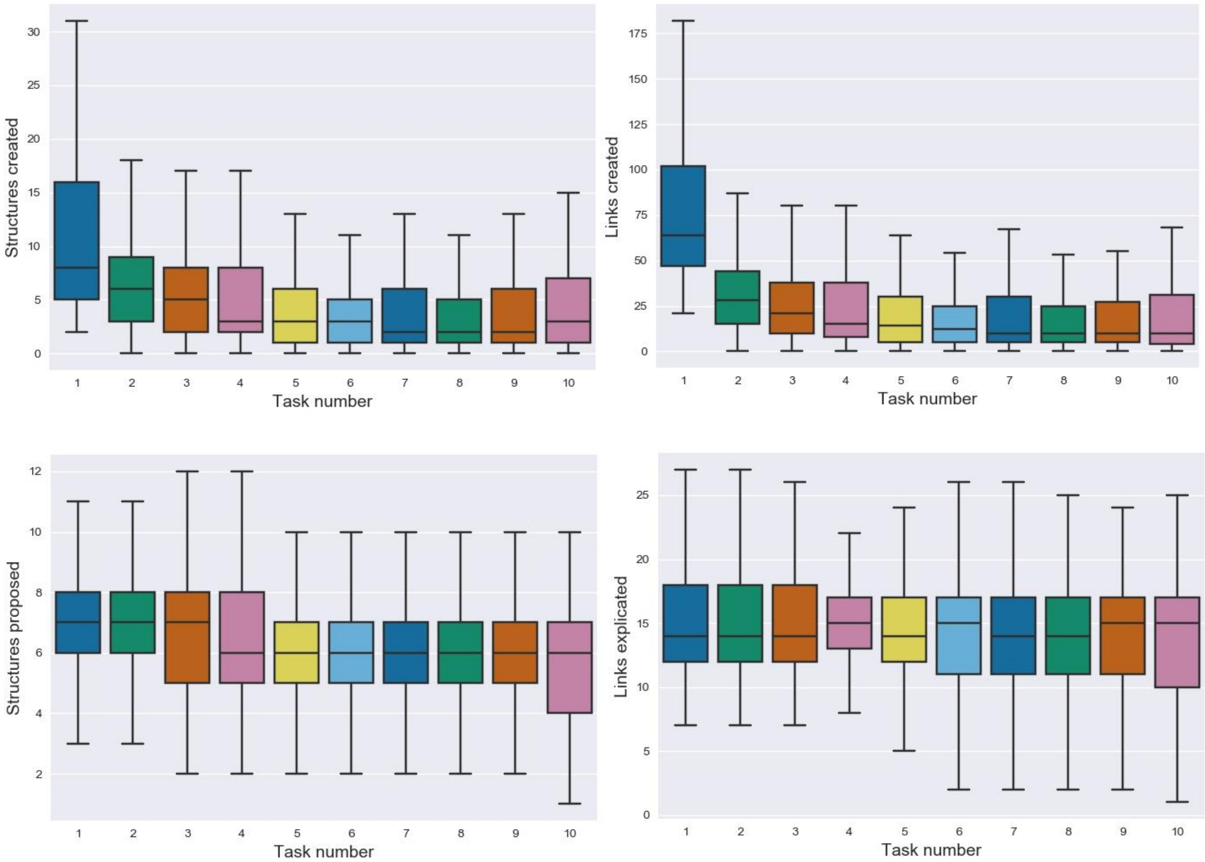


Figure 6.4 The number of knowledge elements learnt and explicated in a sequence of unsuccessful tasks

The differences between unsuccessful and successful tasks can also be observed when comparing the number of focused knowledge elements (i.e. the number of distinct knowledge elements which were focused during the task execution). Namely, as the agent is unable to find a solution, the links related to the task are repeatedly used, which results in further link grounding. Each time a task is reinitiated, the agent re-examines the relevant nodes and links and creates and grounds several more. Thus, as seen in Figure 6.5, the number of focused elements increases with each repetition of an unsuccessful task.

Finally, one can observe how the content of messages communicated by the agent changes throughout a task sequence performance. Figure 6.6 shows the average amount of each of the possible message types (requirement-function, function-behaviour and behaviour-structure links, solution proposal message and solution evaluation message) for each of the tasks in a sequence. In Figure 6.6, one can observe how – by gaining experience in performing the task successfully – agent’s messages shift toward solution-related issues (particularly, solution proposal and evaluation). This indicates that the agent grounded relevant requirement-function, function-behaviour and behaviour-structure links and is capable of quickly traversing them. In other words, as the agent repeatedly performs a task for which it has found a solution, it starts proposing the solution without reasoning on the problem-related issues. On the other hand, during the unsuccessful tasks, the agent rarely creates a structure deemed suitable for proposing as a solution. Rather, the agent is forced to re-examine the links related to the unmet requirements (due to the implementation of Reinforcement III process) and, thus, the proportion of the links related to requirements and functions does not decrease with such a large slope as seen in the case of successful tasks.

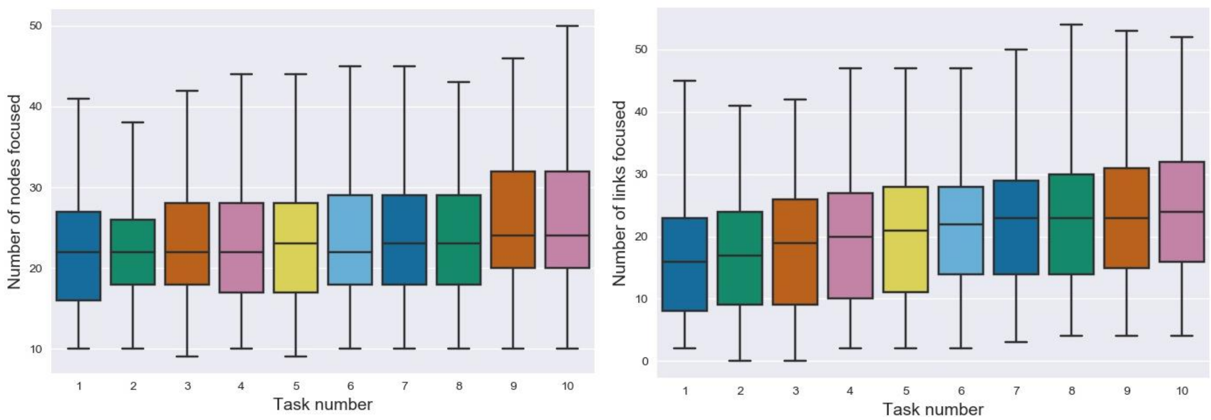


Figure 6.5 The number of focused nodes and links over the course of a sequence of unsuccessful tasks

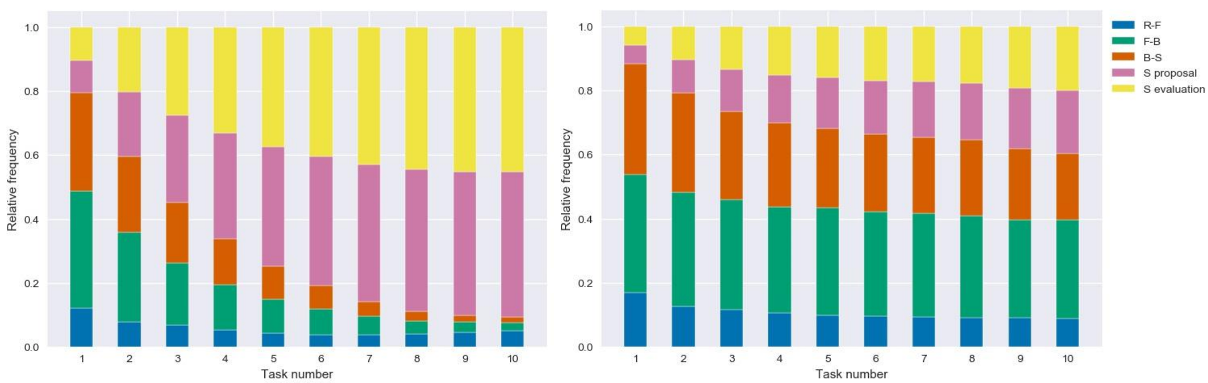


Figure 6.6 The distribution of agent’s messages for a) successful and b) unsuccessful tasks

The findings of the conducted experiments align well with expectations drawn from the presented implementation. The collected data provided evidence of the agent’s learning and demonstrated its impact on the agent's performance. Thus, the obtained results - coupled with the structured programming, code walkthroughs, syntax check and use of traces to determine the adequacy of link weight update - increase the confidence in the correctness of the implemented learning mechanism.

6.1.1.2 Learning over a sequence of different tasks

To study whether the agent's performance improves over a sequence of different tasks, the conducted experiment consisted of 1,000 simulation runs, where one run simulated a sequence of ten different tasks. Agent's performance on each task was compared to the performance the agent would display if no previous experience was gained. In other words: one simulation run was of the form $T_1\ T_2\ ...\ T_{10}\ T_1T_2\ ...T_{10}$. The agent was permitted to learn in between the first ten tasks (i.e. update its mental model after the task completion). After the first ten tasks, the agent's mental model was restarted back to its initial state and was similarly restarted after each subsequent task performance. In this manner, one can study the agent's performance improvement by comparing it to the one the agent would display if no experience (other than initial) was gained.

In contrast to the experiment reported in Section 6.1.1.1, when simulating a sequence of various tasks, it should not be expected that successful performance on one task is followed by successful performance in every subsequent task. Thus, the analysis of the results did not make any success-based distinction among simulated runs. **Figure 6.7** and Figure 6.8 present the change in success rate and steps distributions over the simulated task sequences. As evidenced by figures, the agent's performance improves over the course of the simulated run. While success rate remains approximately 34% after the 5th task, the mean number of steps required to find a solution drops with almost (as 7th and 8th task have very similar means) every subsequent task performed (Table 6.2).

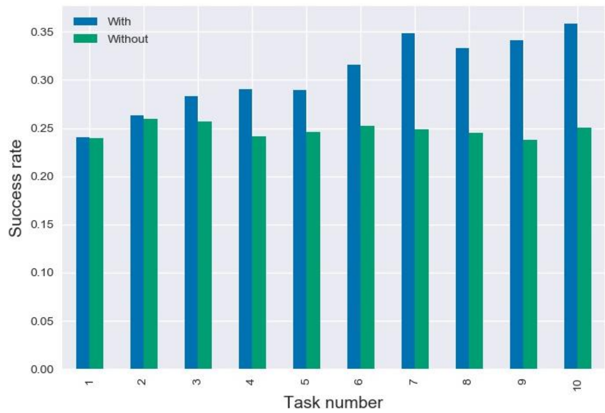


Figure 6.7 Agent's change in performance in terms of a success rate

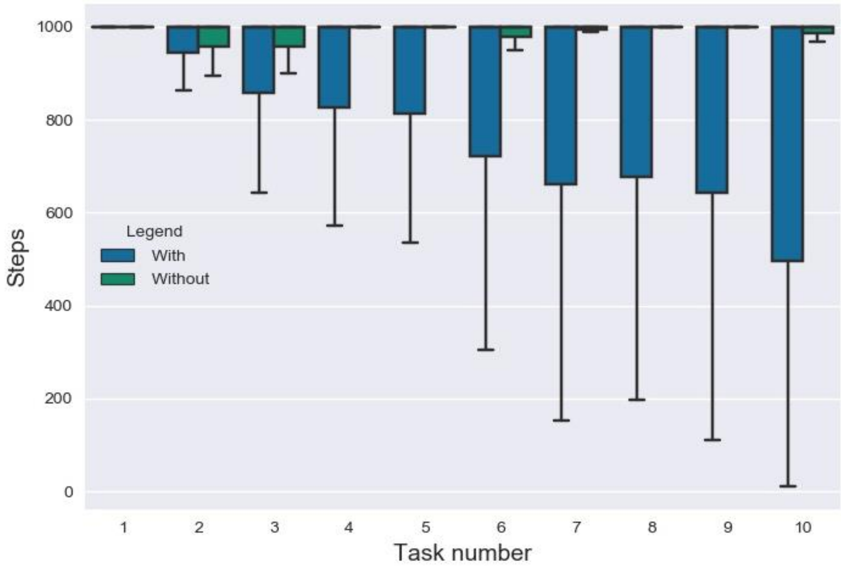


Figure 6.8 Agent's change in performance in terms of the number of steps over a sequence of different tasks

To test whether the distributions of steps required to find a solution significantly differ among cases, pairwise Wilcoxon sign rank tests with continuity correction were conducted, and Holm–Bonferroni correction method for multiple comparisons was used. Results of the tests are reported in Table 6.3. As seen in the table, p-values for pairwise comparisons indicate that in most of the cases, one can conclude - at the level of significance of 0.05 - that distributions differ. In other words, on average, the number of steps required to finish the task drops as the agent gains experience. Several task cases were not found to be significantly different from their immediate predecessors or successors, but a general trend shows the improvement of performance over the course of a simulation run. The increase in the percentage of successful tasks and a decrease in the number of steps needed to finish the task serve as evidence of the agent's learning and experience gain.

Table 6.2 Statistics on the number of steps needed to perform a sequence of different tasks

Task	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Min	84.0	62.0	52.0	8.0	17.0	30	13.0	25.0	23.0	10.0
1st Q	1000.0	945.2	857.5	827.0	813.8	721.2	661.0	678.2	643.8	496.0
Median	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Mean	894.1	876.7	857.1	841.0	838.2	807.8	784.8	786.5	777.5	758.0
3rd Q	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Max	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

Table 6.3 The p-values for each of the pairwise Wilcoxon tests conducted to compare the differences in steps distribution for each task in a sequence

Task	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
2nd	0.0126								
3rd	1.2e-05	0.0232							
4th	5.2e-08	0.0010	0.1278						
5th	2.3e-08	0.0005	0.0953	0.4371					
6th	6.2e-16	3.0e-09	6.0e-05	0.0039	0.0059				
7th	0.0000	0.0000	3.1e-11	4.9e-08	1.1e-07	0.0041			
8th	0.0000	1.3e-14	6.4e-09	3.2e-06	6.3e-06	0.0289	0.7599		
9th	0.0000	0.0000	5.1e-11	5.6e-08	1.2e-07	0.0032	0.4420	0.2001	
10th	0.0000	0.0000	0.0000	1.3e-12	3.6e-12	5.2e-06	0.0324	0.0062	0.0495

* Non-significant values ($p > 0.05$) are marked red

6.1.2 Inhibition

To test whether the implemented inhibition mechanisms affect the agent's performance, a comparison of agent's behaviour on a sequence of five tasks during which inhibition was enabled was compared to the performance on the same set of tasks when inhibition is not included. More precisely, each simulation run was formed as follows: a sequence of tasks consisted of 5 different tasks, each performed twice: $T_1 T_2$ ... $T_5 T_1 T_2$ ... T_5. During the first five tasks the inhibition mechanism was included in the agent's mental model. Then, the agent's mental model was restarted and the agent again performed each of the five tasks, but this time without the inhibition mechanism enabled. Note, during both of the five tasks long sub-sequences, the agent was permitted to learn in between tasks. The experiment consisted of 2,000 simulation runs.

Figure 6.9 presents the difference in success rates caused by the enabling (or disabling) the inhibition mechanism. It can be observed that omission of the inhibition mechanisms prevents the agent from building on past experiences and current perceptions to improve its success rate. On the other hand, if the agent can inhibit the structures not suitable for the task at hand, and use memories to set initial inhibitions on the start of each task, the agent improves its performance on each subsequent task. Correspondingly, in case of enabled inhibition mechanism, the number of steps needed to find a solution continuously decreases as the simulation progresses (Figure 6.10).

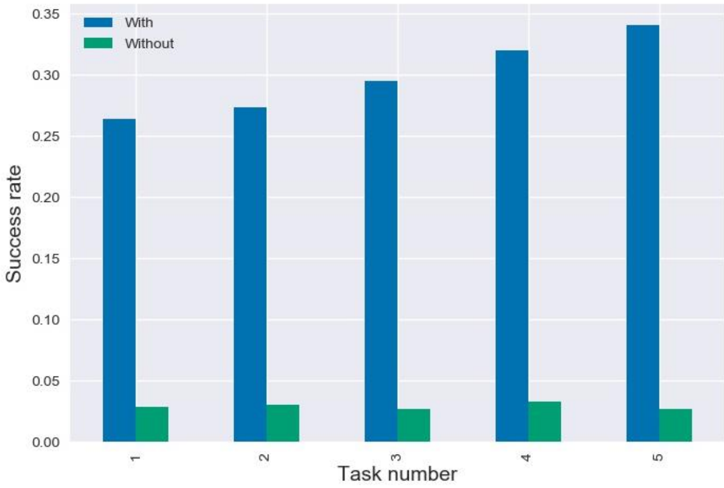


Figure 6.9 The change in a success rate in cases with and without inhibition mechanism enabled

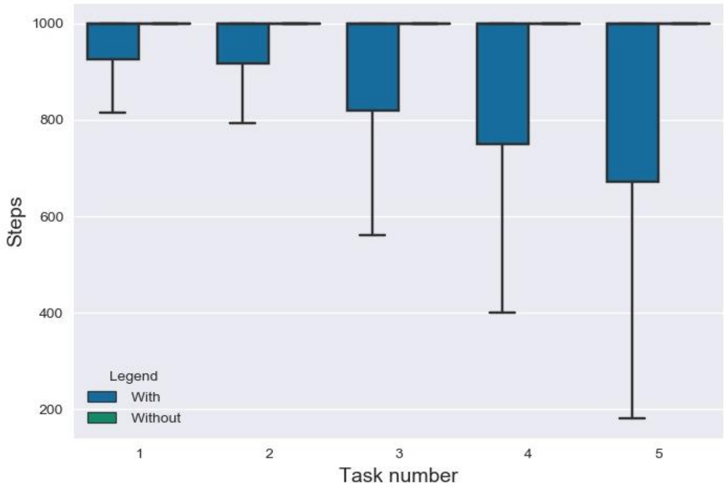


Figure 6.10 Change in the number of steps required to solve a task with and without inhibition mechanism enabled

To further study the effect of inhibition mechanism, particular attention was given to the agent's behaviour regarding solution space exploration. Figure 6.11a presents the distribution of the number of new structures created over the task sequence. In both cases – with and without inhibition enabled – the number of new structures created tends to decrease as the agent gains experience. In the first simulated task, the agent without inhibition mechanism creates more structures than the (otherwise similar) agent with inhibition mechanism enabled. These differences result from the tendency of the 'with-inhibition' agent to dismiss the structures which are deemed as unsuitable, while the 'without-inhibition' agent keeps the structures active for a longer period and, thus, increases the likelihood of modifying or combining activated structures into new ones. However, the ‘without-inhibition’ agent is unlikely to switch its attention towards newly generated structures. Instead, it will likely remain focused on the first-activated (and often unsuitable) structure. Although the ‘with-inhibition’ agent does not create as many structures, the inhibition enables it to search a more diverse space of solutions: if one solution is inhibited, the agent’s attention shifts towards other, different (in terms of properties) structures, until the suitable solution is found. Such search of a wider space creates a basis for the ‘with-inhibition’ agent to find appropriate solutions in subsequent tasks: although the number of new structures generated decreases as a task sequence progresses, the overall success rate increases. Thus, as more experience is gained, the ‘with-inhibition’ agent manages to find a suitable solution more often and does so with less solution-space exploration.

The ‘without-inhibition’ agent, on the other hand, is less likely to learn about the new structures (due to its continuous focus on the first-activated one). This, in turn, means that the links relevant for the focused structure will get even more grounded, increasing the likelihood that the agent will focus the same structure in subsequent tasks as well. Such structure will probably be unsuitable – leaving the agent in a deadlock in which no new structures can be created (the structures generated from the focused structure have already been made in previous tasks), and attention cannot be shifted. Due to these processes, the ‘without-inhibition’ agent gradually starts proposing a smaller number of distinct structures (Figure 6.11b).

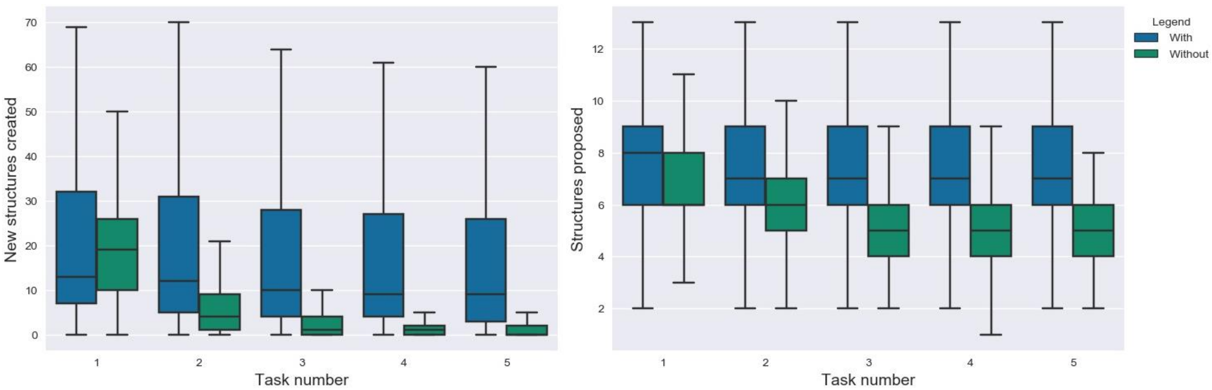


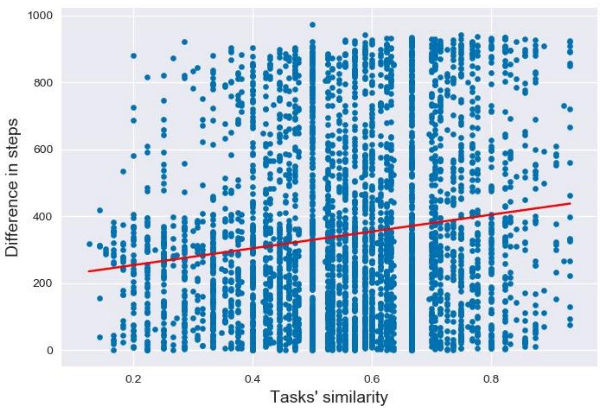
Figure 6.11 The comparison of effect of inhibition mechanism on solution space exploration in terms of a) the number of new structures created, and b) the number of distinct structures proposed during the task

6.1.3 Base-level activation

The similar experimental setup as that conducted for the studies of inhibition effect was used to study the effect of base-level activation on the agent’s performance. In other words, the simulation design in which simulation runs were of the form: $T_1T_2 \ldots T_5T_1T_2 \ldots T_5$ was reused. During the first five task performances, the agent was permitted to initialise the base-level activations for each known node based on the recollections of the past tasks and its expertise. After the completion of the first task T_5 the agent's mental model was restarted, and the base-level activation mechanism was switched off for every remaining task in a simulated sequence. Again, 2,000 simulation runs were performed.

In cases in which the agent had no previous experience and base-level activation mechanism was enabled (i.e. first T_1 task), the base-level activations were based solely on the agent's expertise area. In each of the subsequent tasks of the first subsequence ($T_2 \ldots T_5$), both, agent's expertise area and past tasks performed were used to set base-level activations.

Based on the implementation developed in the previous chapter, it can be expected that the impact of base-level activation is correlated with the similarity among the current task and past experiences and expertise. However, such impact may not necessarily be positive: if one is primed with very similar tasks, the memories of the past solutions - which were suitable in plenty of past tasks, but are not solutions for the current task - may be detrimental. Thus, the first experiment studied the correlation among similarity (of the current task and past experiences), and the difference in steps required to perform tasks with and without base-level activation mechanism enabled. The absolute difference in steps between two conditions was hypothesised to be significantly related to the similarity among past and current tasks. The results obtained are presented in Figure 6.12a. For clarity, Figure 6.12b presents the same data in the form of boxplots. Using Spearman's rank correlation test, a very week ($\rho = 0.117$), but statistically significant ($p < 0.001$) correlation was found.



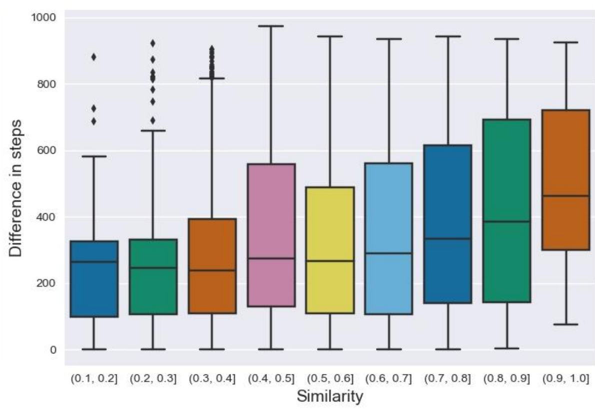


Figure 6.12 The relation among difference in steps needed to find the solution with and without base-level activation mechanism, and similarity among past experiences and the current task a) scatterplot b) boxplot representation

Then, the data were classified based on the ‘polarity’ of the base-level activation’s influence: cases in which agent without base-level activation perform better (i.e. base-level activation had a negative influence) were separated from cases in which agent with base-level activation enabled was more successful. The scatterplots for each case are reported in Figure 6.13. In cases in which agent performed worse when base-level activation is enabled, the statistical analysis using the Spearman’s rank correlation test found no significant relation between similarity (of past experiences and current task) and step difference. Regarding the cases where agent benefited from base-level activations, a weak ($\rho = 0.151$), but significant ($p < 0.0001$) relation was found.

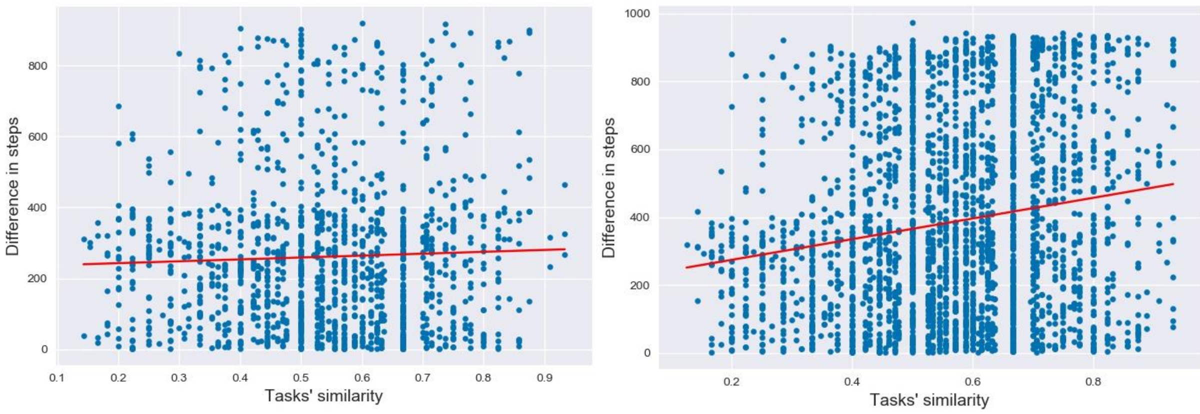


Figure 6.13 The relation between similarity among tasks and difference in steps introduced by addition of the base-level activation to the agent’s mental model; a) Cases where base-level activation had a negative effect, b) Cases where base-level activation had a positive effect

Finally, the data were analysed to extract the success rates and step distributions for each task of the two simulated conditions. The results are presented in Figure 6.14.

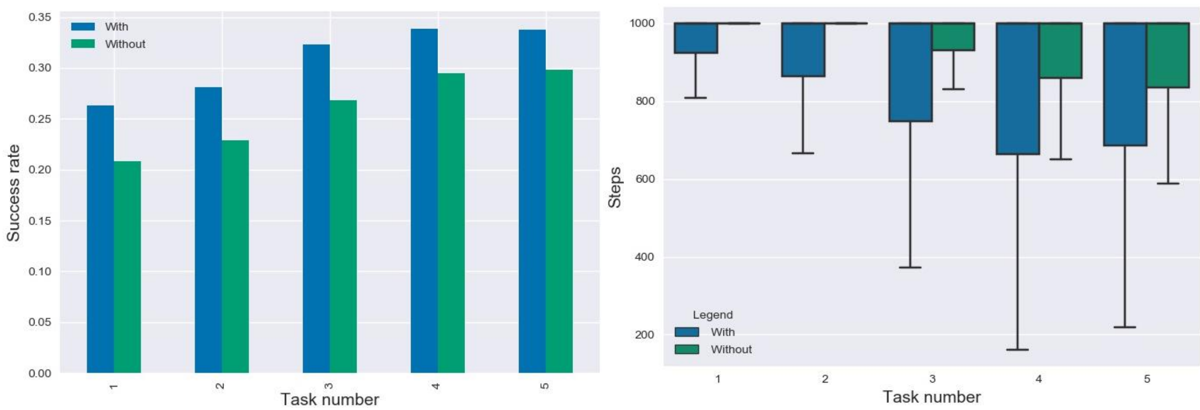


Figure 6.14 The change in performance regarding success rate (a) and number of steps needed to finish the task (b) in cases where base-level mechanism was and was not enabled

As can be observed from Figure 6.14, the agent manages to improve its performance regardless of the base-level activation inclusion – thus demonstrating how learning in terms of new knowledge elements creation and link grounding affects the agent's success on the tasks. But the figure also shows how base-level activations add to the effect of the learning process and positively influence the agent's task performance. Nevertheless, future experiments should study more closely the conditions in which memories are beneficial – and in which they may hinder the performance.

6.1.4 Cognitive ability

To study the effect of cognitive ability (and a related parameter of focus capacity) on individual agent's performance, several experiments were conducted. In order to gain better understanding of the influence of the focus capacity, the experiments included two values outside of the permitted focus capacity range: 2 and 6.

The first experiment aimed to study how different values of focus capacity influence the success of an individual agent on a single task. The simulated task did not include any prior learning; i.e. the simulated agent used only the initial knowledge and expertise to perform the task. Similar to the studies reported in previous subsections, to study the relationship between agent's cognitive ability and task performance, the overall score (i.e. value stating whether agent failed, or found and accepted a solution) and the steps needed to perform the task were collected. In the designed simulation experiment, each simulation run consisted of five repetitions of a single task where the agent's mental model was restarted at the end of each. Further, in between two task performances, agent's focused capacity was gradually increased. More precisely, one simulation run consisted of a sequence: $T_1T_1T_1T_1T_1$, where agent's cognitive ability (the first pair value) and focus capacity (the second pair value) were changed as follows: (0.01, 2) in the first, (0.168, 3), (0.5, 4), (0.832, 5) in second, third and fourth, and (0.99, 6) in the fifth task performance. Overall, 5,000 simulations were run.

Figure 6.15a presents the success rate for each cognitive ability case. As can be deduced from the success rates, the median value of steps needed to perform the task equals the number of allocated steps (i.e. 1,000) in each case. The distribution of the steps needed to perform the task in cases in which the solution was found and agreed upon (i.e. 'successful tasks') is presented in Figure 6.15b.

The hypothesis that agents with higher focus capacity levels reach a solution in fewer steps (in cases in which a solution is found) is tested using pairwise Wilcoxon signed-rank tests – all of which confirmed a statistical significance of the differences ($p < 0.05$).

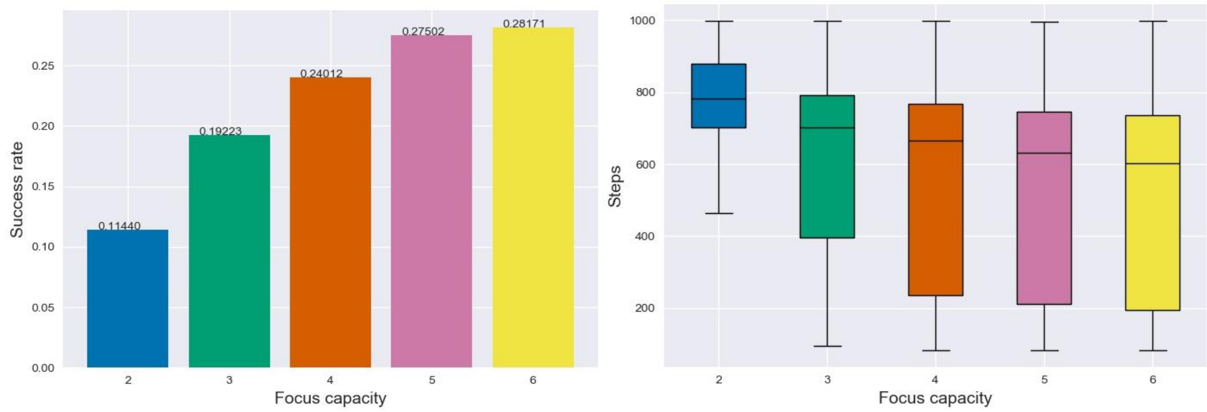


Figure 6.15 Success rate (a) and distribution of required steps in successful trials (b) on a single task performed by agents with different focus capacity levels

To further study the impact of distinct cognitive ability levels on the agent's performance, a change of agent's performance over a sequence of tasks was tracked. Similar to the conducted study of agent's learning over a task sequence (Section 6.1.1), separate analyses studied the agent's performance on a repeated task and agent's performance over a sequence of different tasks.

Figure 6.16 presents the agent's performance over ten repetitions of a single task. Figure 6.16a shows an improvement in a success rate, while Figure 6.16b for clarity represents only the average number of steps required to complete the task. As can be seen on the figure, as the agent repeatedly performs the task, its performance improves while following an exponential learning curve, which is compliant with theoretical proposals [318]. The slope of the performance improvement varies with the change in focus capacity (and cognitive ability): more capable agents show more considerable improvement over time.

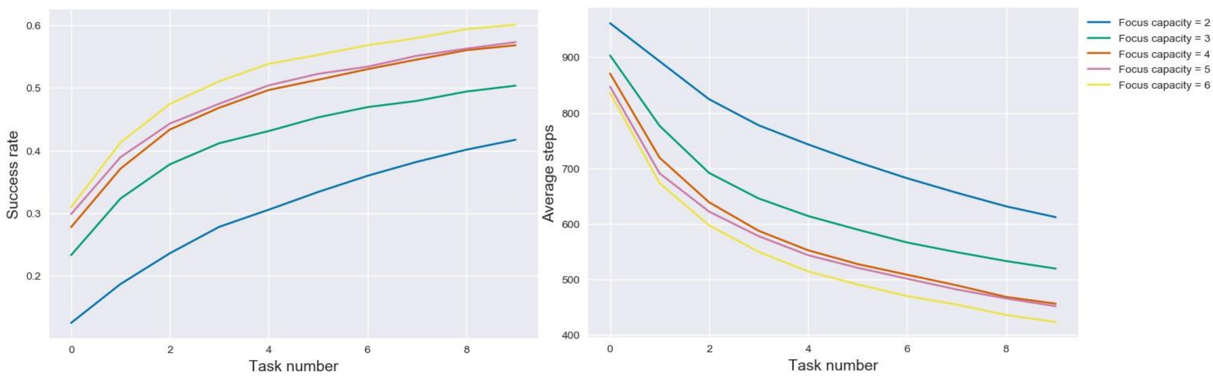


Figure 6.16 The relation among cognitive ability/focus capacity and a change in (a) success rates, and (b) the average number of steps needed to finish a task over multiple repetitions

Similarly, when tracked over a sequence of ten different tasks, all of the agents (i.e. the agent's instances diversified solely based on the cognitive ability/focus capacity parameters) display the performance improvement. This serves to show that all of the agents are capable of learning. However, cognitive ability influences the learning rate. To clearly depict the differences in the performance improvement over the course of the simulated task sequence, Figure 6.17a represents the increase in the success rate of each agent benchmarked against the performance it would display if no previous tasks were performed. The least competent, 'theoretically implausible' agent – which is capable of focusing only two knowledge chunks in a single simulation step - improves its performance by solving approximately 3% more tasks (in comparison to the success rate it would display if it had no experience). On the other hand, the most capable agent (whose focus capacity is also outside of the permitted range and equals six knowledge chunks) learns sufficiently to solve up to 20% more tasks than it would if it had no experience. The performance improvement of the agents with plausible values of focus capacity (i.e. from 3 to 5) ranges from 7.5% for the least capable agent, to slightly more than 15% for the highly capable agent. The trend in change of the average number of steps needed to finish the task is in accordance with the improved rate of successful task completion (Figure 6.17b).

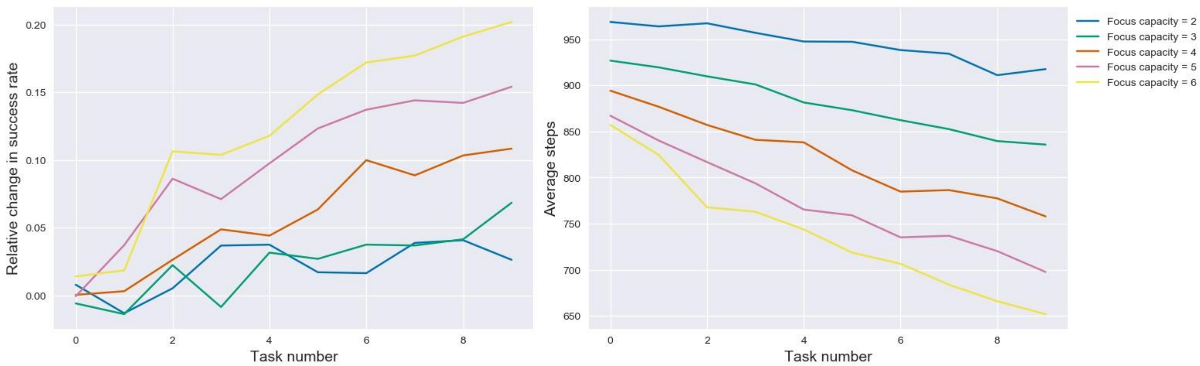


Figure 6.17 The relative change in success rate (a) on a sequence of different tasks and the average number of steps needed to finish the tasks for each focus capacity value

The simulated experiments demonstrate how cognitive ability mechanism captures the differences in learning among agents. If their level of knowledge and experience is similar (as in the first simulated task – see Figure 6.15a), highly capable agents are more likely to find a solution. Additionally, the more capable agent will make a better use of subsequent experiences by managing to learn more and using the obtained knowledge to improve its performance further. In other words, the less capable agent will have to put in more effort (i.e. will be required to gain more experience) to match the performance level of its more capable counterpart. These results are in line with the findings that cognitive ability is one of the strongest predictors of task success [226], [233], [319].

6.1.5 Affect

To test whether the implemented affective mechanism has an impact on agent's performance, a set of 2,000 simulations were run. Each simulation consisted of two repetitions of a single task: one in which affective module was enabled, and the second in which the affect was not taken into account. Thus, the simulation runs consisted of a sequence $T_1 T_1$. To enable comparisons, the agent's mental model was restarted in between two task performances.

First, a success rate the agent obtained in each of the conditions was extracted. The agent managed to find the solution in 25.3% of the simulation runs in which the frustration variable was allowed to change. In the cases in which the agent was unaffected by the frustration value, the success rate was 13.9%. Wilcoxon signed ranked test confirmed the statistical significance ($p = 0.0123$) of the differences in distributions of steps needed to find a solution. Thus, at the level of significance of 0.05, an alternative hypothesis stating that affective agents take less time to finish the task can be accepted. These results imply that a frustration parameter indeed influences the agent's behaviour by directing it towards finding the solution.

To further study the impact of the affect model and implementation, the agent's performances were separated based on the maximal frustration value achieved during the task performance. In other words, for each of the simulated tasks in which affect was taken into account, the maximal value of frustration the agent experienced was recorded. The obtained maximal frustration value was used to separate the data into nine bins of approximately equal size: i.e. each containing 220 simulated pairs of tasks (with and without frustration considered). The boxplots shown in Figure 6.18 demonstrates the differences between the two tasks' settings regarding the number of steps needed to finish the task - with respect to the maximal frustration value achieved. As can be observed in the Figure 6.18, the tasks which were 'easier' – and, thus, the frustration levels were low – were shorter (in terms of steps needed to find a solution) in cases where the affective component was enabled. However, when the agent's frustration level was high, the affective agent's performance (regarding steps needed to conclude the task) was not different than that of an affectless agent. Wilcoxon signed ranked test applied to cases where frustration did not exceed 0.2 confirmed the statistical difference between the two simulated settings ($p = 0.0004$). The collected data does not provide a distinction among the remaining cases with (i.e. in which frustration level is greater than 0.2) and without affective component. This finding is not surprising as the implementation intertwines high frustration level and task failure: if the task is difficult to solve, the frustration level rises. Additional experiments will be conducted in the future to further study the effect of high frustration levels on task success.

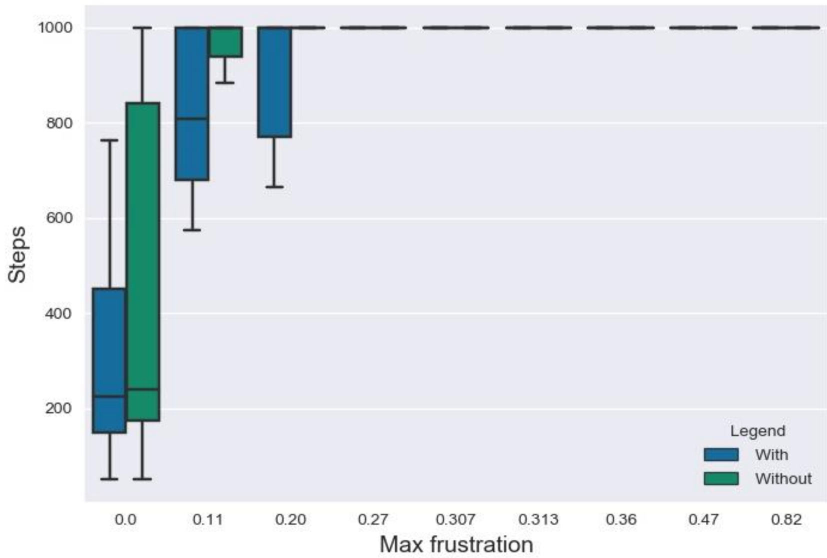


Figure 6.18 Task duration distribution with and without affective component enabled, with the respect to the maximal level of frustration experienced when affect is enabled

To study the effect of frustration on the agent's solution space exploration, the data were inspected to detect if agents with the affective component focus more on the solution-finding. Figure 6.19 shows the differences regarding the number of new structures created (a) and the number of distinct structures proposed (b) between the affective and affectless agents' behaviour. Figure 6.19 demonstrates how agents with lower levels of frustration generate a relatively large number of new structures. As the frustration level increases, the number of newly created structures decreases. This behaviour is in line with the theoretical background laid out in Section 4.4.2, e.g. [203], [219]. However, it is important to note that in the current model implementation, the low frustration level does not result in a high number of new structures. Rather, it is the other way round: the 'easier tasks' are those closer to the agent's knowledge. Thus, more knowledge elements get activated, which in turn facilitates the creation of a large number of structures. These structures are likely to display at least some of the required properties. Therefore, the agent's level of frustration remains low. In short, the negative correlation between frustration levels and the number of newly generated structures is a result of task difficulty. The behaviour of affectless agents confirms such statement: although its frustration level does not change, the number of new structures follows the same trend as that of the affective agent.

Despite the similarities of structure generation trends, there are differences in the overall number of new structures created. Wilcoxon signed rank test was employed to study the differences in the creation of the new structures for each of the frustration levels. The results showed that irrespective of the frustration level taken, at the level of significance of 0.05, the affective agent generates more new structures than the agent unaffected by the emotions. The obtained results demonstrate the effect of frustration on the solution space exploration: due to an increase in a frustration level, the affective agent focuses more on the solution space, which increases the number of newly generated structures. Similar results can be obtained when studying the differences in the number of distinct structures proposed. At the level of significance of 0.05, for every frustration level can be concluded that the affective agent proposes more structures than the affectless agent. These results demonstrate how affective state can motivate the agent to elicit more effort in finding a solution [218].

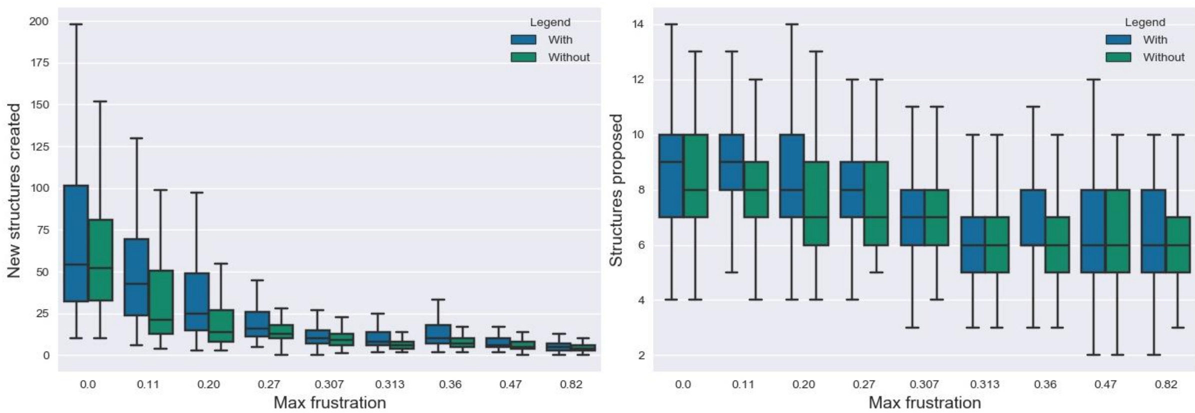


Figure 6.19 The difference in agent's solution space exploration behaviour – in terms of the number of new structures created (a) and the number of distinct structures proposed - when affective component is and is not enabled

The affect component is intended to influence task dynamics. Thus, an analysis was directed towards studying the change in the content of messages communicated by the agent over the task course. Particularly, the number of distinct structures proposed, as well as the ratio of problem- and solution- related messages were studied. Figure 6.20 represents a typical change of the PS indicator [320] for affective and affectless agents over the course of an unsuccessful (a) and a successful (b) task. PS indicator is a measure that relates to the focus of design activity. It is calculated as the ratio of the difference in a number of problem- and solution-related messages, and the overall duration of the design task. Thus, it ranges from -1 to 1, and positive values indicate a problem-oriented design activity (while negative signals solution-oriented activity).

To create graphs shown in Figure 6.20, first, a set of tasks where frustration had an impact had to be extracted. Thus, simulations for which more than 500 steps (i.e. 50% of the simulated task duration) were needed to find a solution were selected and classified based on the success of the task. Then, a PS indicator was calculated using the moving window whose size was set to 10% of the overall task duration. Finally, the obtained PS indicators were averaged over each of the time steps.

The results demonstrate the differences in behaviour between two simulated agents. Similar to the affective agent, the affectless agent spends the first task period exploring the problem space. Then affectless agent's PS indicator value stabilises showing a constant ratio of the number problem and solution related issues (among which solution-related design issues are slightly more prevalent). Affective agent, on the other hand, starts changing its behaviour (diverging from the affectless agent) once the frustration begins to influence its mental model. Due to an increased effort to find a solution, the affective agent reconsiders the past solutions (see Section 5.1.2), determines their insufficiencies and as a consequence relates back to the unsatisfied requirements. In other words, the agent shifts back to the problem space exploration. Such shifts may lead to progress, which in turn enables solution finding (as shown in Figure 6.20b).

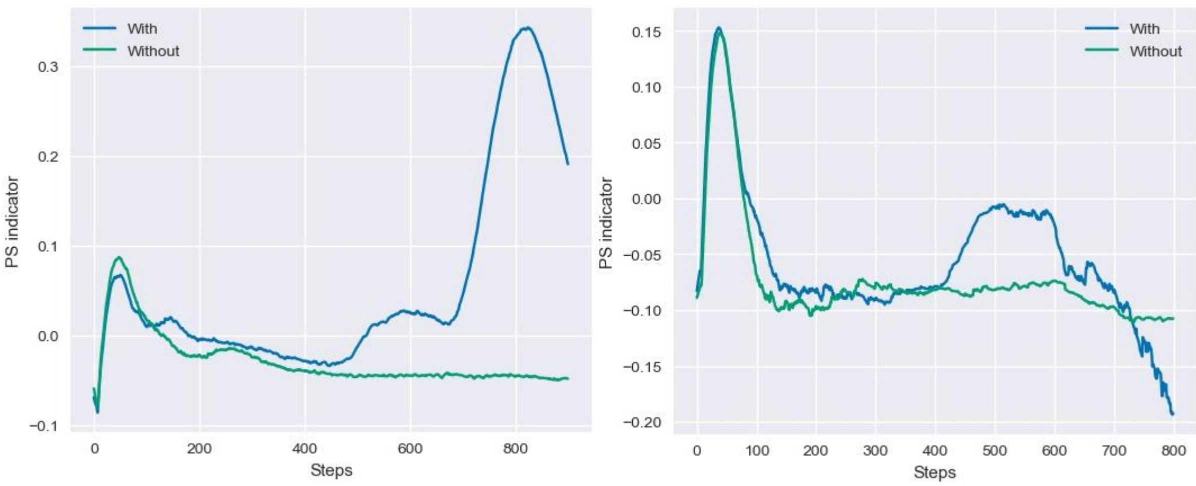


Figure 6.20 Moving PS indicator for a) unsuccessful, and b) successful tasks

In other cases, however, the agent becomes aware of its inability to satisfy the requirements, and its frustration continuously increases. As a result, the relevant requirement-function and function-behaviour links are repeatedly focused and further grounded, while the structures connected to focused behaviour nodes get more inhibited (since they do not meet all of the requirements). These processes in sum stop the agent from shifting its attention towards structures which – albeit less related to the problem – could serve in the generation of the suitable solutions (through union and concatenation processes). In other words, the agent enters a deadlock in which it is unable to shift its attention towards more fruitful knowledge paths. This process could be understood as what Norman [203] refers to as *tunnel vision*.

To further study the dynamics of the affective agent's behaviour, the number of distinct structures proposed in the first and the second half of the simulation were collected (Figure 6.21). Since the agents are not affected by the frustration value during the first half of the task, the number of proposed structures has a similar distribution in both affective and affectless, cases. In the second simulation half, however, the affectless agent proposes significantly fewer structures than its affective counterpart. On the other hand, both – affective and affectless – agents propose a smaller number of structures in the second simulation half. This behaviour is in line with the view of initial task stages as a more *creative stage* of the design process [203].

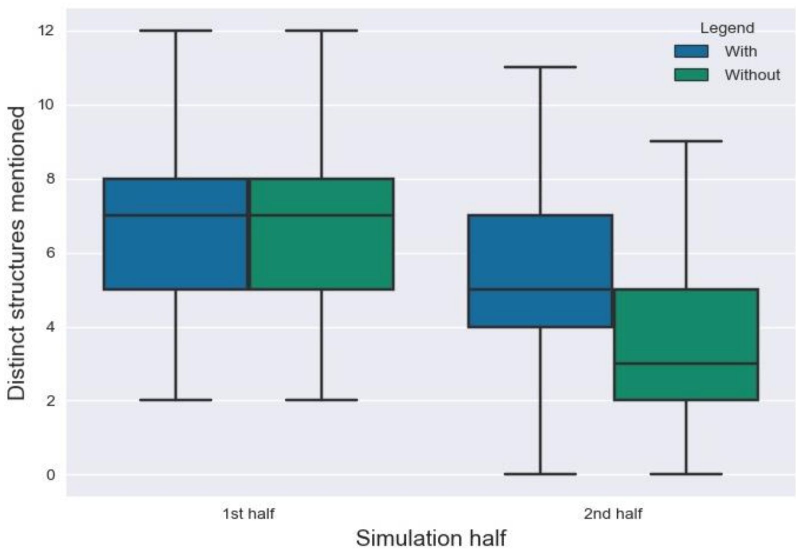


Figure 6.21 The number of distinct structures mentioned during the first and the second half of the simulated task

A final analysis studied the portion of each of the agent's message types in different phases of the simulated task. To account for the differences in tasks' duration, each task was represented in terms of percentiles. Then, the number and type of the messages communicated during each of 10% task intervals were collected, which enabled calculation of the relative frequency distribution of message types for each of the 10% intervals. The results of such analysis are presented in Figure 6.22. The results obtained for the affective agent demonstrate its effort to converge to a solution: as the second half of the task progresses, the agent spends more time proposing and evaluating the structures. The behaviour of the affectless agent remains undisturbed by the approaching deadline. Thus, its distribution of communicated message types remains stable throughout the simulated task. Such behaviour of the affectless agent is difficult to validate: can anyone have a constant motivation but be unaffected by emotions and drives? Nevertheless, the obtained behaviour matches the expectations based on the developed model implementation. Thus, the obtained results increase the confidence in the correctness (in terms of the absence of bugs) of the implemented mechanisms.

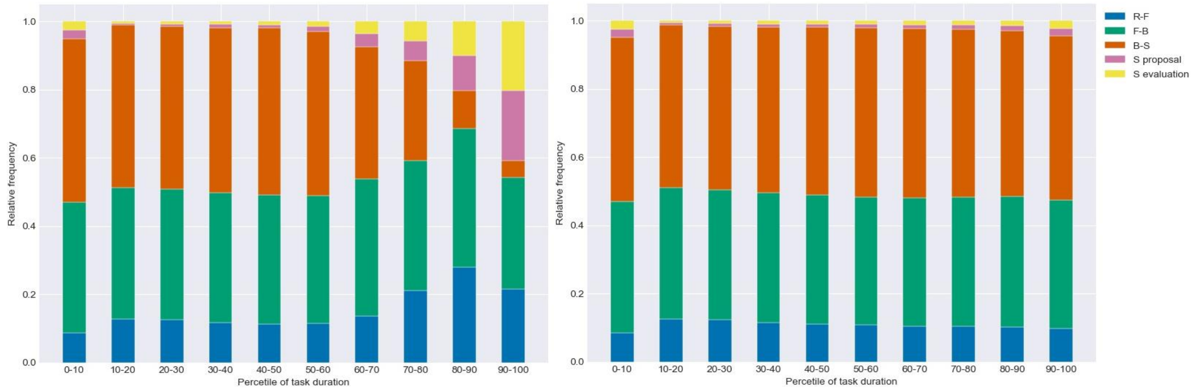


Figure 6.22 The relative distribution of the message types in a task performed by a) an affective and b) an affectless agent

6.1.6 Expertise

A final element of the agent's architecture that needs to be tested is the agent's expertise area. To study the impact of the expertise area on the task performance, a series of simulation runs were conducted and diversified based on the level of overlap (i.e. similarity) between the simulated task requirements and the agent's expertise area. The similarity was calculated using the Dice coefficient. In the experiments reported in this subsection, the number of behaviours within the agent's expertise ranged from a single behaviour node to five behaviour nodes. The number of expertise areas was randomly drawn from a uniform distribution. The simulation runs were carried out until 1500 data points were collected for each of the similarity categories defined by the bounds 0, 0.3, 0.5, 0.7 0.9, and 1.0. Thus, in – for example, second similarity category, tasks had at least some overlap with the agent's expertise area (i.e. overlap is higher than 0), but the overall similarity was lower or equal to 0.3.

Due to the manner in which the agent's expertise area is modelled, the agent necessarily knows at least one structure satisfying the requirements within its expertise domain. Thus, it is expected that the agent will successfully finish each task that perfectly aligns with its expertise area. As the similarity among the required and the expert sets of nodes decreases, the agent will likely have less success in finding the tasks' solutions. The described experiments were aimed at testing these hypotheses.

The results regarding the overall success rate and the distribution of the number of steps needed to finish the task are shown in Figure 6.23. As seen on the figure, the success rate is correlated with the similarity of task requirements and the agent's expertise. In accordance, the number of steps needed to find a solution is higher as the knowledge required for task solving is more distant from the agent's expertise area. These results align well with the laid out expectations.

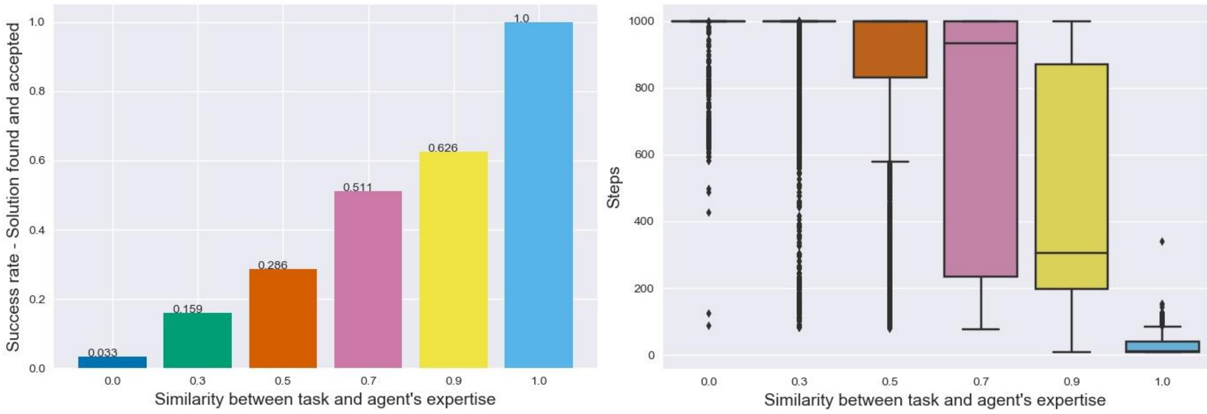


Figure 6.23 The impact of similarity between task and agent's expertise on the task performance regarding a) success rate, and b) number of steps needed to find a solution

Additionally, one can explore if the similarity of task's requirements and the agent's expertise has an impact on the agent's search of the solution space. Figure 6.24a presents the distributions of the number of new structures created for each of the defined similarity categories. Since the agent already knows the solution to the task perfectly aligned with its expertise area, when faced with such task the agent does not create a large number of structures. Instead, it simply proposes some of the already known solutions. This behaviour is also evidenced in Figure 6.24b where the median number of structures proposed (in case of the perfect task-expertise match) equals one.

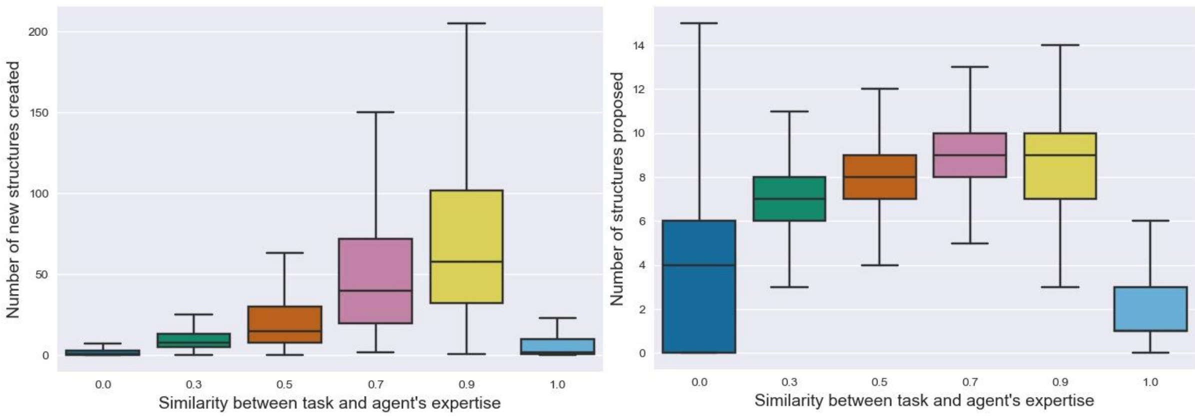


Figure 6.24 The relation between similarity of tasks and agent's expertise and agent's solution space exploration behaviour regarding a) the number of new structures created, and b) the number of distinct structures proposed

If the task differs from the expertise area, the agent is forced to search the space in order to find a suitable structure. The similarity among required and expertise behaviour nodes provides the agent with multiple (relevant) knowledge elements on which it can build its search for the solution. Thus, the more similar the task is to its expertise, the more (already known) knowledge elements can the agent use – consequently generating a large number of potential solutions. With a decrease in task-expertise similarity, the amount of the agent's initial knowledge useful for the task decreases and restricts its search process. The distributions of the number of new structures created, and the number of distinct structures proposed, are significantly different for every pair of the similarity categories ($p < 0.05$) and follow the predicted trend. It is interesting to note, however, that in case of no overlap between the task and the agent's expertise the number of structures proposed shows a large variance – i.e. in some simulations the agent suggested a large number of structures. The data revealed that such occurrence is a result of the deficiencies in the agent's knowledge and frustration level: since the agent knows little about the task requirements, it knows several structures equally (un)suitable for the task. Due to low performance, the frustration level rises, eventually generating an impulse sufficient to activate these structures. The agent then proposes the structures one by one but dismisses them once they are subjected to a more thorough analysis. For the same reason, these structures are not used further to generate new structures – and, thus, the number of newly created structures remains low.

6.2 Model testing in multi-agent setting

Following the testing of a model in scenarios comprising a single agent, the additional simulation experiments were conducted to study the model's behaviour in a more complex setting: the one where multiple interacting agents are simulating a team of designers. Unless stated differently, the simulations presented in this section utilise teams of three agents working together on a task by explicating the relevant knowledge links and proposing the possible solutions (see Section 5.2).

This section commences with an experiment directed towards studying the implications of a chunk formation implementation (as described in Section 5.1.1). The second subsection is dedicated to a study of the impact of communication on team behaviour. The third subsection examines the effect of personality (i.e. extraversion and agreeableness) on the team performance, and finally, the fourth subsection deals with the effect of trust on the team behaviour.

6.2.1 The number of focused nodes in one simulation step

As already discussed in Section 5.1.1, no upper bound on the chunk size was posed. This may create an unrealistic situation in which a large number of knowledge elements are simultaneously active in the agent's mental model. This issue is related to learning and, thus, can be studied by utilising simulations of an individual agent's performance. However, the primary purpose of the developed model is to simulate teams. Thus, it is important to collect the data from both settings (i.e. work in isolation and teamwork).

The experiment consisted of a series of 1,000 sequences, where each sequence contained ten different tasks (in between which the agents were learning). The tasks were performed by the agent working in isolation, as well as by the team of three agents. The data regarding the number of focused nodes were collected for each step of every simulated task. Figure 6.25 presents the results of such analysis. Although in both settings the agents focused ten or fewer knowledge nodes in the majority of simulation steps, one can also note several instances in which agents focused a very high number of nodes in a single step - up to 40. Additionally, Figure 6.25 demonstrates how in team settings agents tend to (simultaneously) activate more knowledge elements than when working in isolation. These differences are likely due to the diversity of knowledge introduced by difference in expertise areas (and thus communicated content) of team members.

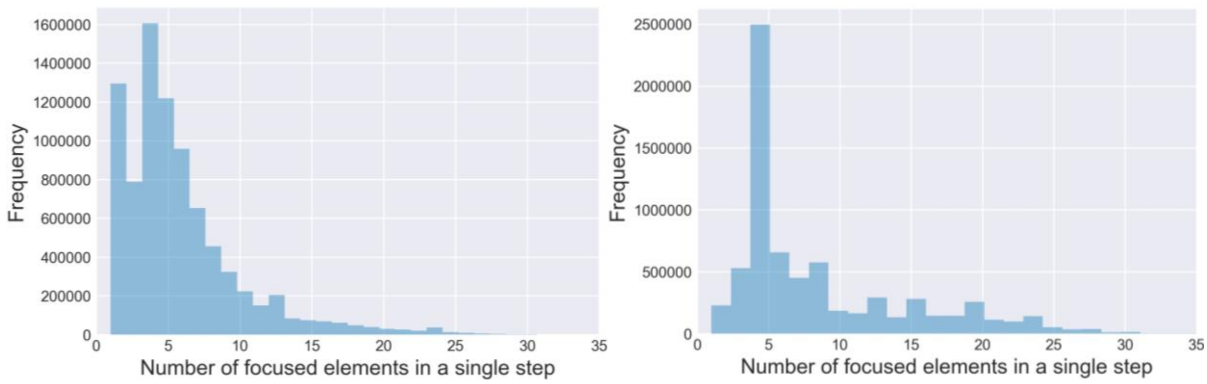


Figure 6.25 Distributions of the number of focused nodes in a single step: a) when the agent is working in isolation, and b) when agent is working in a team setting

Figure 6.26 shows the distribution of the maximal number of nodes that were simultaneously active during a simulation. As shown in the figure, the maximal number of nodes the agent will focus (in a single step) while working in isolation tends to be between 7 and 17. When working in a team setting, the agent's maximal focus in a single step typically contains between 15 and 25 knowledge nodes. Although it is difficult to determine a plausible range for the number of simultaneously active knowledge elements, the future studies will necessarily have to take a closer look at the implications and possible refinements of the current model regarding this issue.

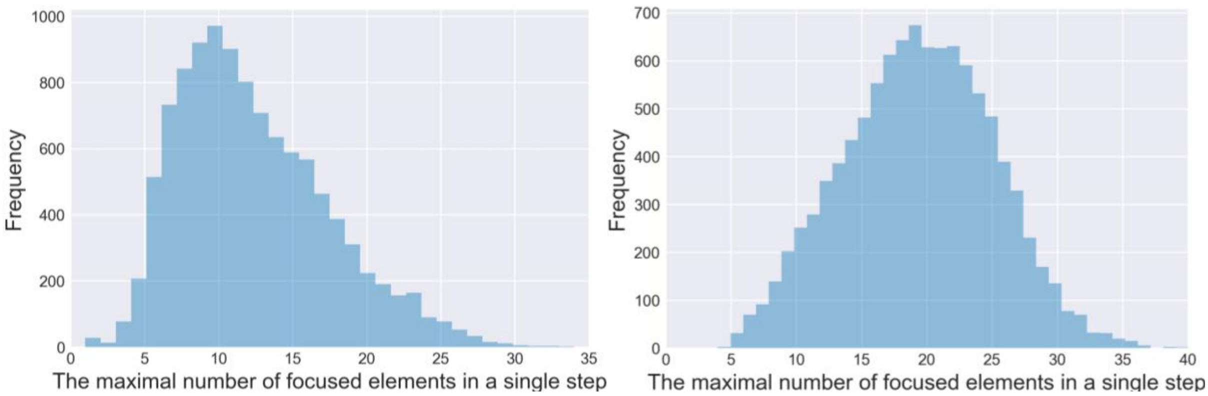


Figure 6.26 The distribution of the maximal number of nodes focused in a single step: a) when the agent is working in isolation, and b) when the agent is working in a team setting

6.2.2 Communication

The implementation of communication mechanisms was studied using animations to display the knowledge nodes' activations in each of the agents' mental models and to track if the activations occur as expected. Further, each of the aspects related to the control agent (see Section 5.2), e.g. random speaker selection, message propagation and solution evaluation and acceptance process, were tested in isolation.

To study the impact of communication on the agents' task performance, an experiment was designed as follows: while the solution proposing, evaluation and acceptance occurred as described in Section 5.2, messages containing knowledge links were not forwarded to the speaker's team members. In other words, the agents discussed the solutions as a team, but the remaining processes (i.e. processes of problem formulation and solution synthesis) were not explicated among the team members. It is hypothesised that such restriction of communication will have a negative effect on team performance by slowing down - or completely preventing – team's convergence to a final solution. Thus, each simulation run consisted of two repetitions of a task: T_1T_1. During the first task performance, the agents communicated knowledge links, solution proposals and evaluations. Then, the agents' mental models were restarted, and the task was reinitiated – but during the second task performance, the communication of knowledge links was suspended. Such experimental design enabled pairwise comparisons of the simulation outcomes.

The effect of communication restriction is likely to depend on the similarity of agents' initial knowledge. Thus, particular attention was given to the overlap in agents' expertise areas in every simulation run. The number of expertise areas was set to two for each agent within a team. There were no agents with the same set of expertise nodes, but a partial overlap was enabled. In other words, the experiments were created to enable five possible configurations of agents' expertise areas. The similarity among team members' expertise areas was calculated using the Dice coefficient.

The teams' success rates in both, finding and converging to a solution are depicted in Figure 6.27. It can be observed that agents that are allowed to communicate knowledge links perform better in every simulated configuration. Further, one can observe that the differences in success rates increase with the increase in diversity of the team's expertise areas. Namely, the difference in success rate - regarding the number of cases in which the agents manage to settle for the solution - in case of the largest expertise overlap (i.e. an instance with three distinct knowledge areas) equals 0.017 - meaning that communication enables solving 1.7% more tasks. The agents whose expertise areas are disjoined solve approximately 5% more tasks when they can communicate knowledge links.

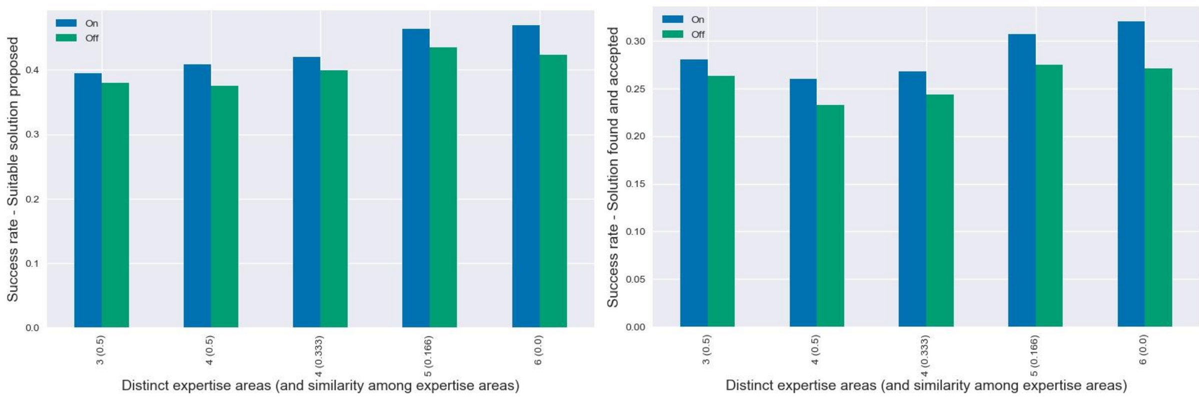


Figure 6.27 The success rates with respect to communication settings and similarity of agents' expertise areas. a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to find and settle for a solution

In addition to a success rate, one can study the differences in distributions of steps required to converge to a solution. Figure 6.28 shows relation of the expertise area overlap and the distributions of steps for the two simulated conditions (only the successful trials are filtered). The obtained findings nicely align with the proposed hypothesis. Wilcoxon signed rank test confirmed a statistical significance of the differences ($p < 0.05$) in steps required to finish the task with and without communication enabled for simulated pair of settings (i.e. for every level of expertise overlap). Additionally, the tests showed how the effect of communication inclusion/exclusion is amplified by the amount of knowledge overlap among team members.

Namely, as the number of distinct expertise areas present in the team increases, the effect of communication on the team performance became more pronounced.

Figure 6.28 The distribution of steps (in successful trials) in conditions in which communication is and is not enabled

Additionally, similar to the studies reported in ‘individual agent’ setting, one can study the differences in the team’s exploration of the solution space by considering the statistics regarding the number of newly generated structures and the number of distinct structures proposed during the task performance. Figure 6.29 presents differences in teams’ exploration of the solution space in successful tasks. One can observe how the communicating agents search the solution space to a lesser extent. When all of the tasks (i.e. both successful and unsuccessful) are taken into account, a different trend can be seen: namely, while the number of proposed structures does not differ significantly among cases (see Figure 6.30), the communicating agents tend to explore the solution space more in search of the solution. This indicates that, if the agents are not able to find the task solution, communication helps them to extend their search and consider a larger number of potential solutions.

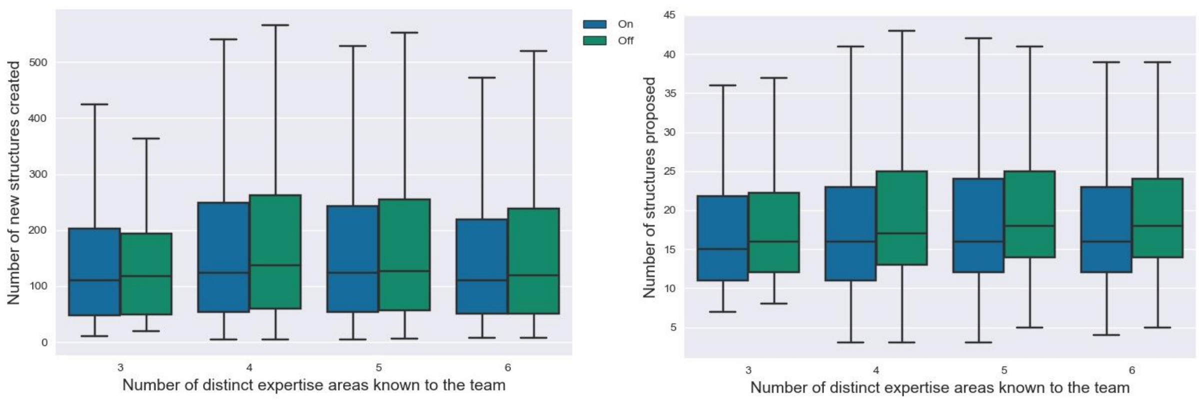


Figure 6.29 The teams’ solution space exploration behaviour in successful tasks: a) the number of new structures and b) the number of structures proposed

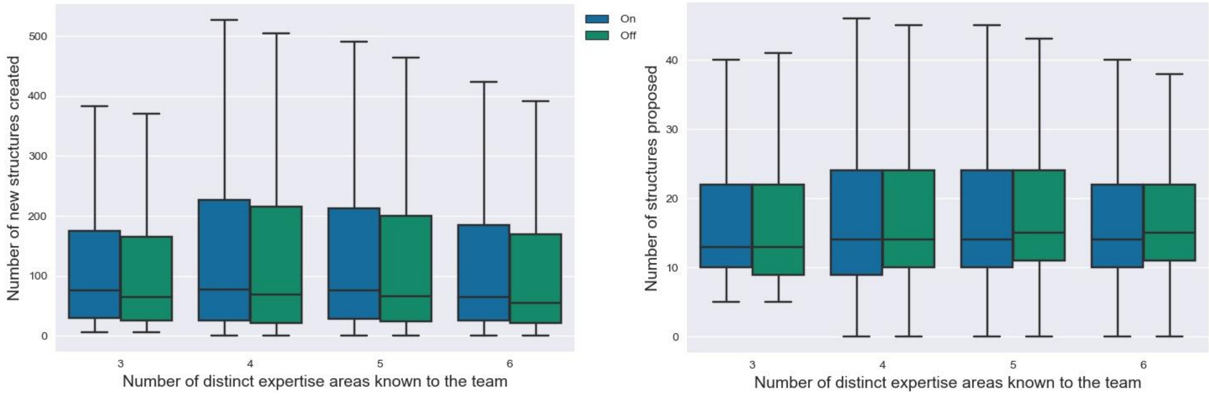


Figure 6.30 The teams' solution space exploration behaviour (all tasks considered): a) the number of new structures and b) the number of structures proposed

6.2.3 Personality

This subsection presents the results of tests directed at studying the impact of the team composition - regarding two implemented personality factors - on the overall team behaviour and performance. Default settings randomly draw agent's extraversion and agreeableness from a normal distribution. To include the boundary values (i.e. very high or very low values), in the experiments presented herein, this mechanism was modified by increasing the likelihood of occurrences of values as low as 0.01, and as high as 0.99. Overall, the simulations included 36,000 tasks performed by teams of agents of various agreeableness levels. Similarly, data on 36,000 tasks were collected to study the effect of team member's extraversion on team performance.

6.2.3.1 Agreeableness

Figure 6.31 depicts the differences in steps distributions for various values of average team agreeableness. First, the data regarding average agreeableness of the members of a team were analysed to separate the obtained data points into categories of equal size. In this manner, the boundaries used to create Figure 6.31 were derived - the value written on the x-axis below each boxplot represents the upper boundary of the agreeableness category. The data presented in Figure 6.31 demonstrates the beneficial effect of the average level of team members' agreeableness on the convergence of the team. Following the implementation presented in Section 5.1.3, these results accord with the expectations: the highly compliant agents try to build on other's ideas and are less likely to dismiss other's solution proposals – which in turn facilitates quick settling for a proposed solution. As the average team agreeableness decreases, the number of steps needed for the agents to reach the consensus increases. Interestingly, the least agreeable group shows a slight performance improvement (in comparison to the second-least agreeable group).

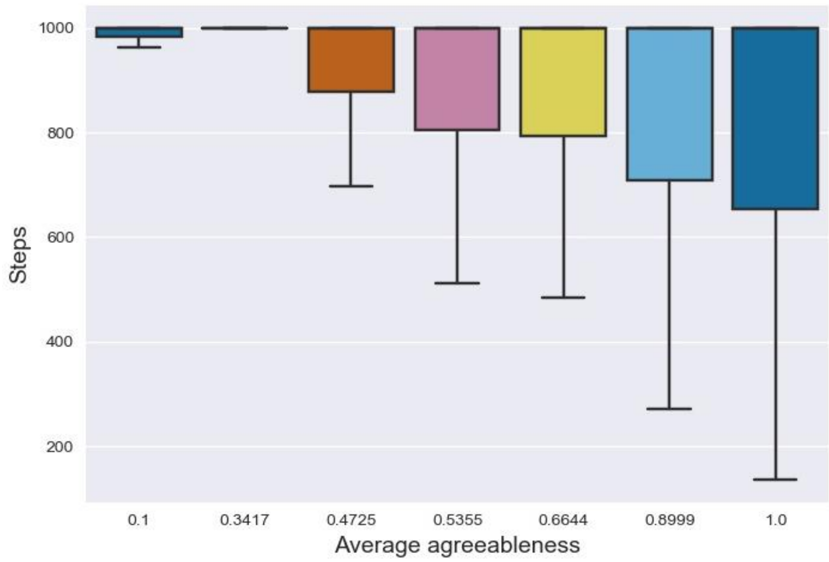


Figure 6.31 The relation between the average team agreeableness and the number of steps needed to complete the task

Figure 6.32b corroborates such findings by revealing that team's success rate (measured as the percentage of tasks in which the team agreed upon a solution) in most cases declines with a decline in average agreeableness. These results are consistent with the claims of agreeableness being one of the predictors of task success in a team setting [232]. The lowest agreeableness category does not follow this trend – as teams in this category solve more tasks than those in the second lowest agreeableness category.

However, Figure 6.32a reveals an interesting insight: if the successful task is regarded as that in which at least one agent managed to find (and explicate) the solution – irrespective of whether the said solution was agreed upon – the findings on what constitutes a good team is almost completely contrary to that obtained by observing Figure 6.32b. Namely, the agents in the second-lowest agreeableness category are the most likely to find at least one suitable solution. This capacity declines with an increase in team members' average agreeableness. In other words, more agreeable agents are more successful in recognising and adopting the suitable structures proposed by others – which is a behaviour intended to be captured by the implementation (as described in Section 5.1.3). On the other hand, disagreement among members seems to promote solution space exploration [237]. These results align well with the findings reported in [214] stating that hostility and negative interpersonal communication facilitate creativity, but undermine its implementation.

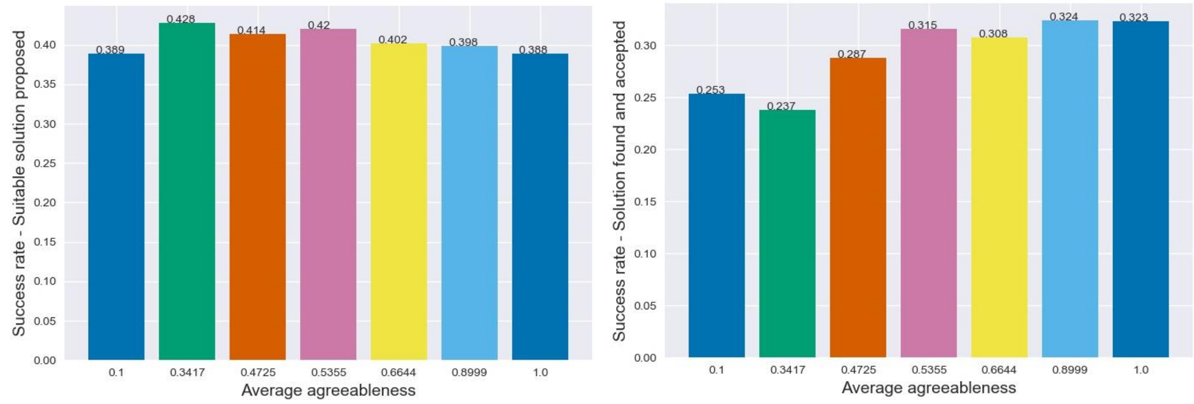


Figure 6.32 The success rates with respect to the average agreeableness of team members a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to settle for a solution

In accordance with the drawn conclusions are the data represented in Figure 6.33 which depicts the team's solution space exploration behaviour with respect to average agreeableness of its members. The level of solution space exploration declines with an increase in the average agreeableness. The two pairs of agreeableness categories were not found to be statistically different. Namely, the two most agreeable categories (i.e. categories corresponding to agreeableness ranges above the 0.6644) were statistically indistinguishable. Similarly, the pair of categories covering the agreeableness range from 0.4725 to 0.6644 was not found to be significantly different. Nevertheless, Wilcoxon signed rank test confirmed the statistical significance of differences among every other pair of categories, thus confirming the observed trend. Figure 6.33 also hints at the reason the most disagreeable teams tend to perform better than the second-disagreeable in terms of time required for solution convergence. A close inspection of the data revealed that very disagreeable agents - due to how the agreeableness is implemented – mostly dismiss other's input, i.e. their work resembles the agent's work in isolation. Since the agents are likely to have deficiencies regarding the required knowledge (to solve the task in isolation), often their search for a solution does not result in highly successful structures. Thus, they are more prone to frustration. If the frustration becomes sufficiently high, the agent starts considering the previously proposed solutions. Since the number of newly-created and proposed structures is small (Figure 6.33), the agents' will likely start considering event the structures that were submitted by others - and in some cases, this will lead to the team success. As already noted in Section 5.1.3.1, future model refinements will be directed at refining the interplay among affective mechanisms and personality.

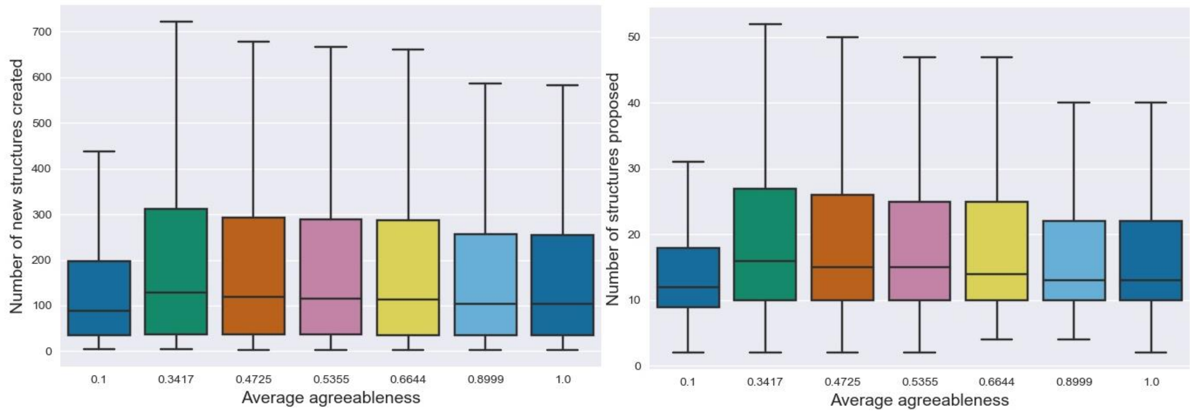


Figure 6.33 The relation between average agreeableness of team members and solution space exploration behaviour regarding a) number of newly created structures, and b) number of distinct structures proposed

Following these results one can conclude that the desired teams' characteristics depend on a task setting: if the team is working in an environment where the overall team score is taken to depend on the best-performing solution found by any of the members (as simulated in [111]), the disagreement among members may be beneficial. If, on the other hand, team members have to reach consensus (and do so quickly), the more agreeable members will perform better. However, one should note that high levels of group conformity sometimes results in team members accepting the solutions that do not satisfy all of the requirements (see Figure 6.34). Such a situation occurs if the highly agreeable agent is 'in charge' of the requirements (i.e. the requirements are close to its expertise) on which others pose low priorities. These results accord with the research published in [321] stating that high levels of agreeableness may cause premature consensus, thus negatively affecting team problem-solving.

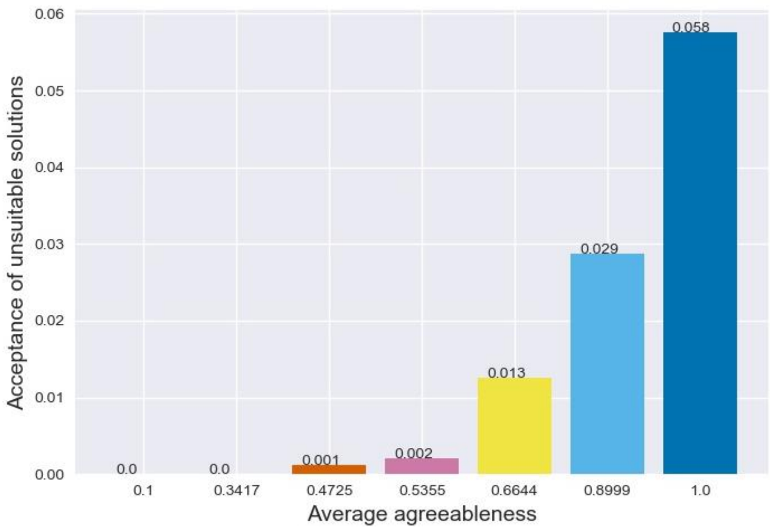


Figure 6.34 Percentage of unsuitable solutions accepted by the team

6.2.3.2 Extraversion

The extraversion parameter guides the value of the agent's sharing threshold. Thus, the first step in the assuring the implementation adequacy consisted of confirming that the agent's share in the overall team communication aligned well with its extraversion parameter and the level of knowledge relevant for the task (relative to the extraversion and expertise of others'). To study the effect of extraversion on the team performance further, similar analysis as that in the case of agreeableness was conducted. Figure 6.35 shows the distributions of steps for different values of average values of team members' extraversion. A curvilinear relationship suggesting that the best-performing team consists of 'moderate' extraverts can be detected.

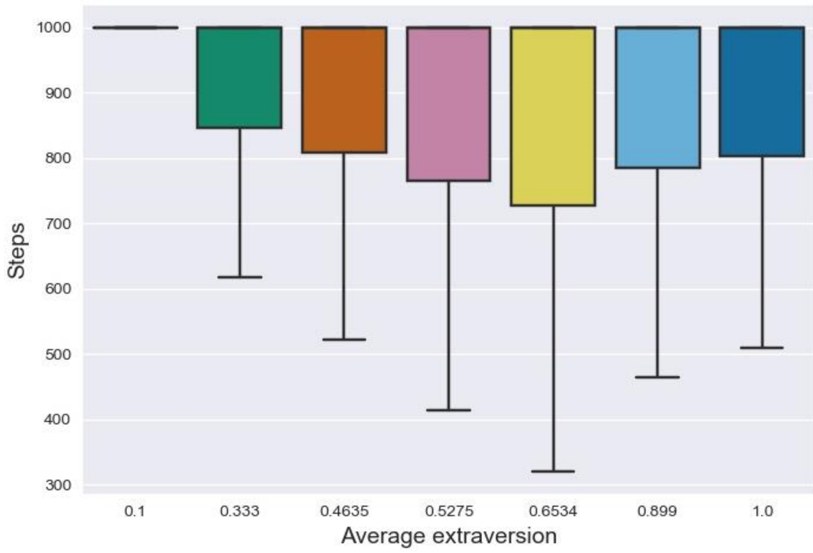


Figure 6.35 The relation between the average team extraversion and the number of steps needed to complete the task

Expectedly, the figure depicting step distribution is in accordance with data presented in Figure 6.36b. Figure 6.36b shows how a team whose average extraversion is slightly above average manages to successfully conclude the highest number of the tasks in a given timeframe. Such curvilinear relationship has also been reported in the literature (e.g. [322], [323]). In particular, Barry and Stewart note [322] that when the team has to come up with a single solution, having many extraverts creates a situation in which there are many 'leaders' but no 'followers', thus introducing the conflicts and prolonging the time for convergence.

On the other hand, Figure 6.36a shows how members of the most extraverted team are the most successful in terms of detecting a solution. This finding is not surprising since highly extraverted agents tend to communicate – but also to create – significantly more structures than the less extraverted teams (Figure 6.37). In other words, extraverted team members may be beneficial to team performance on tasks that require creativity and generation of a large number of ideas [237]. But in a setting where agents need to converge to a single structure such a large extent of ideas seems to 'overburden' the team and prevents any solution from standing out.

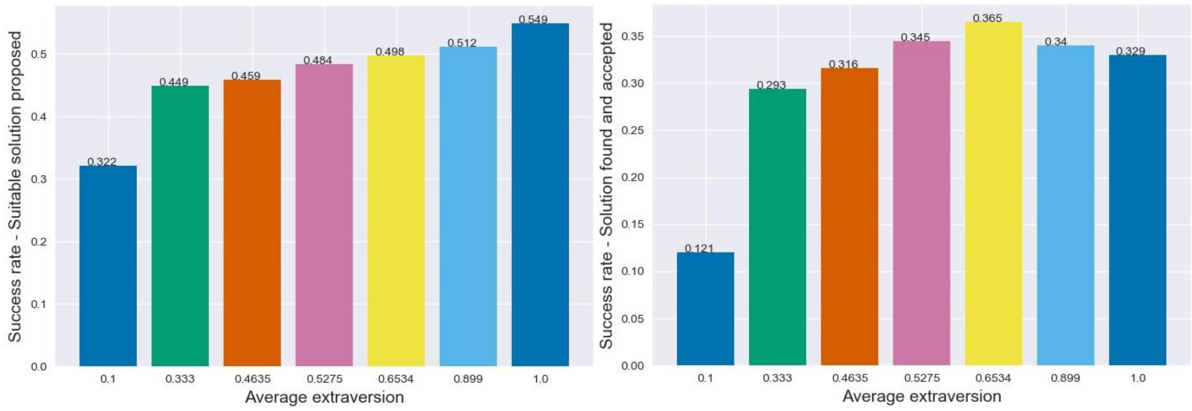


Figure 6.36 The success rates with respect to the average extraversion of team members a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to settle for a solution

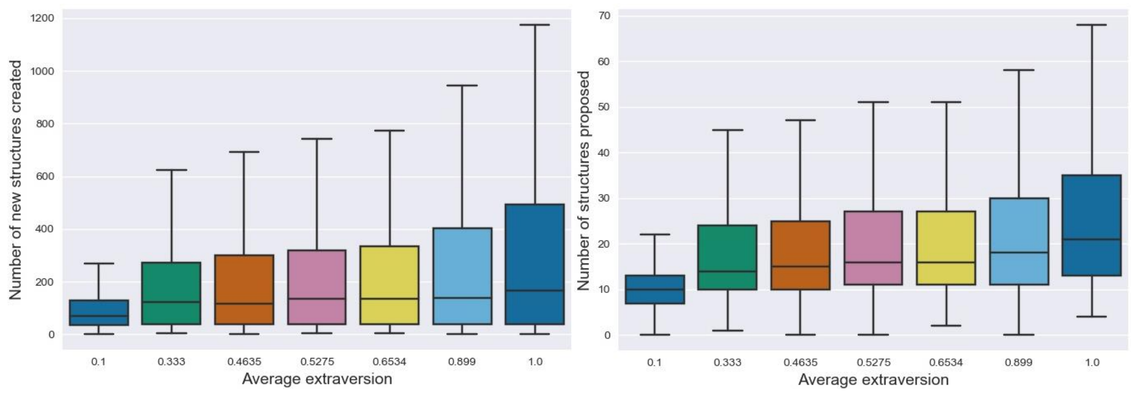


Figure 6.37 The relation between average extraversion of team members and solution space exploration behaviour regarding a) number of newly created structures, and b) number of distinct structures proposed

Additionally, data presented in Figure 6.37 demonstrates how a team of very introverted agents communicates a significantly lower number of structures. In turn, there are fewer opportunities for idea cross-fertilisation [237], and the agents do not produce as many new structures. All of this leads to low success rates for a highly introverted team.

Finally, one can discuss these ideas from the perspective of the desired team diversity (with regards to the extraversion of members). If the data is categorised based on a difference in extraversion parameter of the most introverted and the most extraverted team members, one can conclude that a slight variability is beneficial to the team: if agents are all equally extraverted, their share in communication is similar – either leading to agents trying to express their thoughts (thus creating a 'noise' in other agent's mental processes), or to agents that take time to reassure the quality of their ideas (thus creating a long periods of silence, omitting communication of some important knowledge elements and prolonging the time needed to solve the task). On the other side of the spectrum are teams where (at least) one agent is very introverted, and (at least) one agent is very extraverted. The reason for the weak performance of such team may be the presence of very introverted agents. Barrick et al. [233] found that in situations in which all of the team members have to contribute to the solution, the presence of introverts may harm the team performance. In the study at hand, the introverted agent possesses expertise distinct from that of its co-workers. If the agent does not communicate its thoughts, the remaining team members may be unable to fill in the knowledge gaps needed to solve the task.

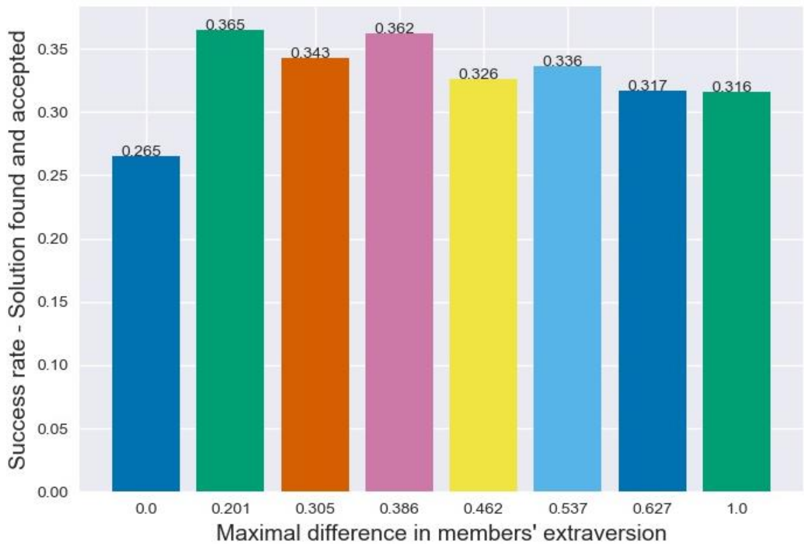


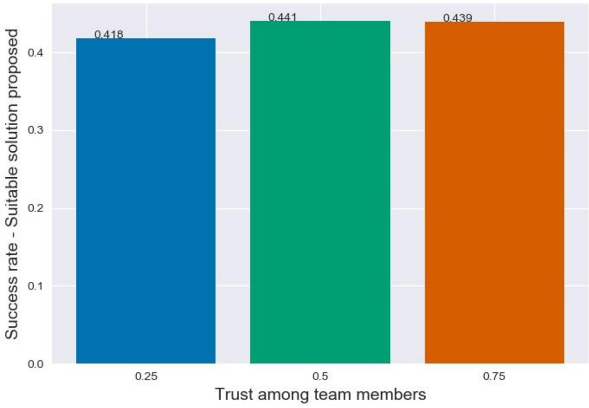
Figure 6.38 The relation between success rate and the difference in members' extraversion scores

It should be noted, however, that due to the modifications of mechanism assigning extraversion parameters, the distribution of extraversion does not follow a normal curve. This means that in the population of teams whose difference in extraversion equals zero, the highly introverted and highly extraverted agents are equally likely as the team of agents all scoring 'moderate' values in extraversion. The success rate presented in Figure 6.36 will likely be higher if the default extraversion assignment mechanism was used. Future studies will test such claims. Additionally, the analysis regarding the minimal and maximal extraversion score, as well as the variance of extraversion among team members will be calculated and tested for relations with team performance [233].

6.2.4 Trust

A final study regarding the elements of the implemented model dealt with the impact of trust on the team performance. To study the effect of trust, in each simulation run, agents with low, moderate and high levels of trust performed a task. In other words, the experimental setup was of the form $T_1T_1T_1$, where agents' mental models were restarted in between two task performances, and the trust values were changed. In the study presented herein, the trust was considered as symmetric, thus leading all agents to have low, moderate or high levels of trust in others. Overall, 5,000 simulations were run.

Figure 6.39 shows that the agents' success rates are comparable, with the non-trusting team scoring slightly lower than the team of moderately-trusting and high-trusting agents. To test whether the lack of trust prolongs the time needed to solve a task, the number of steps required to reach the solution is observed. These data is presented in Figure 6.40a.



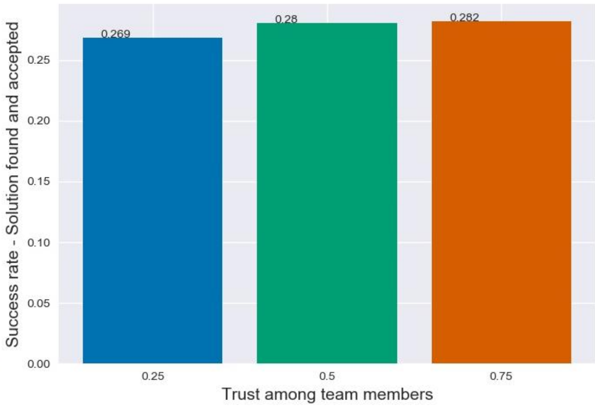


Figure 6.39 The success rates with respect to trust among team members a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to settle for a solution

Wilcoxon signed rank test was employed to study the differences in steps distributions among teams of varying trust levels. The teams low on trust value performed statistically significantly worse ($p < 0.01$) – with the regards to time needed to finish the task – then moderate- and high-trusting teams. This accords with the proposition laid out in Salas et al. [32]. In other words, as modelled, low-trust agents do not rely on knowledge links communicated by others. Rather, they opt to ensure the adequacy of communicated knowledge by themselves. This, as demonstrated, prolongs the time needed for a team to find and converge to a solution. However, no statistically significant differences were found between the two remaining simulated conditions.

Additionally, one can study if the amount of trust has an effect on the team's solution space exploration. Although the attention given to other's solutions is mostly guided by the related – but separate – agreeableness parameter, the higher amounts of trust facilitate learning from others and thus can provide agents with new 'paths' (i.e. knowledge links) leading to solutions.

The found differences among the simulated cases were small (see Figure 6.40b), but significant ($p < 0.05$). Since the agents in high-trust condition do not take significantly more time to finish the task (or, in fact, solve the task quicker than non-trusting agents), the obtained result indicate that the agents managed to more quickly traverse the problem-space related links and activated more structures (and used them to generate new ones).

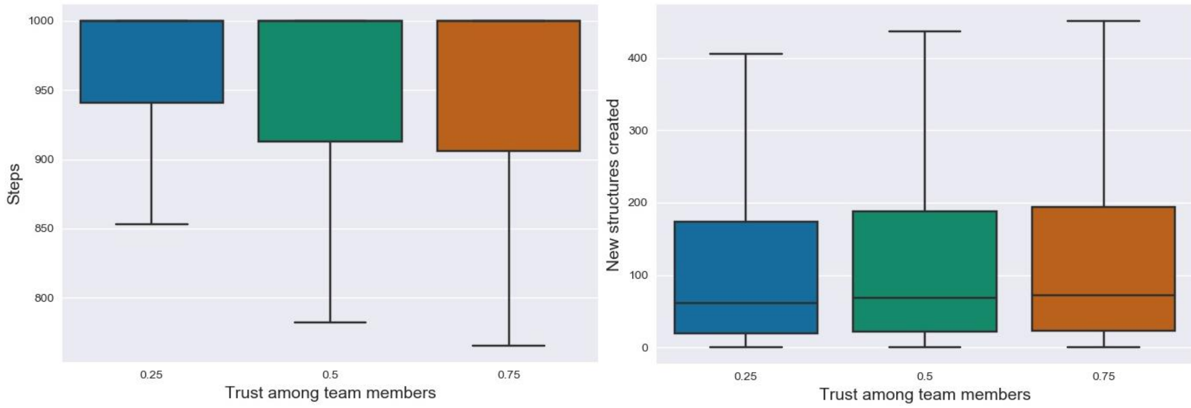


Figure 6.40 The distribution of a) steps needed to finish the task, and b) the number of new structures created with the respect to the level of trust among team members

The conducted analysis revealed the small, yet significant, impact of trust on the team's behaviour and performance. Additional refinements will have to be made to elaborate on the potential influences of trust further. For example, as currently modelled, all of the knowledge links present in agent's mental model are 'correct', and most can be deduced by agents (by, for example, analysing the structure's properties). However, if the agents were primed with the 'wrong' knowledge links, the effect of trust on team performance may be amplified. Similarly, as currently modelled, the minimal amount of trust studied still permits learning from others. Thus, future refinements will include changing the lower (and perhaps upper) bounds of viable trust range. Finally, in the experiment reported here, all of the trust values were shared among agents. In other words, each agent had a similar reputation. Thus, the effect of trust on solution evaluation and acceptance was not studied within the frame of the experiment presented. Future work will include simulations were, for example, one well-respected agent does not trust its team members.

7 STUDY OF DESIGN TEAM LEARNING AND ADAPTATION

The previous chapter studied the interplay of various modelled elements and mechanisms. This chapter aims at demonstrating the model's capability to serve as a tool for studies of teams over time. In particular, as discussed in Chapter 4, the aim is to study the team learning and adaptation processes.

Two experimental setups were created to study how the behaviour of a simulated team changes over time. The first – reported in Section 7.1.1– tested if the team's behaviour changes when the team is faced with the task it was exposed to previously. The second – detailed in Section 7.1.2 - serves to study teams' change in behaviour across different tasks. These experiments, in sum, display the simulated team's capability of learning and changing its behaviour over time. As emphasised in previous chapters, within this work, team learning is modelled as an emerging phenomenon resulting from interactions among agents. To capture team learning and adaptation, the change in team performance regarding success rate and convergence speed were extracted. These metrics accord with literature (e.g. [31], [167]) stating that team learning can be observed through the lenses of team performance and cooperation improvement. Further, the details on communication among team members are collected, and the communicated content is analysed. The data on newly created and communicated knowledge elements (structures and links) depict the changes in the collective knowledge and, thus, serve as an indicator the team learning has occurred [167].

7.1.1 Effect of learning and experience on team performance on a similar task

The study reported in this subsection extends the work published in [261]. In particular, the study presented here reuses the experimental setup introduced in the cited work. Thus, one simulation run consisted of four task sequences of the form: $AB_1B_2 \ldots B_nA$ where n equals 1, 3, 5 and 10. Every task B_m is different from task A (for any m). Any two B_i and B_j can - but do not have to - share some of the requirements.

The team's performance on the first task A was compared to its performance on the second task occurrence. It is expected that the experience gained in between two task-A performances enables the agents to find a solution more often, converge to a solution quicker, and improve their search process on the second task-A performance. Thus, the hypotheses are introduced as follows:

H1: As team members work together over time, they become more efficient.

H2: As team members work together over time, they explore less.

To test the listed hypotheses, 1500 simulation runs were performed. Each run consisted of the four task sequences composed of 3, 5, 7 and 11 tasks – of which first and the last task were similar.

Figure 7.1 shows the results obtained regarding the first hypothesis (H1). In the figure, the x-axis labels are set as follows: *A0* denotes the performance achieved on the first task *A*. *A1*, *A3*, *A5* and *A10* denote the scores obtained on the second task-*A* performance where the number indicates how many tasks (B_m) were performed in between two task-*A* performances.

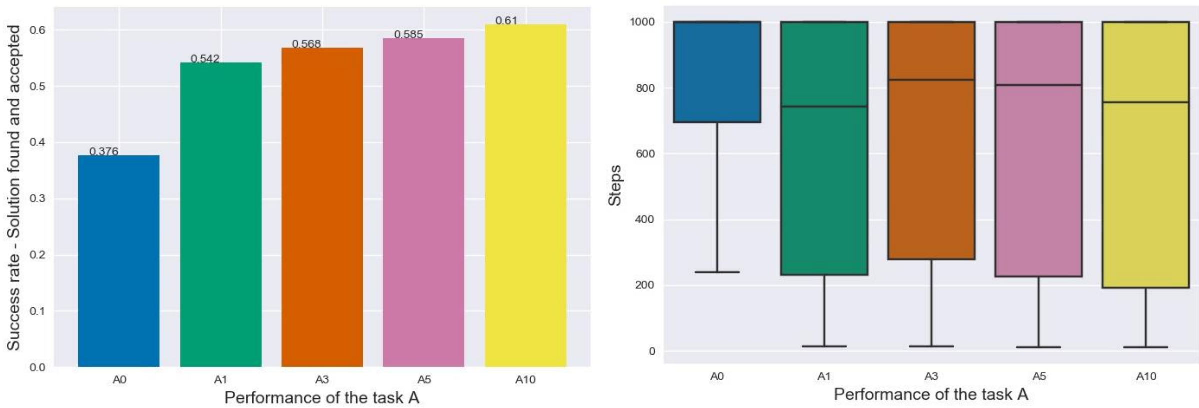


Figure 7.1 Comparison of team's performance on the two instances of the same task regarding a) the success rate and b) the distribution of steps needed to finish the task

Although the Figure 7.1a demonstrates a positive correlation between task success and the number of tasks performed in between two task-*A*s, the Figure 7.1b shows the agents were significantly quicker to find a solution if the first task-*A* was performed very recently (i.e. with one task in between). These results indicate that when two task-*A* performances are separated by just one other task, the team is capable of recalling previously found solution and proposes it promptly. As more time passes between two task-*A* performances, the agents forget their initial solution (i.e. the links leading to it decrease in weight) and the team tries to reconstruct it. The inspection of the data revealed that, when performing task *A1* after the *A0* was successfully finished, the agents picked the same solution as that proposed in the first task in 84.1% of the cases. In other words, the agents differ from the originally proposed (successful) solutions in less than 16% of the cases. As more time passes, the likelihood of the agents remembering the exact solution proposed in the first task-*A* performance drops. Namely, the same solution is recalled after three tasks 71.89% of times in case of *A3*, and 70% of the times in task *A5*. After ten tasks (i.e. in task *A10)*, the agents converge to the same solution as the one found in the first (successful) task in 54.3% of the cases.

The listed statistics indicates that after a certain time (counted by the number of tasks) had passed in between two performances of the same task, the agents forget their initially produced ideas (i.e. recalling them takes more time). Thus, although the more experienced agents are more successful (Figure 7.1a), they may, in some cases, be slower to reach the solution. Namely, if their knowledge is not yet sufficiently developed to support quick convergence to a new solution, the more experienced agents may take more time to converge to a solution than they did in the first trial (case A3). If however, even more time passes in between two task performances, the agents gain experience on which they can rely to generate new solutions (if the past ones cannot be recalled).

Wilcoxon signed-rank test was used to determine the significance in step distribution differences, and the resulting p-values are presented in Table 7.1. As one can observe from the table, the second time agents are performing the task *A*, the convergence is quicker (than in 'no-experience' *A0* case), irrespective of the number of tasks conducted in the meantime. However, a significant drop in performance occurs if three, rather than only one task is performed in between tasks *A*. As more tasks are performed, the performance starts to improve slowly, ultimately improving significantly from *A3* case.

Table 7.1 The statistical significance (p-values) of the steps distributions differences

Task	**A**	$AB_1\mathbf{A}$	$AB_{1,2,3}\mathbf{A}$	$AB_{1,...,5}\mathbf{A}$
$AB_1\mathbf{A}$	0.0000			
$AB_{1,2,3}\mathbf{A}$	0.0000	0.0182		
$AB_{1,...,5}\mathbf{A}$	0.0000	0.4429	0.1625	
$AB_{1,....,10}\mathbf{A}$	0.0000	0.1882	0.0038	0.1174

To study the agents' behaviour regarding the exploration of the solution space (H2), the statistics on the number of new structures generated, as well as the overall number of distinct structures proposed during the performances of task *A* are conducted (see Figure 7.2). Additionally, the details on the number of new links learnt and number of distinct links communicated are obtained and presented in Table 7.2.

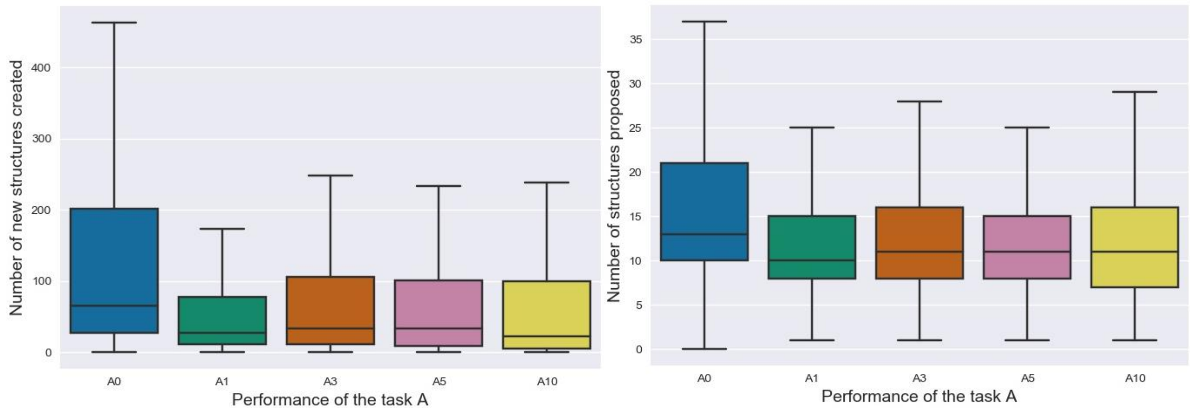


Figure 7.2 The change in a) the number of new structures generated, and b) the number of distinct structures proposed as a result of experience gain

Table 7.2 The change in task performance as a result of experience gain

Task		**A**	AB$_1$**A**	AB$_{1,2,3}$**A**	AB$_{1,..,5}$**A**	AB$_{1,...,10}$**A**
Steps	min	8.0	14.0	13.0	11.0	10.0
	1st Q	671.0	231.2	279.2	226.2	192.0
	median	1000.0	743.5	824.0	809.5	755.0
	mean	811.2	630.6	671.4	647.9	618.6
	3rd Q	1000.0	1000.0	1000.0	1000.0	1000.0
	max	1000.0	1000.0	1000.0	1000.0	1000.0
New links created	min	3.0	1.00	6.00	1.0	0.0
	1st Q	252.0	85.25	86.25	85.0	54.0
	median	482.5	194.00	229.00	232.5	175.0
	mean	887.0	402.23	518.42	535.6	502.3
	3rd Q	1139.5	450.50	641.75	641.5	617.5
	max	7256.0	5289.00	4336.00	10501.0	4362.0
New structures created	min	0.0	0.00	0.00	0.00	0.00
	1st Q	31.0	11.00	11.00	9.00	5.00
	median	69.0	27.50	34.00	33.00	23.00
	mean	146.4	68.78	86.41	89.99	82.43
	3rd Q	190.8	78.00	106.00	101.00	99.50
	max	1271.0	975.00	804.00	2082.00	816.00
Links communicated	min	0.00	0.00	1.00	0.0	0.00
	1st Q	22.00	18.00	18.00	18.0	13.00
	median	28.00	24.00	26.00	25.0	25.00
	mean	28.46	24.43	25.08	24.8	23.36
	3rd Q	36.00	31.00	32.00	32.0	33.00
	max	68.00	57.00	68.00	67.0	70.00
Structures proposed	min	0.0	1.00	1.00	1.0	1.00
	1st Q	10.0	8.00	8.00	8.0	7.00
	median	14.0	10.00	11.00	11.0	11.00
	mean	17.1	13.77	14.17	13.7	12.83
	3rd Q	22.0	15.00	16.00	15.0	16.00
	max	72.0	73.00	64.00	62.0	54.00

In general, the presented data aligns well with the previously presented discussion on the difference in the number of steps: if the task is repeated shortly after the first completion, it is performed quickly and with less solution space exploration. If the sufficient time passes for the links relevant for the task to decrease in weight, the agents' performance depends on the level of experience obtained: the more experienced the agent, the quicker the convergence (and the fewer solutions are generated and proposed).

Finally, the change in the content of the messages exchanged during the first and second performances of task *A* can be explored. Figure 7.3 presents the relative frequency of message types over the course of task *A* performances.

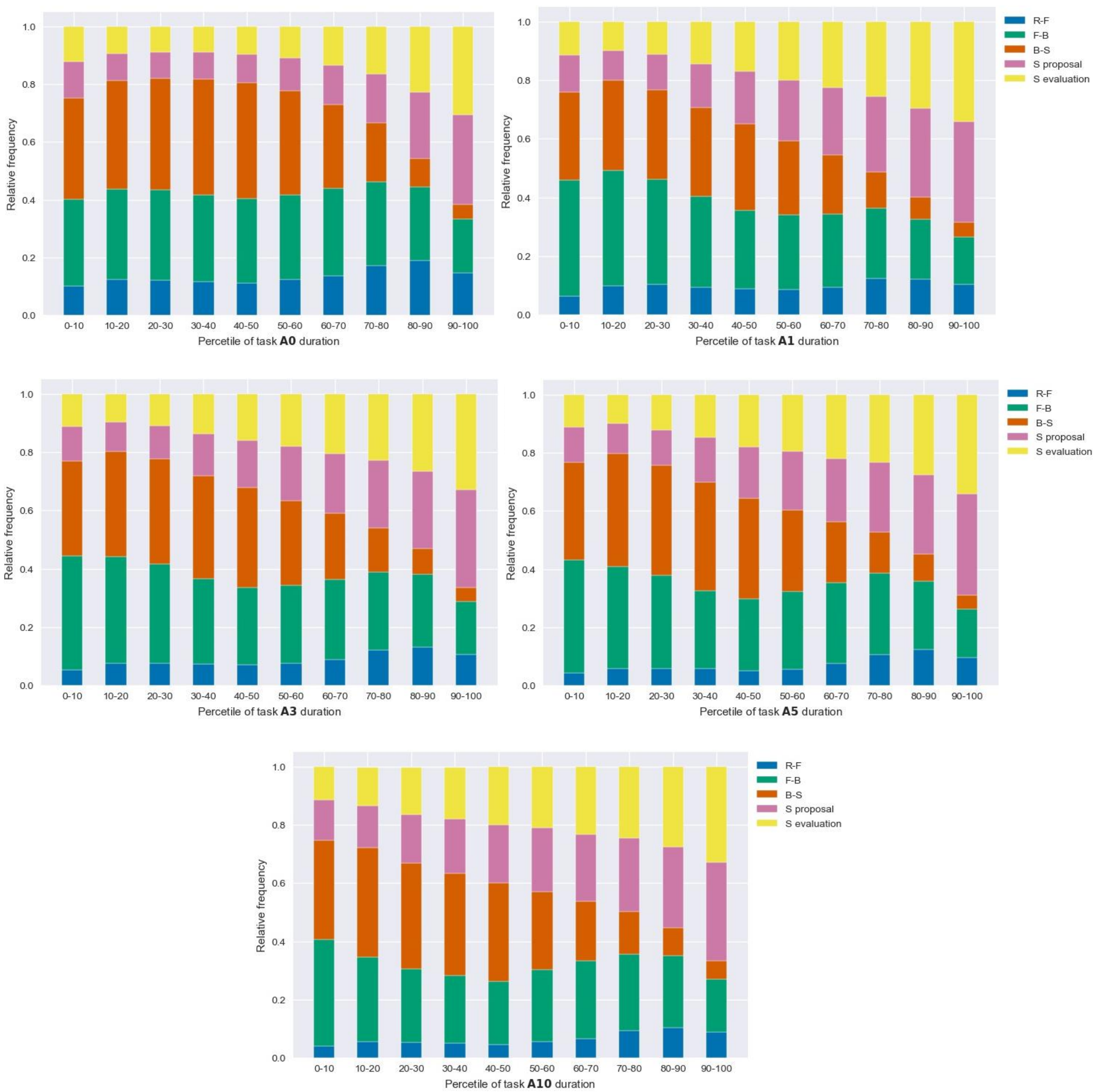


Figure 7.3 The content of the communicated messages in the repeated performance of a task

From Figure 7.3, one can observe how inexperienced agents commence the task by exchanging an approximately equal number of function-behaviour and behaviour-structure links. The second half of the task shows an increase in solution proposal and evaluation messages, but also a slight increase in communication of requirement-function links. This behaviour stems from agents' discussion of the problem space due to Reformulation III processes (occurring due to detection of unsatisfied requirements). The second task performance – irrespective of the number of tasks performed in between – commences with a greater focus on the problem formulation, and less to synthesis (than the behaviour displayed by novices). This difference stems from the fact that – in comparison to more experienced agents - the 'novices' know fewer function nodes. Thus, when faced with a task, a few function nodes get activated, the relevant links discussed, and the agents start communicating (more numerous) behaviour-structure links. As the agents gain experience, more relevant function nodes are learnt and can be activated, thus influencing agents' communication. Finally, after multiple tasks are performed, the agent's function-behaviour and behaviour-structure links get sufficiently grounded, so the agents performing task *A10* can quickly traverse such nodes and propose structures – thus displaying an increase in solution proposal and evaluation during the first periods of the task.

Similarly, the differences in communication trends of experienced and inexperienced teams can be observed with regards to the later phases of the task. The most evident such difference is a gradual increase in a share of solution proposal and evaluation messages, which stems from the experience gained by the team. These trends accord with the literature [309], [324] discussing the effect of experience on the solution space exploration. The study reported in the next section takes a closer look into this behaviour.

7.1.2 Effect of learning and experience on team performance on different tasks

To further study the effect of learning on the team behaviour, the analysis similar to the one reported in the previous subsection was conducted, this time taking a closer look into agents' behaviour over various tasks.

The study reported in this subsection extends the work presented in [325]. Similar to the experiments reported in the cited research, the simulated task sequences consisted of several different tasks. To enable comparisons, each of the tasks was performed by both – experienced agents ('experts') and inexperienced agents ('novices'). The simulation runs were of the form $T_1\, T_2 \ldots\, T_7\, T_1 T_2 \ldots T_7$, and – similar to the study presented in Section 6.1.1.2 – the agents learned over the first set of tasks, while during the second set of tasks the agents' mental models were restarted in between every task pair. Overall, 1500 simulation runs were performed.

Following the theoretical background laid out in [325], the hypotheses tested in this subsection are formulated as follows:

H3: Experienced (expert) teams perform better than inexperienced (novice) teams in terms of time needed to find a solution.

H4: Experienced teams spend more time than inexperienced teams exploring the solution space than the problem space.

As seen in Figure 7.4, the experience gained throughout past tasks enables 'expert' agents to improve their performance continuously, consequently increasing the difference in obtained success rates and the success rates achieved by 'novice' agents. In line with these results, Figure 7.5 shows the effect of experience on the number of steps needed to find a solution. The distribution of steps presented in the figure provides support for the hypothesis H3 – but differences in steps required to finish the task can be a result of the increasing success rate. To determine if the 'expert' and 'novice' agents differ in cases where the solution was found by both teams, the details regarding the novices' performances and experts' performance in each of the tasks (marked with *T1*-*T7*) are presented in Table 7.3. For simplicity, the novices' performances on every task are aggregated under the name *T0*.

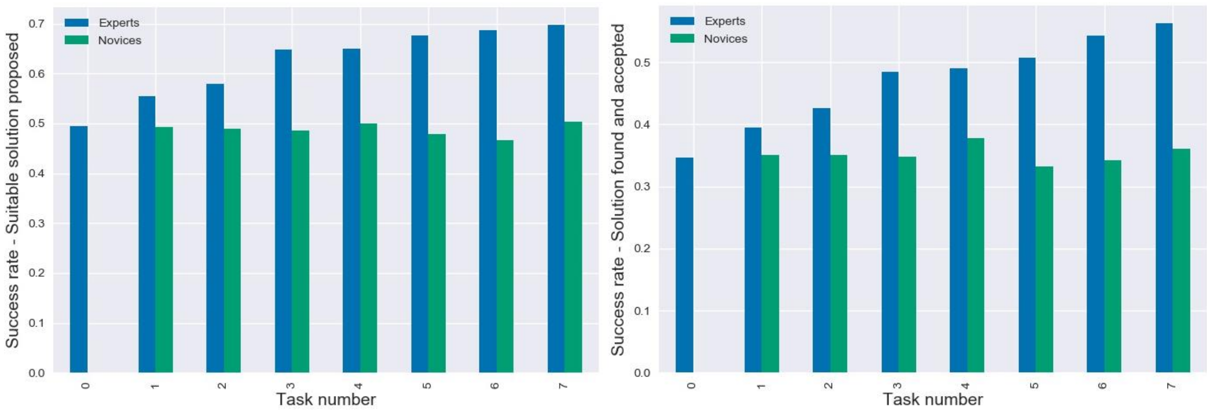


Figure 7.4 The success rates with respect to the team members expertise gain a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to settle for a solution

In addition to data presented in Table 7.3, Figure 7.6 and Figure 7.7 depict the differences in agents' solution space exploration in both – successful and unsuccessful – tasks. The agents' performance on the successful tasks aligns well with the second hypothesis proposed in the previous subsection. Namely, as the agents gain more experience, they tend to create fewer new structure nodes (Figure 7.6a) and propose fewer structures (Figure 7.7a). Unsuccessful tasks, however, elicit different behaviour. In an attempt to solve a task, the more experience agents extensively search the solution space – therefore generating a large number of structures. The number of new structures created (in unsuccessful tasks) increases with the increase in agents' experience. Interestingly, although the more experienced agents generate a significantly higher number of structures, they propose fewer distinct structures than inexperienced agents. This behaviour is a result of the behaviour-structure links grounding: while novices rate newly generated structures as similar (in score) to known structures - and, thus, worth sharing; due to well-grounded behaviour-structure links, experienced agents favour known (albeit unsuitable) structures to new ones – until sufficient analysis is performed to determine the suitability of the newly generated structures.

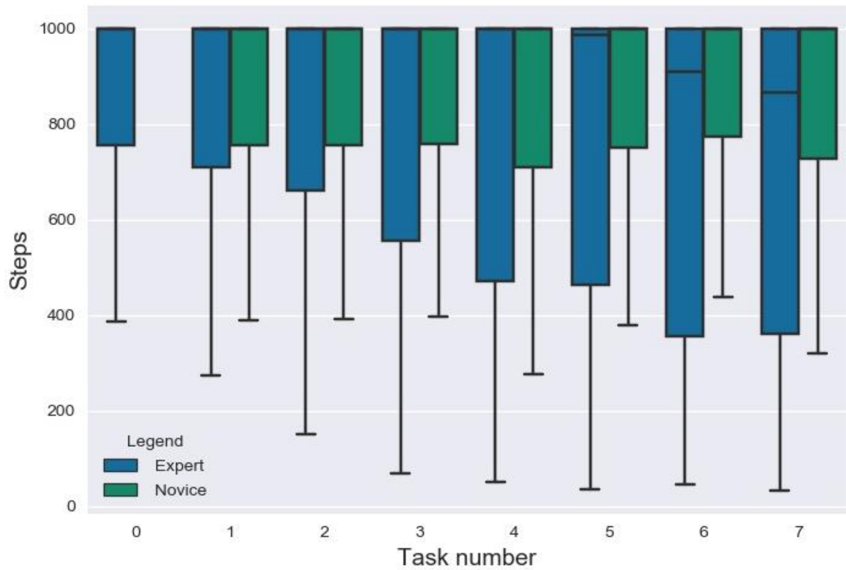


Figure 7.5 The distribution of steps needed to finish each task in a sequence of different tasks

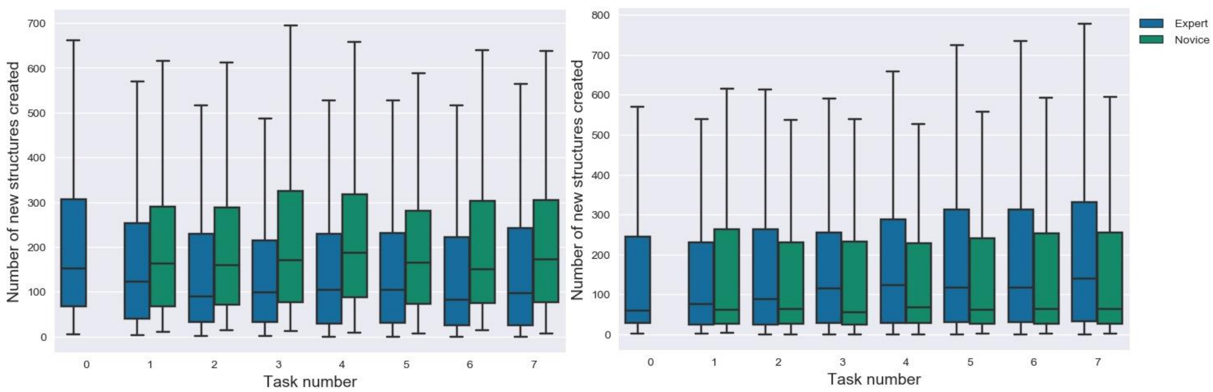


Figure 7.6 The distribution of the number of new structures created in a) successful and b) unsuccessful tasks

Table 7.3 Details of the team performance in successful tasks

Task		T0	T1	T2	T3	T4	T5	T6	T7
Steps	min	87.0	86.0	68.0	70.0	51.0	35.0	45.0	32
	1st Q	379.0	323.0	308.0	294.5	251.8	232.0	207.0	219
	median	612.0	594.0	579.5	535.0	462.0	471.5	392.0	419
	mean	585.5	562.8	559.1	535.7	500.4	499.0	460.8	464
	3rd Q	767.0	771.0	798.8	773.0	743.0	778.8	717.0	714
	max	999.0	999.0	999.0	999.0	998.0	999.0	999.0	999
p - value			0.0607	0.0089	0.0005	0.0000	0.0000	0.0000	0.0000
New links created	min	74	49	35.0	26.0	27.0	23.0	23.0	16
	1st Q	559	319	280.0	270.5	249.8	269.0	216.0	221
	median	1067	783	614.0	648.0	685.0	698.0	563.0	670
	mean	1311	1067	973.1	962.5	1000.6	998.5	942.2	1003
	3rd Q	1790	1509	1394.0	1358.5	1413.2	1391.5	1352.0	1494
	max	6165	7147	5603.0	6552.0	6385.0	7920.0	10488.0	6154
p - value			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
New structures created	min	6.0	3.0	2.0	1.0	0.0	0.0	0.0	0.00
	1st Q	74.0	40.0	33.0	33.5	30.0	31.0	25.0	26.0
	median	169.0	123.0	89.0	99.0	104.5	105.0	82.0	97.0
	mean	216.7	175.7	157.2	155.9	162.2	162.0	151.9	161.3
	3rd Q	302.0	254.0	229.5	215.5	230.0	230.5	223.0	242.0
	max	1282.0	1506.0	1027.0	1319.0	1168.0	1467.0	2031.0	1190.0
p - value			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Links communicated	min	0.00	2.00	2.00	0.0	0.00	0.00	0.00	1.00
	1st Q	26.00	24.00	24.00	23.00	23.00	21.00	19.00	20.00
	median	31.00	29.00	28.00	28.00	28.00	27.00	26.00	27.00
	mean	31.45	29.65	28.63	28.09	28.21	27.07	25.81	26.34
	3rd Q	37.00	35.00	34.00	33.50	34.00	33.00	32.00	33.00
	max	69.00	58.00	64.00	59.00	58.00	54.00	56.00	71.00
p - value			0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Structures proposed	min	4.0	4.0	2.00	2.00	1.00	1.00	1.0	1.00
	1st Q	13.0	11.0	10.00	10.00	10.00	9.00	9.0	9.00
	median	17.0	15.0	14.00	14.00	14.00	14.00	13.0	13.00
	mean	19.7	17.4	16.43	16.34	16.12	16.18	15.1	15.64
	3rd Q	25.0	22.0	22.00	20.00	20.00	21.00	20.0	20.00
	max	75.0	62.0	52.00	59.00	59.00	64.00	57.0	55.00
p - value			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

To test the second hypothesis proposed in this subsection (H4), the PS index and PS indicator [320] values were calculated for each of the tasks (Table 7.4). PS index and PS indicator measure the ratio of the time spent in a problem space and time spend in a solution space.

The PS index value above 1.0 and PS indicator value below zero indicate the more solution-focused task performance. As seen in Table 7.4, in all of the tasks, simulated agents tend to be more solution-oriented, which accords with the studies on behaviour of designers (e.g. [326]).

However, the claim that experienced agents tend to display a more significant focus on solutions than their inexperienced counterparts is not displayed until sufficient experience is obtained. Namely, it is not until the 7th task that the differences in the ratio of the problem and solution space exploration between experience and inexperience agents become statistically significant.

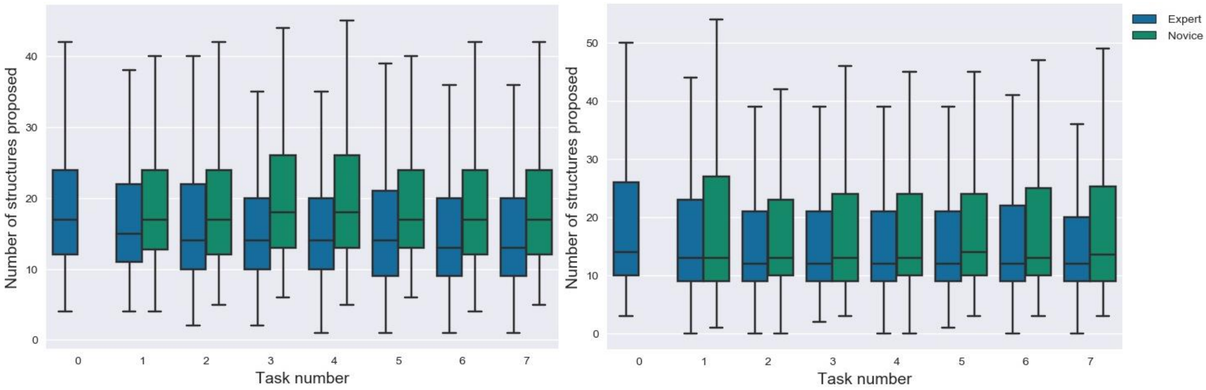


Figure 7.7 The distribution of the number of distinct structures proposed in a) successful and b) unsuccessful tasks

Table 7.4 Statistics regarding PS index and PS indicator values for each of the tasks

Task		T0	T1	T2	T3	T4	T5	T6	T7
PS index	min	0.0000	0.0010	0.0023	0.0000	0.0000	0.0000	0.0000	0.0000
	1st Q	0.3523	0.3232	0.3214	0.3287	0.3458	0.3093	0.2671	0.2664
	median	0.6876	0.6307	0.6442	0.6564	0.6537	0.6347	0.6000	0.5604
	mean	0.8077	0.9083	0.9708	0.9857	0.9079	0.8650	0.8363	0.7699
	3rd Q	1.0704	1.0811	1.1502	1.1387	1.0746	1.0508	1.0704	0.9571
	max	10.108	7.4530	7.1868	8.8617	7.3373	10.263	8.1613	9.1429
PS indicator	min	-1.000	-0.998	-0.995	-1.000	-1.000	-1.000	-1.000	-1.000
	1st Q	-0.479	-0.511	-0.514	-0.505	-0.486	-0.526	-0.578	-0.579
	median	-0.185	-0.226	-0.216	-0.207	-0.209	-0.223	-0.250	-0.282
	mean	-0.220	-0.217	-0.197	-0.198	-0.211	-0.238	-0.261	-0.289
	3rd Q	0.0340	0.0389	0.070	0.065	0.0360	0.0248	0.0340	-0.022
	max	0.8200	0.7634	0.756	0.797	0.7601	0.8224	0.7817	0.8028
p - value			0.5091	0.9306	0.4361	0.09727	0.07049	0.1724	0.0007

Finally, the analysis of the task dynamics regarding the content of the messages exchanged can be studied. Figure 7.8 and Figure 7.9 depict how the experience changes the content of communication among agents. When all tasks are taken into account (Figure 7.8), trends similar to those detected in the previous subsection can be seen: the increase in experience seems to decrease the amount of behaviour-structure links, and - in turn - increase the number of solution proposal and evaluation messages. Additionally, one can observe a significant increase in function-behaviour links in the second half of the task. This trend is a consequence of the frustration parameter and the Reformulation III processes occurring due to detected unsatisfied requirements.

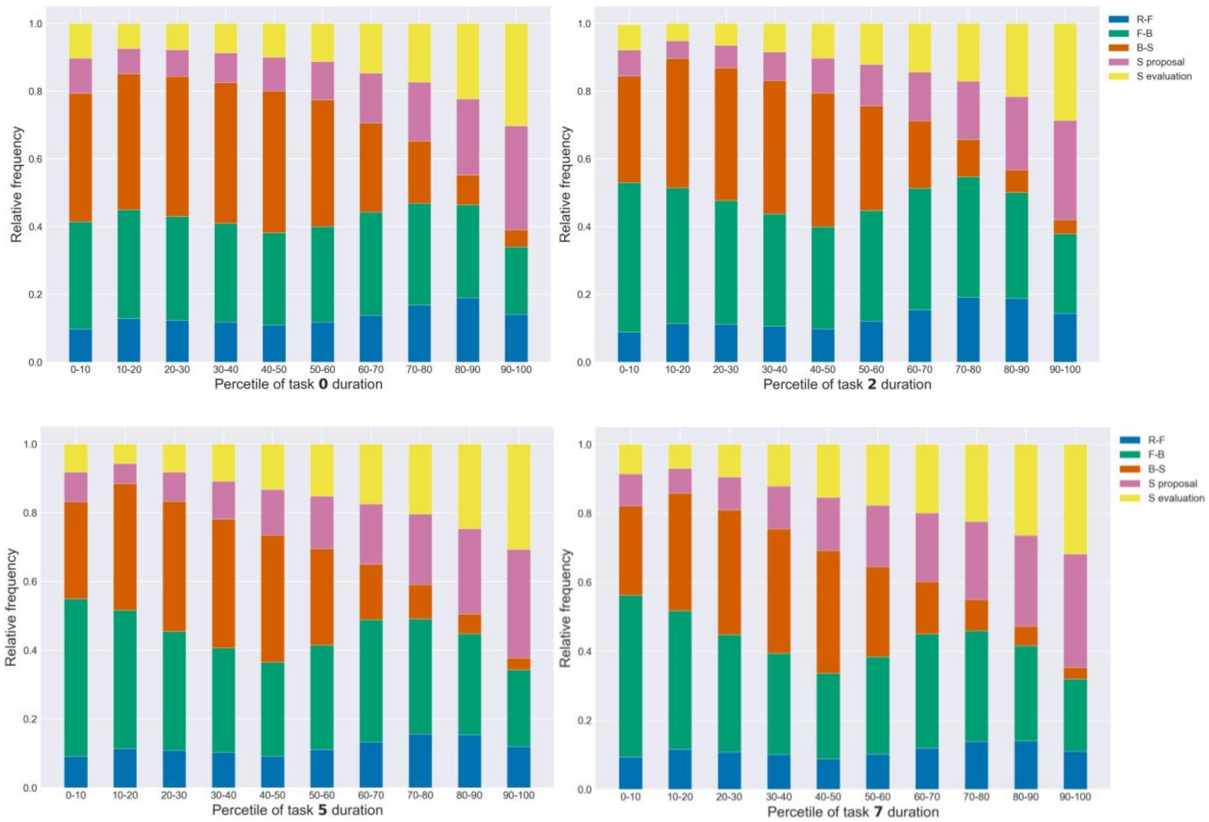


Figure 7.8 The content of the communicated messages of the teams of various levels of experience (performance on both – successful and unsuccessful - tasks taken into account)

When only the successful tasks are considered (Figure 7.9), one can observe the significant increase in the messages related to solution proposal and evaluation. Further, the share of behaviour-structure links reduces with the experience obtained. The experienced agents tend to first discuss the problem-related issues (more so than the novice agents), and then gradually shift to the solution proposal and evaluation. In the successful tasks, the frustration remains low, and thus there is no increase in problem-related issues in the second phase of the task.

Some of the obtained trends align well with the existing theories: for example, experienced agents outperform novices [327], [328], experts are considering fewer options in favour of high-quality solutions [329], and experts' space exploration is more biased towards solutions than that of novices [330], [331]. However, some of the trends are not in accordance with theoretical findings. For example, Liikkanen and Perttula [332] found that novice agents tend to start by exploring the problem space, while experts were able to propose solutions quickly. On the other hand, however, it should be emphasised that: limitations in agent's communication mechanism (see Section 5.2), the prespecified preference for solution (rather than knowledge link) sharing, and the significant grounding in function-behaviour links due to Reformulation III processes all impact the distributions of the message types. Future experiments should be directed towards mitigating these limitations and calibrating the agents' communication to align the distribution of message types with real-world data. Nevertheless, the experiments conducted provided support for the hypotheses listed in this subsection.

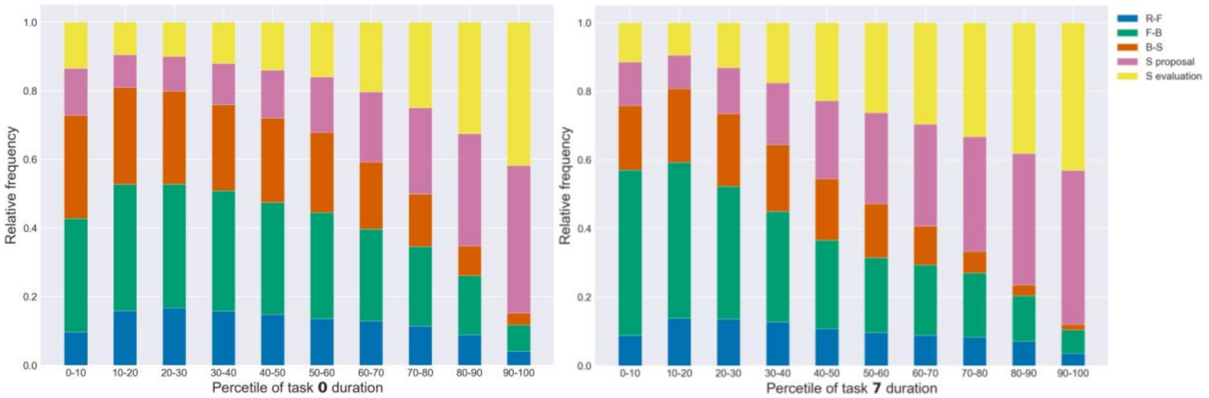


Figure 7.9 The content of the communicated messages of the teams with no experience (i.e. novices) and the teams with the 7-tasks experience (only successful tasks regarded)

The experiments reported in Section 7.1.1 and Section 7.1.2 demonstrated that model's output manages to capture several trends observed in the real world. However, one should ensure these findings are not dependent on the simulated team's size. Thus, the reported experiment was repeated on the teams comprising five and ten agents. Each of the experiments consisted of 500 simulation runs.

7.1.2.1 Simulations comprising five agents

Similar to the findings obtained in case of teams consisting of three agents, the five-member teams use past experiences to improve their success rates (Figure 7.10) and converge to a solution in fewer steps (Figure 7.11).

Further, the number of new structures created (Figure 7.12a) and the number of distinct structures proposed (Figure 7.12b) decline as the agents gain more experience – thus confirming the hypotheses that experienced agents tend to explore less and converge quicker.

Finally, the analysis of the communication content is performed. From Figure 7.13, several similarities with the case of a smaller team can be deduced: the share of behaviour-structure links decreases, while the number of messages related to solution proposal and evaluation increases with the experience gained. Further, the period during which the number of function-behaviour links communicated increases is noticeable in the second simulation half.

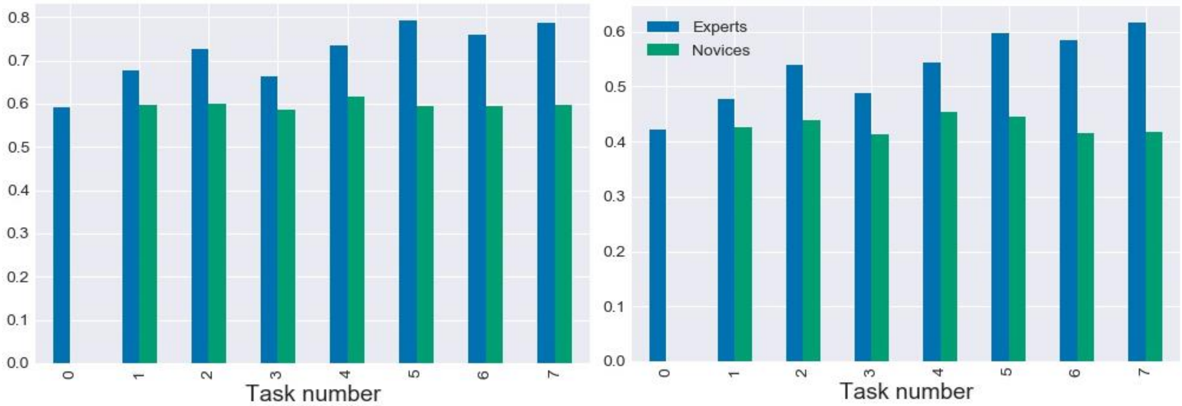


Figure 7.10 The success rates of five-member teams with respect to the expertise gain a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to settle for a solution

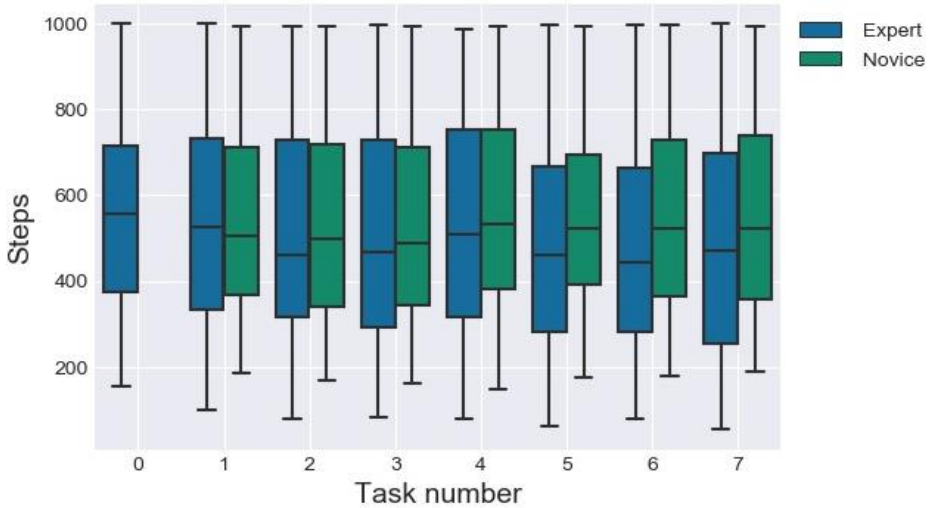


Figure 7.11 The distribution of steps needed for a five-member team to finish each task in a sequence of different tasks

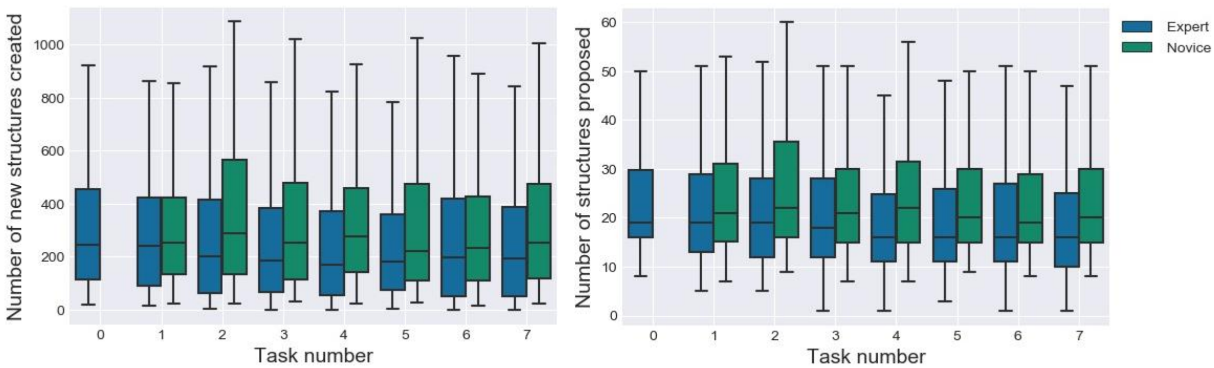


Figure 7.12 Distributions of the number of a) new structures created and b) distinct structures proposed by a five-member team

In comparison to smaller teams, teams of five agents spend more time searching the solution space. This difference is even more evident when regarding only the performance on the successful tasks Figure 7.14. The results obtained for the five-member team align well with those of a smaller team.

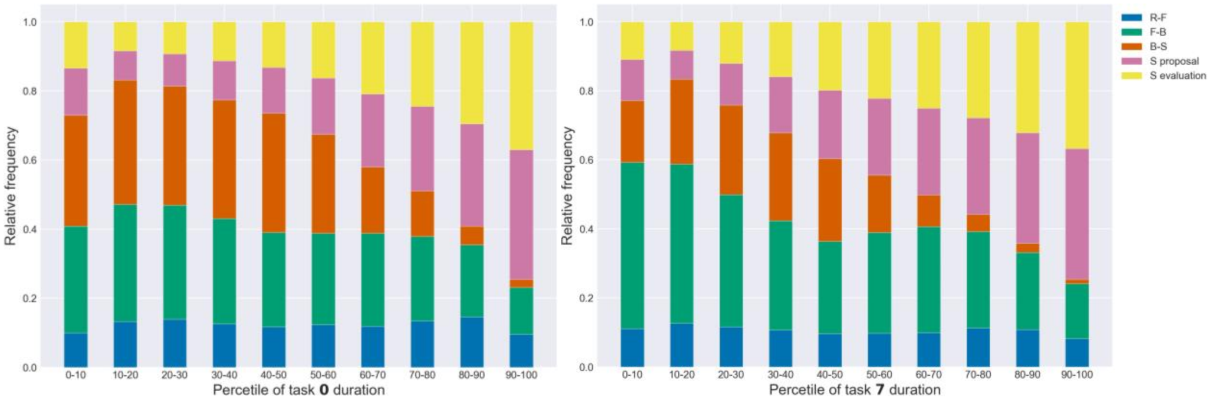


Figure 7.13 The content of the communicated messages of the five-membered teams of various levels of experience (performance on both – successful and unsuccessful **-** tasks taken into account)

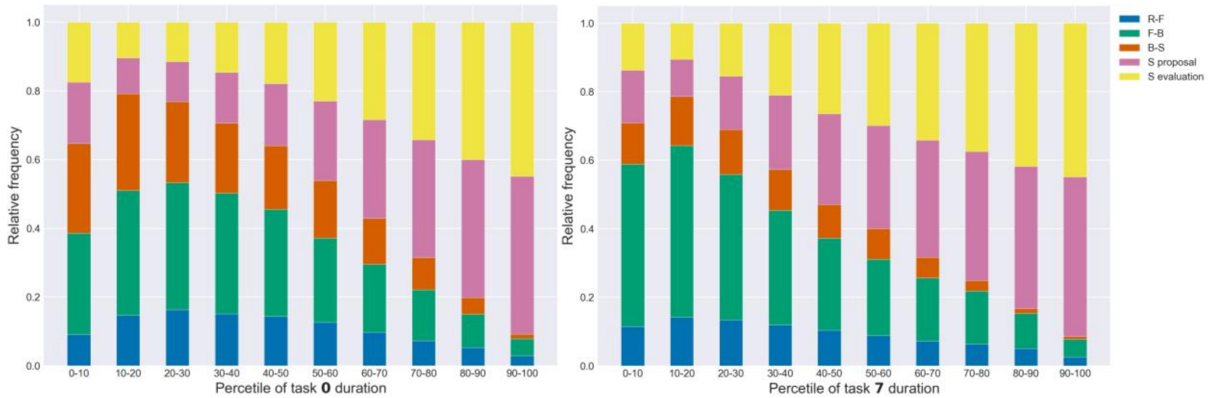


Figure 7.14 The content of the communicated messages of the five-membered teams of various levels of experience (only successful tasks taken into account)

7.1.2.2 Simulations comprising ten agents

Similar to the findings obtained for three- and five-member teams, as a consequence of learning over the previous tasks, the ten-member teams improve their performance over the course of simulation (Figure 7.15 and Figure 7.16). Accordingly, in subsequent tasks, ten-member teams propose and create fewer structures (Figure 7.17).

One additional remark can be made: the inspection of the data regarding the solution acceptance revealed that – while three-member agent accepted approximately 0.19% of the unsuitable solutions, only a few such instances were observed in five-member setting (three in 7500 simulated tasks), while none was observed in the ten-member teams. This finding is not surprising: larger teams consist of team members with diverse expertise and, thus, more diverse priorities. This, in turn, means that every requirement is considered as important by at least one member. On the other hand, the plurality of preferences is likely to hinder quick convergence to a solution. Future work will explore the relation between the number of members in a team and the steps needed to reach the consensus.

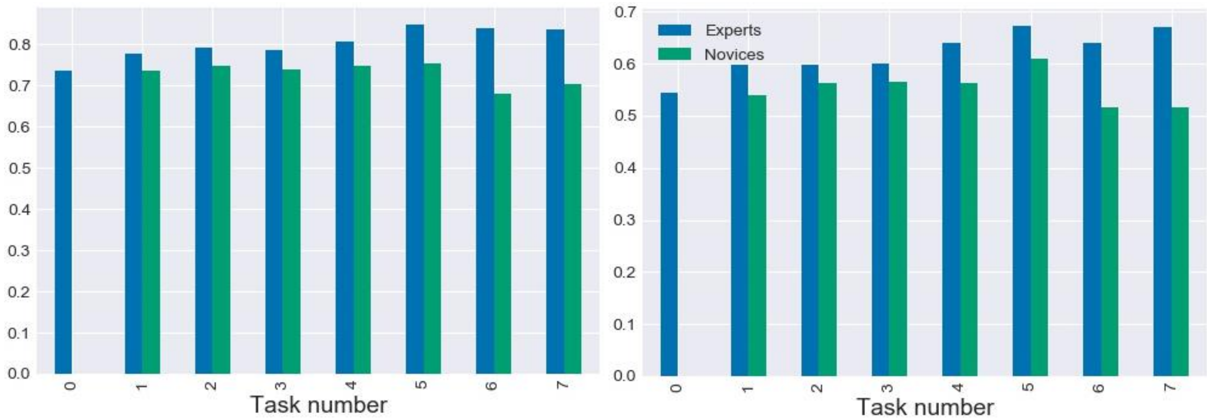


Figure 7.15 The success rates of ten-membered teams with respect to the expertise gain a) Percentage of tasks in which one of the team members proposed a suitable solution, and b) Percentage of tasks in which the team managed to settle for a solution

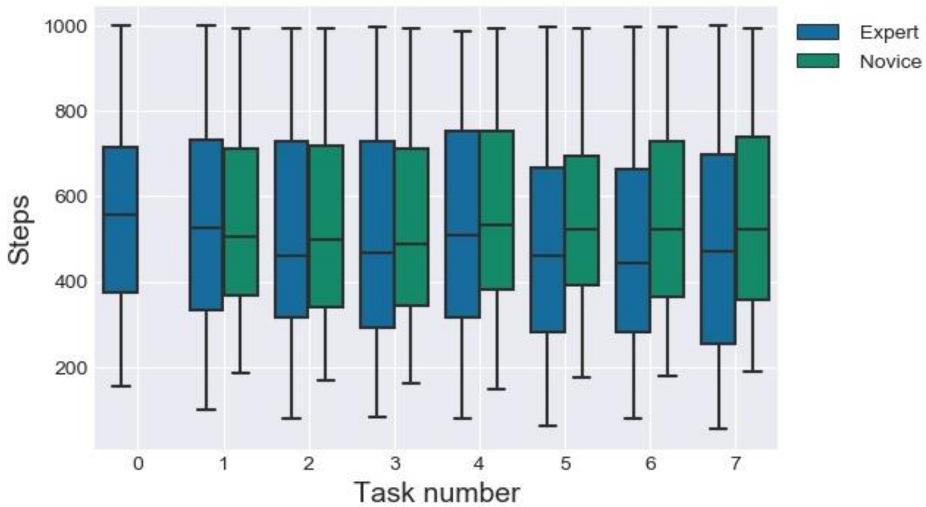


Figure 7.16 The distribution of steps needed for a ten-member team to finish each task in a sequence of different tasks

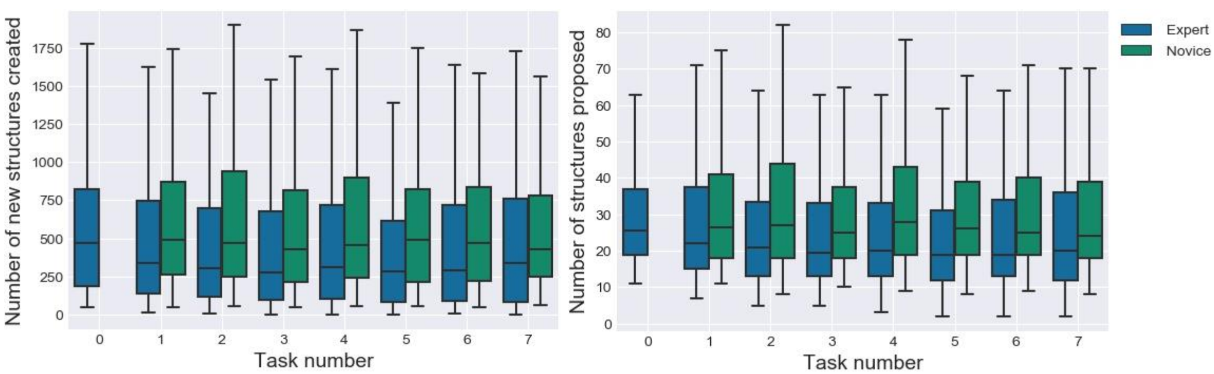


Figure 7.17 Distributions of the number of a) new structures created and b) distinct structures proposed by a ten-member team

The final analysis – as in the case of three- and five-member teams – concerns the content of the communicated messages. Figure 7.18 and Figure 7.19 demonstrate how in ten-member teams, there is a large focus on solution proposal and evaluation. One explanation is that, in larger teams, a particular agent does not get to explicate its thoughts as often. In between two of its ‘speaking turns,’ the agent manages to find a solution it deemed as worth sharing.

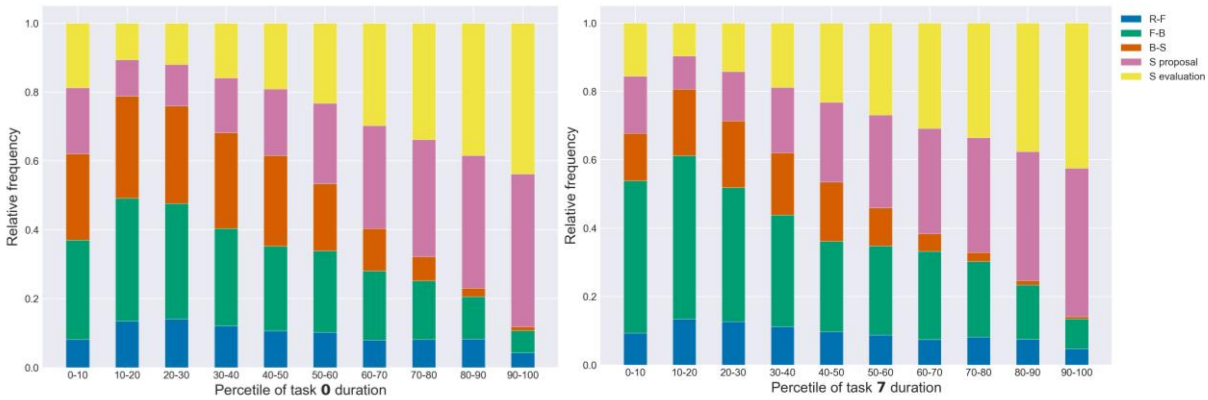


Figure 7.18 The content of the communicated messages of the ten-membered teams of various levels of experience (performance on both – successful and unsuccessful - tasks taken into account)

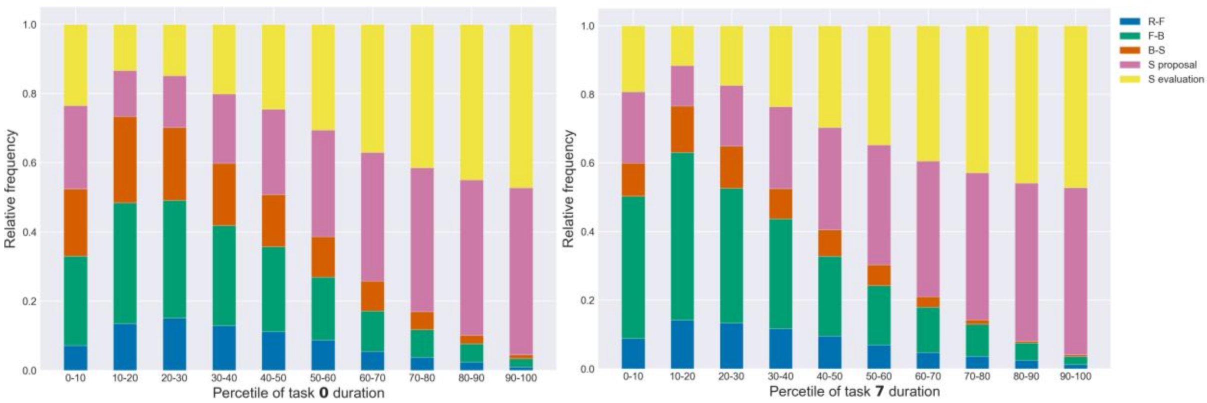


Figure 7.19 The content of the communicated messages of the ten-membered teams of various levels of experience (only successful tasks taken into account)

For completeness purposes, Figure 7.20 represents the data from Figure 7.8, Figure 7.13, and Figure 7.18 side by side to facilitate comparisons. One may note how adding members to a team increases the number of solution proposals and evaluations: both, novice and expert ten-member teams propose significantly more solutions than smaller teams of equivalent experience (in terms of tasks performed). The first decile of the task performed by (either novice or expert) teams of ten members contains significantly more solution proposals than the same period in smaller teams. This indicates that every member of the team had a solution idea to share with others. But in the next periods (2nd and 3rd decile), larger teams focus more strongly on the communication of function-behaviour links. Such an exchange is necessary for every team member to understand the task requirements. Namely, due to the diversity of their expertise areas, each of the team members has a particular "view" of the importance of functions and behaviours and wishes to share them with others.

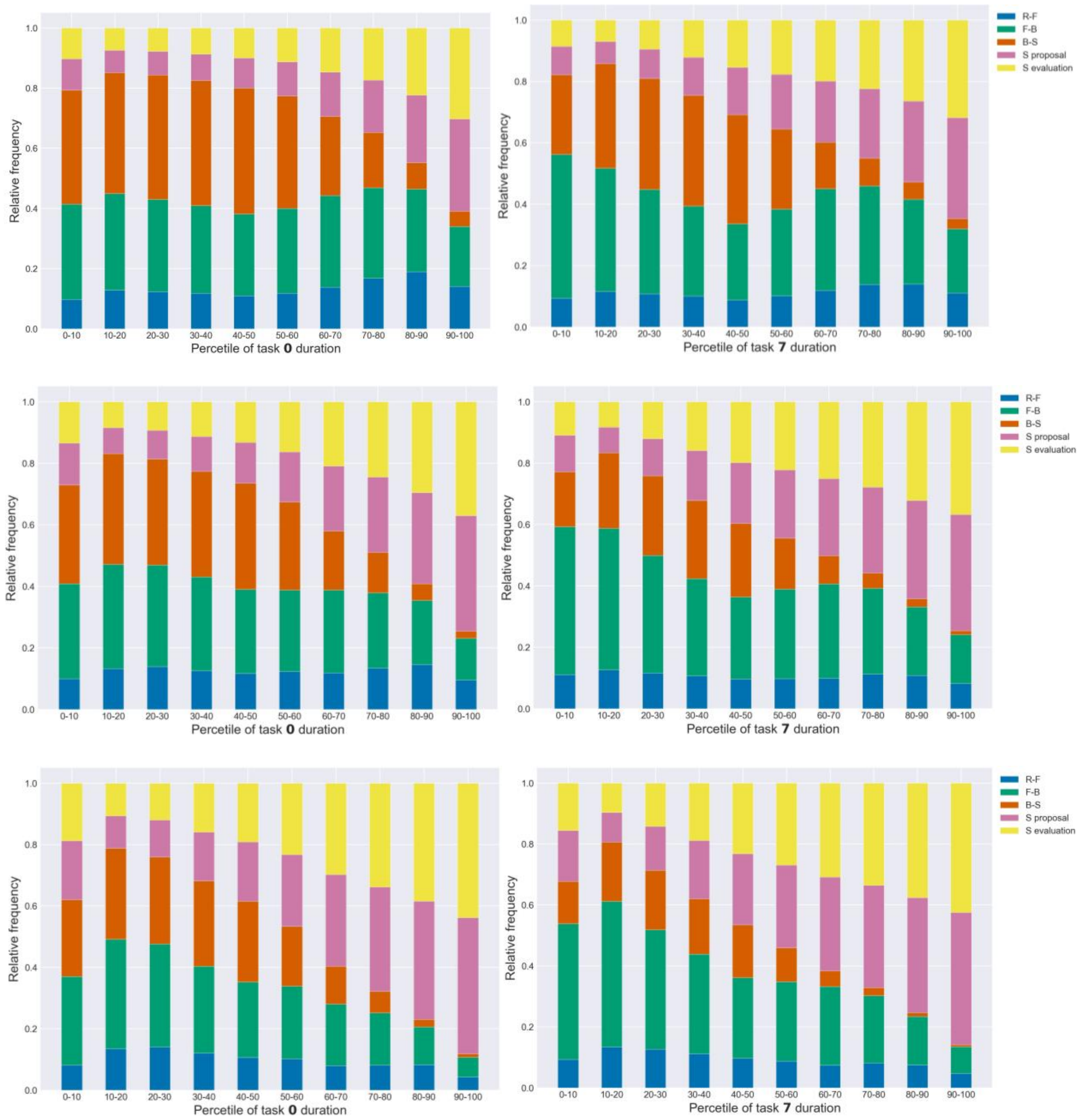


Figure 7.20 Comparison of message type distributions of novices and experts (after 7 tasks) for three-member, five-member and ten-member teams

As discussed earlier, in the later phases of a task, larger teams demonstrate more solution-oriented communication. One possible cause of such behaviour has already been stated. But it may also be that diversity of knowledge enables larger teams to propose a suitable solution quicker. In particular, one limitation of the developed model is the relatively small number of behaviours (network properties) upon which requirements are posed. A ten-member team has a higher chance of being faced with a task for which one of its members knows a solution in advance (or knows a partial solutions satisfying most of the requirements). Thus, the absence of an increase in problem-related messages in the second simulation half may stem from the low frustration levels of team members - as a suitable solution has been found, but not yet accepted. If such is the case, then the data presented indicate that larger teams take longer time to reach consensus. Future work will examine the differences in communication patterns displayed by teams of various sizes more closely.

8 AN EXAMPLE OF DEVELOPED MODEL'S USAGE

The agent-based model of product development team presented in preceding chapters was developed with a goal of enabling simulation and study of a wide range of scenarios. This chapter describes an example of the model's usage. In particular, although the modelled agents do not (as currently implemented) explicitly assess creativity of proposed solutions, this chapter demonstrates how the developed model can be used to further the studies on creativity. The results outlined in this chapter are published in [333].

The capability of modelled agents to learn from past experiences, change its behaviour, and produce new structures can be exploited in studies of the impact of solution space change on the creativity assessment. In other words, one can utilise the developed model to track the solution space expansion and study the dependence between the structure's creativity score and the context in which said creativity was assessed. This chapter takes such situated view of one creativity dimension – novelty. Therefore, the chapter commences with a short introduction in novelty, its definition and measurement. Then, a hypothesis stating that novelty assessment is situated is introduced, and experiments used to test the hypothesis are described. The results of the simulations conducted by utilising the developed agent-based model of a product development team are presented, and their implications for computational creativity studies are discussed.

8.1 Novelty

In order to earn a label 'creative', a design necessarily has to be seen as novel. Boden's [334] theory of creativity marks novelty, alongside with value, as a necessary feature of a creative idea. Similarly, many authors [335] define a creative artefact as the one marked as novel and appropriate, or novel and useful [336], [337].

Earlier studies of creativity [334], [338] regarded novelty as a term covering aspects of originality and surprise. However, later work [339] emphasised that if novelty is viewed as a measure of unusualness or difference of a design in comparison to the existing designs, its definition does not necessarily imply violation of expectations (i.e. surprise) in a space of projected designs. Maher et al. [339] further note that a creative design should be unexpected, thus deeming surprise as a third relevant aspect of creativity. Following such distinction between surprise and novelty, researchers [340], [341] introduced measures of novelty and surprise based on k-means clustering and Euclidean distance. The proposed approach uses design’s features to assign it with a position within the conceptual space. The space of existing and possible designs is clustered utilising the k-means algorithm, and novelty of each design is defined as a distance to its cluster’s centroid. On the other hand, surprise is assessed based on the design’s distance to all of the present centroids. Authors [340], [341] note that, if a new design is distant to all of the existing clusters, it essentially forms a new cluster – thus failing the previously established expectations. Since it uses well-known and easy to implement mechanisms, the described measure is particularly suitable for computational creativity studies. Another measure of computationally-assessed novelty can be found in works of Marsland, Nehmzow, and Shapiro [342] whose novelty-detection approach is based on Self-Organizing Maps. Although the work of Marsland, Nehmzow and Shapiro [342] concerns real-time novelty detection of a mobile robot roaming the environment, its premises are relevant for this work as robot is learning through experience and is situated within its environment. Novelty detection and measurement have been studied within multiple scientific fields. For example, the field of signal processing offers multiple different approaches to novelty detection (for a review, see [343]). The overview of measures used in design field can be found in, for example, the work of Ranjan, Siddharth, and Chakrabarti [344].

8.2 Hypotheses

The aim of the work presented in this chapter is to study how the design’s novelty assessment changes with the change in a design frame. Following the literature presented in previous chapters, a situated stance on design is taken. The situation in which the design occurs shapes the perception of design’s novelty. In other words, novelty - in terms of difference from other designs - of each design is situation-dependent. The hypothesis stating that novelty measurement in design is situated can be detailed through the formulation of two sub-hypotheses:

> *H5: Over the time course of designing, (some of) the solutions that were previously recognised as novel, will become not novel.*
> *H6: Over the time course of designing, (some of) the solutions that were previously not regarded as novel, can be recognised as novel.*

The proposed hypothesis is tested on a series of experiments conducted using the model described in the previous chapter. For this study, however, one modification of the detailed implementation was made. Namely, the original settings deem the simulation as over when the agents find and agree upon a structure that satisfies the requirements. Since this study aims to compare multiple solutions based on their novelty score, it is necessary that agents continuously search and expand the solution space (irrespective of whether some solutions have already been found or not). The mechanism preventing team fixation on a single feasible structure was implemented as follows: if team members propose a unique structure repeatedly over 10 simulation steps, and all of the links from relevant (i.e., required) behaviour nodes to the structure are well-grounded (i.e., the link's weight exceeds the reflex threshold) in each team member agent's mental model, the structure is inhibited and agents reduce the weight of relevant behaviour – structure links. As a consequence, the structure is not used in (at least some) subsequent steps, but the activation level of behaviours connected to the structure remains unchanged, thus influencing the further search. The described mechanism simulates the situation in which all of the team members agree upon one solution and "leave it aside" to produce additional ideas (while still remembering the behaviour and properties of the solution).

8.3 Design of the experiments

To enable structures' novelty assessment and comparison among structures, a representation of each structure within a design space had to be created. Thus, each structure is characterised by some of its respective network's properties. However, as described previously, the task poses constraints on the solution network's properties. As a consequence, structures which can be labelled as solutions necessarily have (at least) some of the properties in common. To distinguish the task-related properties and properties upon which the novelty assessment is performed (and avoid high correlations among them), in work presented in this chapter the tasks have been modified to pose requirements on the properties of the largest connected component of a structure's network. As before, structures' properties (used for novelty assessment) were calculated on the whole respective network. In particular, each structure is characterised by three values: network degree centralisation, average clustering coefficient and hierarchy measure (see Section 5.3). For the study presented here, tasks were modelled to pose requirements on the diameter, closeness and/or betweenness centralities of the largest connected component.

The details on all of the agent-generated structures were collected throughout each simulation run. At each time step, a *reachable structure space* was determined: at a step t, the reachable structure space (RSSt) consist of all structures which can be created by the team in the next step. Thus, it is a space of all structures which can be derived by utilising union and contraction mechanisms (see Section 5.1.1) on the structures known to agents at the step t. At each step t, a subset of structures which meet the requirements can be extracted from the reachable structure space (RSSt). Such subset forms a *reachable feasible solutions space* (RFSSt): a space of all structures meeting the task requirements which can be generated in the step $t + 1$. Finally, a subset of all reachable feasible solutions can be considered as novel, thus constituting a space called *reachable novel solution space* (RNSSt). Mahalanobis distance was used to detect feasible solutions sufficiently different from others, thus extracting RNSSt from its corresponding RFSSt. Mahalanobis distance is frequently used to detect outliers in multivariate data and has often been utilised in machine learning systems to identify data distinct from the samples used for the system's training [343], [345].

Overall, 300 simulations were run, each terminating when the size of a reachable structure space reached 100,000 nodes.

8.4 Results

The average sizes and standard deviations of the size of reachable structure space (RSS) and reachable feasible solution space (RFSS) over time are represented in Figure 8.1. Similarly, Figure 8.2a depicts the corresponding values for the number of novel solutions encountered (RNSS). Since the simulation experiments were of varying lengths, to average over all of the simulation experiments, the data from each simulation run was sampled to extract values obtained at a hundred of equidistant time steps (essentially representing the percentage of simulation duration covered). Finally, to express the relation between the size of the feasible structure space and the number of (feasible) novel structures, the percentage of feasible structures which were marked as novel is presented in Figure 8.2b.

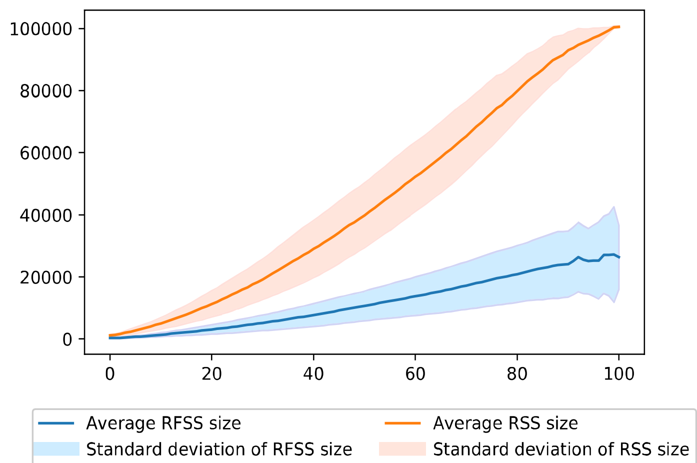


Figure 8.1 The average number (and standard deviation) of reachable structures and reachable feasible solution over time

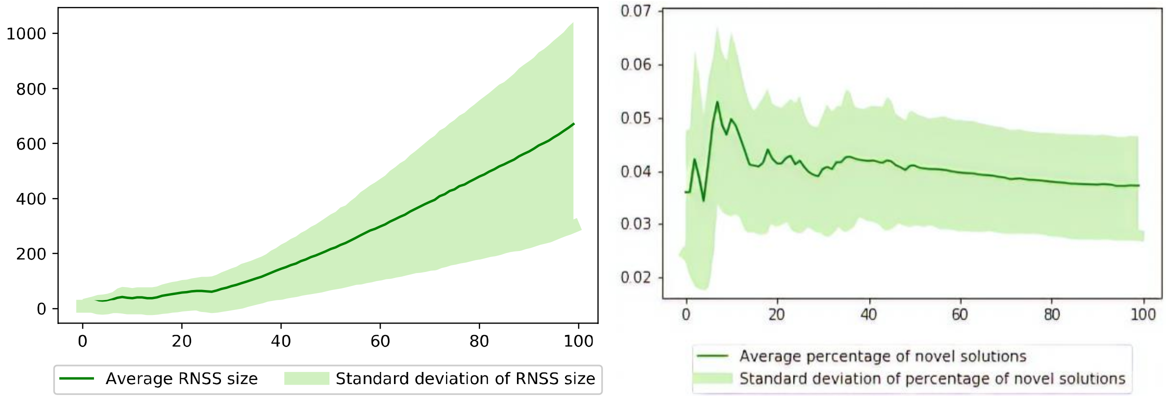


Figure 8.2 The average a) number (and standard deviation) of novel solutions over time and b) percentage (and standard deviation) of novel solutions over time

To test for the hypotheses posed in Section 8.2, the statistics on the number of structures which – throughout simulation - turned from being labelled as novel to being regarded as not novel was collected. Similarly, each structure which initially (i.e., at the time of their first occurrence within reachable structure space) was not seen as novel, only to be labelled as such as the simulation progressed was counted. The average (aggregate) numbers of solutions turned from novel to not novel, or from not novel to novel, and the respective standard deviations are given in Table 8.1. Finally, to illustrate the dynamics of simulated experiments and to provide deeper insights into the obtained results, Figure 8.3 presents a series of snapshots extracted from one simulation run. The figure shows feasible solutions and further differentiates them based on their novelty status. The structures coloured in red are those that are more than two standard deviations apart from the sample mean. In other words, they are considered novel.

Table 8.1 Statistics on the number of novel solutions turned nonnovel and nonnovel solutions turned novel

Number of **Novel -> Not Novel** transitions per simulation		Number of **Not Novel -> Novel** transitions per simulation	
Average	Standard deviation	Average	Standard deviation
187	167.06	653	256.88

8.5 Discussion

Figure 8.1 demonstrates how implemented mechanisms of union and concatenation enable agents to extend the solution space and to create new (feasible or not) structures over time. In accordance, as seen in Figure 8.2a, the number of novel solutions found in each simulation step increases throughout the simulation. On the other hand, the percentage of feasible solutions recognised as novel slowly declines over time (Figure 8.2b).

More precisely, during the early stages of simulation (the first one-fifth of the simulation run), the number of novel solution is small (less than 100). In this period, the percentage of novel solutions shows several increases and decreases, accompanied by a large standard deviation, thus reflecting the differences between simulation runs. Namely, in some cases, agents needed several steps to find novel solutions. At other runs, however, the structures generated at the early stages of the simulation were diverse, and thus several were already labelled as novel. As the simulation progresses, the percentage stabilises and starts to decline.

It is interesting to note that significantly more structures turned from not novel to novel, than the other way around (see Table 8.1). To understand this, one may take a look at Figure 8.3, which displays the dynamics of one simulation run. The top left subfigure displays the RFSS at the simulation start, where one can notice several nodes on the left side of the space marked as novel. However, as the simulation progresses and agents learn and create more structures, the RFSS changes to include structures closer to those previously regarded as new. As a consequence, the notion of what is novel (i.e., different from others) gradually shifts from the left to the far right of the space (third subfigure). As the process continues, the space of reachable feasible structures grows, and the majority of structures in RFSS concentrate in a cluster on the left of the space. Consequently, the standard deviation of the population decreases and a relatively large number of structures which are at the outskirts of the cluster will now be labelled as novel. This example demonstrates how agent-produced solutions, after some time, start to converge to “similar looking” structures satisfying the requirements. As already discussed in the previous chapters, such a process is a consequence of two things: first, the lack of the agent’s perception of the structure’s novelty. Since the agents are unable to determine the novelty score of the proposed solutions, they are restricted from detecting that subsequently generated structures moved the space towards increasingly similar structures. Secondly, the implemented synthesis mechanisms (union and contraction, among which union is more frequently applied) led to the generation of larger, well-connected networks for which calculated measures differ to a lesser extent. Due to these limitations of the system, over time, the majority of reachable structures “concentrates”. As a consequence, the number of structures which will be marked as novel and then changed to not novel decreases over time. On the contrary, as the simulation progresses, the large proportion of the previously not novel structures will become regarded as novel.

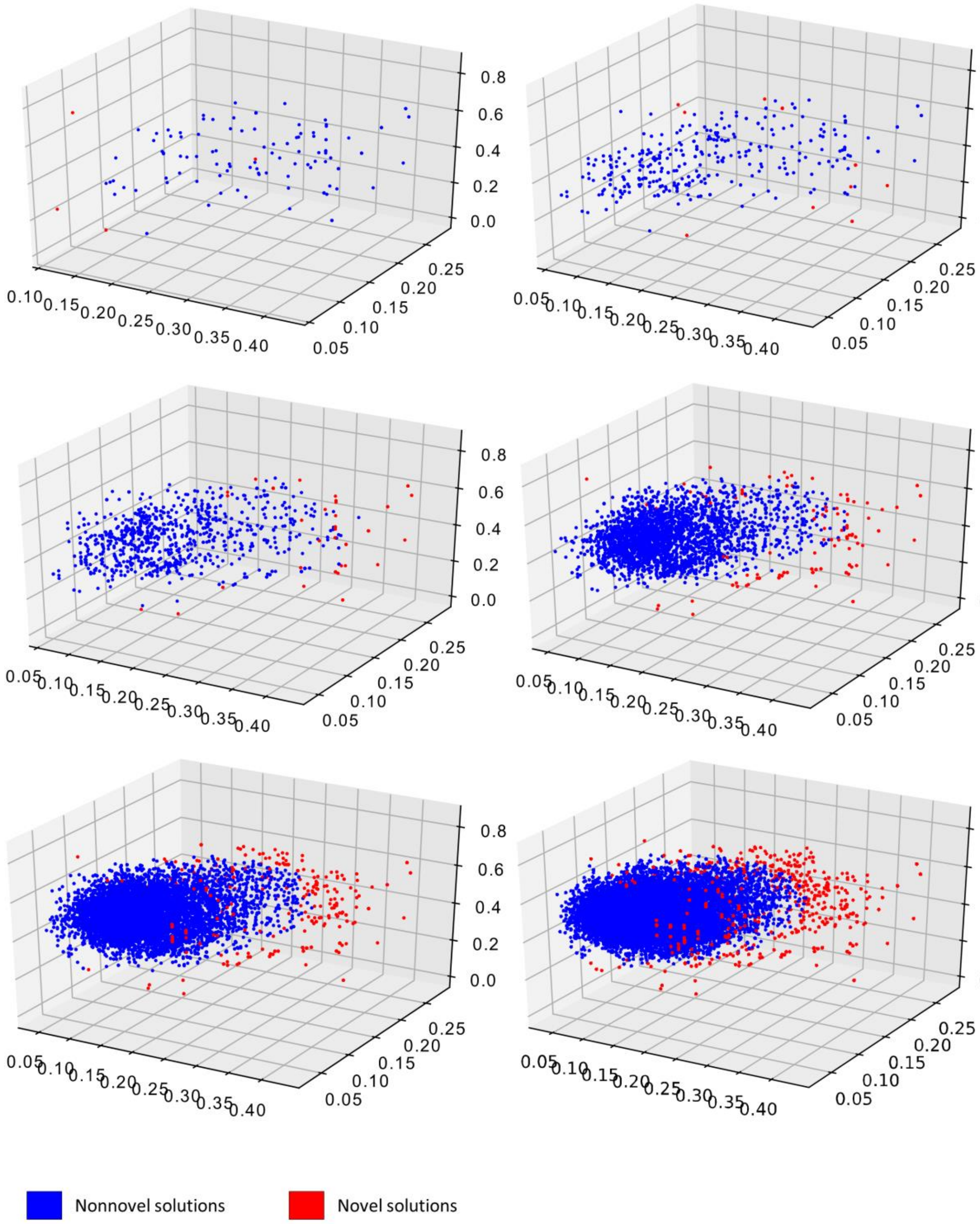


Figure 8.3 An example of the simulation run

Despite the discussed limitations, the obtained results enable interesting insights. If a task is reframed to broaden the solution space, the difference in characteristics between the initial and newly obtained space may cause some solutions to no longer be regarded as novel. Perhaps more surprisingly, the same act of broadening the solution space can cause some (previously uninteresting) solutions to stand out. Another way of discussing this effect may be through the notion of "norms" and development of immutable expectations. An example presented by Maguire, Maguire, and Keane [346], [347] emphasises the importance of width of knowledge in assessing surprise, noting that a "flying rabbit" may not be as surprising to a child as it would be to an adult. Although the present work is not primarily concerned with surprise, the complementarity of the findings can easily be deduced. For example, a novice designer aware only of the solutions presented in the first subfigure of Figure 8.3 would differently assess novelty of each design than would do an expert familiar with broader design space. In a sense, the large proportion of structures concentrated in a cluster over time created a "norm" of how agent-generated solutions look like, thus shaping the perception of what is "different."

An interesting direction for future work may consist of studying whether similar results would be obtained if the agents were modelled as the mentioned robot with real-time novelty-detection mechanism [342]. Marsland et al. [342] robot has habituation and recovery mechanisms which enable the agent to get accustomed to stimuli, develop different value system and forget. As a result, the agent sometimes marks a stimulus as novel even though the stimulus has already been encountered in previous stages of the simulation.

8.6 Conclusion

This chapter builds on the notion of situatedness and uses it to demonstrate how the computational model introduced in previous chapters can be used to study creative aspects of design. Utilising the computational model and relying on the existing empirical findings and computational models of creativity in design, this chapter explored how a vital aspect of creativity – novelty – changes with respect to a situation. A series of computational experiments were conducted using the developed agent-based model of a product development team with the aim of studying how agents explored and extended the solution space. At each simulation step, structures present in the solution space were categorised based on their assessed novelty (i.e. difference from other solutions) and the changes in structures' novelty status over a simulation run were observed. The obtained results demonstrated the importance of a frame within which the design is occurring in assessing novelty. As a design frame (or a situation as defined by Kelly and Gero [348]) is changing to broaden the solution space, a design may turn from appearing as novel to being regarded as not different from the majority of others. In contrast, a change of a design frame may cause a design previously marked as 'typical' (i.e., not sufficiently different from other solutions), to be considered as interesting and different from others.

The performed computational simulations utilised the Mahalanobis distance to assess novelty status, i.e. to detect structures distant from the general distribution of designs. It can be studied if similar results would be obtained by using previously established metrics (e.g., [339]). Thus, future experiments can be directed towards comparing different measures and assessing their capability to capture the notion of novelty. Such studies will help determine whether the observed trends are emerging from the chosen novelty measure. Furthermore, once insufficiencies in the agent's synthesis mechanisms are mitigated, the experiments similar to the ones presented here will be conducted, and the obtained results will be matched against the results presented herein. An additional limitation of the presented study is the fact that the developed framework does not enable the dynamic introduction of new variables along which solutions can differ. It is likely that some of the subsequently added structures which were determined to "belong to the cluster of similar ideas" would be found as novel based on some additional dimension (i.e., different network characteristic). In the present study, the difference in designs is manifested through either new values for a particular attribute or as a novel combination of attributes. But, as Gero [128] postulated and Maher and Fisher (2012) demonstrated on the example of Bloom Laptop design, new (and surprising) designs can introduce new variables (i.e., dimensions) along which designs can differ.

Nevertheless, the conducted experiments emphasise the importance of regarding novelty as a situated measure. Numerous examples from fashion [349], or even healthcare industry [350], illustrate how "old" designs can again come under the spotlight due to the change in the contextual factors. Finally, there is still much to be understood about the relationship of the core aspects of creativity (novelty, value and surprise). For example, research regarding questions such as: how come that some designs are rated as highly creative, but quickly lose their appeal (e.g., [349] notes such examples in fashion industry), while others maintain their creative status over long periods of time; how different, useful and surprising should the product be to be regarded as creative and how does each of these aspects of creativity change over time; – and many others would benefit from taking the situated perspective. The developed agent-based model of a product development teamwork may again prove to be useful in such studies.

9 CONCLUSION

The final chapter concludes the work by reflecting on the research aim, hypothesis and scientific contributions formulated in the introductory chapter of this work. The developed model is discussed with regards to the existing agent-based models of product development teams, and its capabilities, as well as limitations, are emphasised. Finally, the chapter discusses the directions for future research efforts which build on the identified shortcomings and the model's capabilities.

9.1 Reflection on the developed model

The research presented in this thesis is motivated by the potential of computational models to provide a cost-effective, easily-controllable environment for various studies of teams. In particular, the need for improved understanding of design teams' cognitive behaviour and the emergence of team properties and processes, renders simulation frameworks as valuable research tools within a product development domain. Thus, the work presented herein strived to develop a theoretical and computational model of a product development team which enables studies of emergent team properties and behaviours.

To support studies of diverse aspects of teamwork, this thesis presented a synthesis of theories and results from multiple research domains and used it to develop a multi-purpose simulation tool. Following the literature review regarding the important components of human behaviour modelling and team performance studies (Chapter 4), the model encompasses cognitive, social, affective and situated aspects of designer's work. The prominent theories and empirical research on cognition, human thinking and emotions formed the basis for the model development. Similarly, the existing cognitive architectures and agent-based models of design teams offered many points of interest and informed the model implementation. For example, the Bott and Mesmer's [105] approach to modelling the agent's mental processes demonstrated the benefits of employing design ontologies in representing the agent's knowledge. The model developed in this work utilised similar idea and furthered coupled it with the algorithms and approaches used in well-known cognitive architectures (e.g. [170], [171]). Similarly, the model developed in [78] inspired the implementation of the agent's search for the solution as the process which builds on the previous knowledge and reuses it to generate new, different (in terms of their behaviour) solutions. The aspects of the developed model such as the representation of the agents' domain of expertise, trust, transactive memory, cognitive ability and personality all stem from the ideas obtained through a comprehensive review of the state-of-the-art (Chapter 3). Building on the previous work promotes comparison with the existing models and can be used to achieve greater validity of the models. For example, the similarity of the conceptualisations of transactive memory implemented in Singh et al. [131] model and the model presented in this work enables the replication of the experiments conducted in [131] and, consequently, corroboration of the reported findings.

The model presented herein, thus, integrated and further extended various aspects of existing models and theories. In order to ensure internal validity and enable a seamless integration of the multiple model elements, a structured, hierarchical testing framework was employed (Chapter 6). As a result of integration, extension and testing, a coherent simulation framework for team studies is obtained. Referring back to the essential elements of team performance models discussed in [38], one can note that the developed model addresses the vast majority of aspects within 'must-be-modelled' and 'should-be-modelled' categories. The detailing of the agent's characteristics and interactions enables conducting a great variety of experiments. A sample of the possible studies is presented in Chapter 6. However, the number and a broad scope of implemented aspects increase the model complexity and may pose difficulties in the results' interpretation and validation. To mitigate these limitations, the model's implementation followed the approach taken in the PECS framework [148] and separated the agent's cognition, affect, personality and cognitive ability, and trust and transactive memory into the respective components. The components can easily be excluded from the simulation, modified and extended, and interactions among components can be redefined to study the interplay among the modelled elements.

The most elaborated element of the developed agent-based model is the agent's cognitive behaviour. The model is implemented with a specific aim of capturing the details on one's information processing, new structure generation, derivation of new knowledge links, memory formation and retrieval, knowledge grounding and forgetting, and learning from interaction with others. As a result, this work presents one of the most elaborate models of design cognition (among the implemented agent-based models of design teams reviewed in Chapter 3). The detailing of agents' knowledge acquisition, and knowledge sharing among team members, provides the means for studying team learning as an emergent team property [167]. Individual and team learning demonstrates as a change in the behaviour, i.e. adaptation. The capability of the developed model to enable studies of team learning and adaptation is demonstrated in the experiments reported in Chapter 7. In the introductory chapter of this work, the research aim was stated as follows: "*The main goal of the doctoral research is to develop and validate a* *theoretical and computational model of teamwork in order to enable simulation and study of team's capability to adapt to changes in circumstances occurring during the product development activities*". The results reported in Chapter 7 demonstrate the model's capability to satisfy the intended goal of this thesis and, in turn, offer support for the hypothesis proposed at the beginning of the study reported in this work.

As discussed in Section 1.4, the scientific contribution of the work reported in this thesis is threefold and manifests in:

1. *A theoretical model of a product development team which consists of a model of an individual team member, a model of a team environment regarding tasks and resources, and a definition of mechanisms guiding the interactions among team members, and between team members and their environment*. The work in Chapter 4 specified important aspects and the theoretical background of the developed model. Building on this, Chapter 5 introduces the details of the developed theoretical model of a product development team. The model description is organised into sub-sections that represent each of the model elements (i.e. model of an individual member, a model of a team environment, and a model of interactions). The developed model encompasses many important aspects of the design task, designer's behaviour, and their interactions.
2. *A computational prototype in the form of a multi-agent system which is built based on the theoretical model and which enables the study of emergent team properties and behaviour.* Along with the description of the theoretical model, the implementation details are presented in Chapter 5. The capabilities of the developed multi-agent system to capture numerous aspects of design teamwork and enable studies of emergent team properties are further explored in Chapter 6.
3. *A framework for calibration and validation of the developed model and computational prototype of team behaviour, with a focus on the component of the team adaptability and learning.* In order to ensure the internal validity of the model, and to increase the credibility of its results, numerous aspects of the developed model are tested (Chapter 6). In particular, the hierarchical testing was employed where the model's performance was first examined in the simple settings comprising a single agent performing a task. A gradual increase in the simulated complexity enabled detail insights into the performance and interplay of the implemented mechanisms. Finally, the model (in its full functionality) is tested to study the model's capability to represent trends in design

teams' change in behaviour resulting from the previous experience gain, i.e. adaptation due to learning. The results of such experiments are reported in Chapter 7.

The listed scientific contributions, taken altogether, form the desired 'computational laboratory' for design team studies, serving for the theory-testing and hypothesis formation.

9.2 Limitations

The developed model should by no means be regarded as 'finished'. As emphasised in the first chapter, the work presented in this thesis only sets the ground for future model developments and iterative refinements based on new empirical findings and research efforts. As such, numerous shortcomings of the conducted research and derived model implementation are emphasised throughout Chapter 5. Each of the implemented model's components should be further refined, and the greater emphasis should be placed on the studies of the interdependencies among the cognitive, affective and social aspects of human behaviour.

The most significant limitation of the current research stems from the lack of application of the simulated experiments to real-world cases. Similarly, due to the shortcomings of the implemented communication mechanisms (see Section 5.2), the simulation output regarding the agents' interactions cannot readily be calibrated to the data obtained through protocol studies. Similar to many of the existing models, the model presented in this work is a generator: it explores whether implemented theories can produce behavioural patterns observed in the real world, and enables experiments aimed at increasing the understanding of little-understood aspects of design teamwork. In other words, the developed model is directed towards offering possible explanations, directions for future work and hypothesis formation. For such use cases - in which acquiring a highly-accurate prediction of the real-world outcomes is not the simulation goal - a comprehensive calibration and statistical validation of the model is not necessary [313]. Nevertheless, the future work should be directed towards refining the model in a manner which enables more direct comparisons to the available real-world data.

Modelling of tasks is an additional aspect of the model, which requires further attention to ease the understanding of the simulation outcomes. In particular, a study of the interdependences of the network properties (used to simulate structure's behaviours) should be conducted, and the obtained results should serve to determine the task generation mechanisms which enables full control of the task difficulty. Relatedly, the task definition should be refined to capture the requirement of novelty, as well as the ambiguity, dynamism and conflicts among the design requirements. To equip agents with the capability to respond to requirements posed on the design's creativity level, the agent's personality traits and affective mechanisms should be further developed. The absence of conflicting, ambiguous or ill-defined requirements, as well as the static nature of the implemented task requirements, emphasise that the developed model has limitations similar to the majority of models reviewed in Chapter 3. Namely, the model assumes the existence of the objective performance function with multiple optima, captures only the technical aspects of the design's performance and neglects fluctuations in customer preferences.

There is also a need for refinement of the agent's cognitive behaviour regarding the nature of working memory capacity, differences in processing sensorial input, cognitive phenomena such as habituation and other mechanisms guiding the agent's attention shifting. The implemented agent's affective mechanisms are very simple and short-term (e.g. no modelling of moods and states, and the frustration from one task does not impact the next tasks). Implementation is missing for several personality traits. Noise is not affecting the agent's communication. The impact of distrust, generalisations formed from transactive memory, or cultural differences on the interactions are not modelled. And many other aspects of the implemented framework require further research and refinement (see Chapter 5 for a detail discussion).

Finally, one can discuss the limitations of the computational approaches used in the current model's implementation. For example, one can consider how to implement the agent's learning and memory formation more efficiently and how the implementation of the algorithms such as spreading activation, can be optimised. In addition, the current implementation takes a sequential approach to schedule the simulated agents' and runs a single simulation at the time. But parallelisation can be employed to speed up the simulation runs.

9.3 Directions for future research

Following the discussed limitations of the developed model, numerous future work directions aimed at refining and extending the model can be derived. However, the high flexibility of the model presented in this work also offers several research opportunities for which little to no model extensions are needed. An example of the docking of Singh et al. [131] model and the developed model has already been mentioned. A similar type of comparison can, for example, be conducted when extracting the influence of trust on team performance and comparing the results to those obtained in [130] model.

The detail modelling of the agent's cognition can be used to extract the similarities among agents' mental modes, thus permitting studies on the conditions facilitating the emergence of shared mental models. Studies of team adaptability can be furthered by simulating membership change and studying its impact on team performance. For example, an experiment may consist of removing or adding agents to the team in the middle of the task performance (or in between two tasks), observing the achieved team performance and behaviour, and comparing it to the one the team would display if no change in membership occurred. Similarly, one can study how knowledge transfer occurs between the teams by simulating the agents with temporary membership, as described in models [44] and [132].

A research question of great importance for supporting the design teamwork relates to the emergence of team fixation and the possible strategies to prevent or reduce its effects. The developed model may offer a basis for simulations directed at tackling this question.

Sub-team formation may offer another stream of research for which the developed model can be readily utilised. The study [351] has shown that, when teams consist of ten or more members, sub-teams start to emerge, thus hindering the effectiveness of team interactions [62]. The recent work on co-design and sub-team formation [274] within design teams may offer multiple points of comparison among the real-world teams and the simulated agents'.

Aside from team cognition, one can study the emergence of team cohesion and climate. Such studies would require detailing the agent's motivational state and social interactions. Nevertheless, such research would provide insights into an emergent team state critical for team effectiveness [26] that is seldom explored by the existing models (see Table 3.1).

In addition to team cohesion, one can study how one of the 'Big five' components of teamwork - team leadership - emerges within a team irrespective of the hierarchy imposed by the organisation. The reputation parameter implemented in the developed model may serve as a starting point for such studies. However, detailing the individual's motivation and attitudes ([106]) should be conducted to enable more detail insights.

Each of the listed future research directions necessarily results in a refinement of the developed model. Namely, following the discussion presented in Section 1.3, the model and the body of knowledge are interconnected in an iterative loop where findings obtained by the model trigger a new cycle of refinement. In future refinements, special attention should be given to detailing the agent's knowledge of the social context (regarding culture and norms derived from past experiences) [150]. The authors in [150] also emphasise the detailing of the agent's cognitive constraints and biases (e.g. the effect of emotions). They argue that enabling a high level of social context knowledge, coupled with a realistic representation of cognitive constraints, leads to a highly veridical representation of human behaviour in a social setting.

REFERENCES

[1] Devine, D. J., Clayton, L. D., Philips, J. L., Dunford, B. B., Melner, S. B., "Teams in Organizations: Prevalence, Characteristics, and Effectiveness", Small Group Research, Vol. 30, No. 6, Dec. 1999, pp. 678–711, doi: 10.1177/104649649903000602.

[2] Kozlowski, S. W. J., Bell, B. S., "Work Groups and Teams in Organizations", in Handbook of Psychology, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003, pp. 333–375.

[3] Edmondson, A. C., Nembhard, I. M., "Product development and learning in project teams: The challenges are the benefits", Journal of Product Innovation Management, Vol. 26, No. 2, 2009, pp. 123–138, doi: 10.1111/j.1540-5885.2009.00341.x.

[4] Chen, S.-J., Lin, L., "Modeling Team Member Characteristics for the Formation of a Multifunctional Team in Concurrent Engineering", IEEE Transactions on Engineering Management, Vol. 51, No. 2, May 2004, pp. 111–124, doi: 10.1109/TEM.2004.826011.

[5] Ensici, A., Badke-Schaub, P., Bayazıt, N., Lauche, K., "Used and rejected decisions in design teamwork", CoDesign, Vol. 9, No. 2, Jun. 2013, pp. 113–131, doi: 10.1080/15710882.2013.782411.

[6] Sarin, S., Mahajan, V., "The Effect of Reward Structures on the Performance of Cross-Functional Product Development Teams", Journal of Marketing, Vol. 65, No. 2, Apr. 2001, pp. 35–53, doi: 10.1509/jmkg.65.2.35.18252.

[7] Mumford, M. D., Feldman, J. M., Hein, M. B., Nagao, D. J., "Tradeoffs Between Ideas and Structure: Individual Versus Group Performance in Creative Problem Solving", The Journal of Creative Behavior, Vol. 35, No. 1, Mar. 2001, pp. 1–23, doi: 10.1002/j.2162-6057.2001.tb01218.x.

[8] Hsu, Y., "Work values, conflict, and team cooperation among engineering designers", Journal of Engineering Design, Vol. 28, No. 10–12, Dec. 2017, pp. 799–820, doi: 10.1080/09544828.2017.1397268.

[9] Diehl, M., Stroebe, W., "Productivity loss in brainstorming groups: Toward the solution of a riddle.", Journal of Personality and Social Psychology, Vol. 53, No. 3, Sep. 1987, pp. 497–509, doi: 10.1037/0022-3514.53.3.497.

[10] Gyory, J. T., Cagan, J., Kotovsky, K., "Are you better off alone? Mitigating the underperformance of engineering teams during conceptual design through adaptive

process management", Research in Engineering Design, Vol. 30, No. 1, Jan. 2019, pp. 85–102, doi: 10.1007/s00163-018-00303-3.

[11] McComb, C., Cagan, J., Kotovsky, K., "Optimizing Design Teams Based on Problem Properties: Computational Team Simulations and an Applied Empirical Test", Journal of Mechanical Design, Vol. 139, No. 4, Apr. 2017, doi: 10.1115/1.4035793.

[12] McComb, C., "Designing the Characteristics of Design Teams via Cognitively Inspired Computational Modeling", Carnegie Mellon Univeristy, 2016.

[13] Alnuaimi, O. A., Robert, L. P., Maruping, L. M., "Team Size, Dispersion, and Social Loafing in Technology-Supported Teams: A Perspective on the Theory of Moral Disengagement", Journal of Management Information Systems, Vol. 27, No. 1, Jul. 2010, pp. 203–230, doi: 10.2753/MIS0742-1222270109.

[14] Janis, I., "Groupthink", in A first look at communication theory, Griffin, E., (ed.) McGrawHill, New York, NY, USA, 1991, pp. 235–246.

[15] Sivasubramaniam, N., Liebowitz, S. J., Lackman, C. L., "Determinants of New Product Development Team Performance: A Meta-analytic Review", Journal of Product Innovation Management, Vol. 29, No. 5, Sep. 2012, pp. 803–820, doi: 10.1111/j.1540-5885.2012.00940.x.

[16] Kratzer, J., Leenders, R. T. A. J., Van Engelen, J. M. L., "The social structure of leadership and creativity in engineering design teams: An empirical analysis", Journal of Engineering and Technology Management, Vol. 25, No. 4, Dec. 2008, pp. 269–286, doi: 10.1016/j.jengtecman.2008.10.004.

[17] Sosa, R., "Computational Modelling of Teamwork in Design", in Experimental Design Research, Springer International Publishing, Cham, 2016, pp. 173–186.

[18] Martinec, T., "A model of information processing and interactions in teams developing technical systems", University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, 2019.

[19] Gero, J. S., Mc Neill, T., "An approach to the analysis of design protocols", Design Studies, Vol. 19, No. 1, Jan. 1998, pp. 21–61, doi: 10.1016/S0142-694X(97)00015-X.

[20] Behoora, I., Tucker, C. S., "Machine learning classification of design team members' body language patterns for real time emotional state detection", Design Studies, Vol. 39, Jul. 2015, pp. 100–127, doi: 10.1016/j.destud.2015.04.003.

[21] Visser, W., Maher, M. Lou, "The role of gesture in designing", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 25, No. 3, Aug. 2011, pp. 213–220, doi: 10.1017/S0890060411000047.

[22] Hay, L., Duffy, A. H. B., McTeague, C., Pidgeon, L. M., Vuletic, T., Grealy, M., "A systematic review of protocol studies on conceptual design cognition: Design as search and exploration", Design Science, Vol. 3, Jul. 2017, p. e10, doi: 10.1017/dsj.2017.11.

[23] Paris, C. R., Salas, E., Cannon-Bowers, J. A., "Teamwork in multi-person systems: a review and analysis", Ergonomics, Vol. 43, No. 8, Aug. 2000, pp. 1052–1075, doi: 10.1080/00140130050084879.

[24] Miller, K. D., "Agent-Based Modeling and Organization Studies: A critical realist perspective", Organization Studies, Vol. 36, No. 2, Feb. 2015, pp. 175–196, doi: 10.1177/0170840614556921.

[25] Epstein, J. M., "Why Model?", JASSS, Vol. 11, No. 4, 2008, p. 12.

[26] Kozlowski, S. W. J., Chao, G. T., "The Dynamics of Emergence: Cognition and Cohesion in Work Teams", Managerial and Decision Economics, Vol. 33, No. 5–6, Jul. 2012, pp. 335–354, doi: 10.1002/mde.2552.

[27] Macal, C. M., North, M. J., "Tutorial on agent-based modelling and simulation", Journal of Simulation, Vol. 4, No. 3, Sep. 2010, pp. 151–162, doi: 10.1057/jos.2010.3.

[28] Hevner, March, Park, Ram, "Design Science in Information Systems Research", MIS Quarterly, Vol. 28, No. 1, 2004, p. 75, doi: 10.2307/25148625.

[29] Perišić, M. M., Štorga, M., Podobnik, V., "Agent-Based Modelling and Simulation of Product Development Teams", Tehnicki vjesnik - Technical Gazette, Vol. 25, No. Supplement 2, Sep. 2018, pp. 524–532, doi: 10.17559/TV-20170912134413.

[30] Salas, E., Dickinson, T. L., Converse, S. A., Tannenbaum, S. I., "Toward an understanding of team performance and training", in Teams: Their training and performance, Swezey, R. W., Salas, E., (eds.) Ablex Publishing, 1992, pp. 3–29.

[31] Kozlowski, S. W. J., Ilgen, D. R., "Enhancing the Effectiveness of Work Groups and Teams", Psychological Science in the Public Interest, Vol. 7, No. 3, Dec. 2006, pp. 77–124, doi: 10.1111/j.1529-1006.2006.00030.x.

[32] Salas, E., Sims, D. E., Burke, C. S., "Is there a “big five” in teamwork?", Small Gr. Res.,

Vol. 36, No. 5, 2005, pp. 555–599, doi: 10.1177/1046496405277134.

[33] Kozlowski, S. W. J., Grand, J. A., Baard, S. K., Pearce, M., "Teams, Teamwork, and Team Effectiveness: Implications for Human Systems Integration", in The handbook of human system integration, Boehm-Davis, D., Durso, F., Lee, J., (eds.) 2015.

[34] Marks, M. A., Mathieu, J. E., Zaccaro, S. J., "A Temporally Based Framework and Taxonomy of Team Processes", Academy of Management Review, Vol. 26, No. 3, Jul. 2001, pp. 356–376, doi: 10.5465/amr.2001.4845785.

[35] Kozlowski, S. W. J., Klein, K. J., "A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes", in Multilevel theory, research, and methods in organizations: Foundations, extensions, and new directions, Klein, K. J., Kozlowski, S. W. J., (eds.) Jossey-Bass, 2000, pp. 3–90.

[36] Hülsheger, U. R., Anderson, N., Salgado, J. F., "Team-level predictors of innovation at work: A comprehensive meta-analysis spanning three decades of research.", Journal of Applied Psychology, Vol. 94, No. 5, 2009, pp. 1128–1145, doi: 10.1037/a0015978.

[37] Bell, B. S., Kozlowski, S. W. J., Blawath, S., "Team Learning: A Theoretical Integration and Review", in The Oxford handbook of organizational psychology, Vol. 2., Kozlowski, S. W. J., (ed.) Oxford University Press, Oxford, UK, 2012, pp. 859–909.

[38] Salas, E., Guthrie, J. W., Wilson-Donnelly, K. A., Priest, H. A., Burke, C. S., "Modeling team performance: The basic ingredients and research needs", in Organizational simulation, Wiley Online Library, 2005, pp. 185–228.

[39] Andreasen, M. M., Hansen, C. T., Cash, P., Conceptual Design: Interpretations, Mindset and Models. Springer International Publishing, Cham, 2015.

[40] Balestrini Robinson, S., "A modeling process to understand complex system architectures", Georgia Institute of Technology, 2009.

[41] Oyama, K., Learmonth, G., Chao, R., "Applying complexity science to new product development: modeling considerations, extensions, and implications", J. Eng. Technol. Manage., Vol. 35, 2015, pp. 1–24, doi: 10.1016/j.jengtecman.2014.07.003.

[42] Badke-Schaub, P., Neumann, A., Lauche, K., Mohammed, S., "Mental models in design teams: a valid approach to performance in design collaboration?", CoDesign, Vol. 3, No. 1, 2007, pp. 5–20, doi: 10.1080/15710880601170768.

[43] Badke-Schaub, P., Frankenberger, E., "Analysis of design projects", Design Studies, Vol. 20, No. 5, Sep. 1999, pp. 465–480, doi: 10.1016/S0142-694X(99)00017-4.

[44] Gero, J. S., Kannengiesser, U., "Modelling Expertise of Temporary Design Teams", Journal of Design Research, Vol. 4, No. 3, 2004, pp. 1–13, doi: 10.1504/JDR.2004.009840.

[45] Hubka, V., Eder, W. E., Engineering design. Zürich: Heurista, 1992.

[46] Bissola, R., Imperatori, B., Colonel, R. T., "Enhancing the Creative Performance of New Product Teams: An Organizational Configurational Approach", Journal of Product Innovation Management, Vol. 31, No. 2, Mar. 2014, pp. 375–391, doi: 10.1111/jpim.12101.

[47] Erdős, P., Rényi, A., "On Random Graphs. I", Publicationes Mathematicae, Vol. 6, 1959, pp. 290–297.

[48] Newman, M. E. J., "Random graphs as models of networks", arXiv preprint cond-mat/0202208, 2002.

[49] Watts, D. J., Strogatz, S. H., "Collective dynamics of ‘small-world’ networks", Nature, Vol. 393, No. 6684, 1998, pp. 440–442, doi: 10.1038/30918.

[50] Albert, R., Barabási, A.-L., "Statistical mechanics of complex networks", Rev. Mod. Phys., Vol. 74, No. 1, 2002, pp. 47–97.

[51] Forrester, J. W., Industrial dynamics. The MIT Press, Massachusetts, Cambridge, 1961.

[52] Behdani, B., "Evaluation of paradigms for modeling supply chains as complex socio-technical systems", in *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 2012, pp. 1–15, doi: 10.1109/WSC.2012.6465109.

[53] Gordon, G., System simulation. Prentice-Hall, 1969.

[54] Chan, W. K. V., Son, Y.-J., Macal, C. M., "Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation", in *Proceedings of the Winter Simulation Conference*, 2010, pp. 135–150, doi: 10.1109/WSC.2010.5679168.

[55] Heath, S. K., Buss, A., Brailsford, S. C., Macal, C. M., "Cross-paradigm simulation modeling: challenges and successes", in *Proceedings of the Winter Simulation Conference*, 2011, pp. 2783–2797, doi: 10.1109/WSC.2011.6147983.

[56] Schelling, T. C., "Dynamic models of segregation†", J. Math. Sociol., Vol. 1, No. 2, 1971, pp. 143–186, doi: 10.1080/0022250X.1971.9989794.

[57] Fernandes, J. V., Henriques, E., Silva, A., Pimentel, C., "Modelling the dynamics of complex early design processes: an agent-based approach", Design Science, Vol. 3, Oct. 2017, p. e19, doi: 10.1017/dsj.2017.17.

[58] Mkrtchyan, A. A., "Modeling distributed product development processes in small and medium enterprises", Massachusetts Institute of Technology, 2015.

[59] Secchi, D., "A case for agent-based models in organizational behavior and team research", Team Performance Management: An International Journal, Vol. 21, No. 1/2, Mar. 2015, pp. 37–50, doi: 10.1108/TPM-12-2014-0063.

[60] Secchi, D., Neumann, M., Agent-Based Simulation of Organizational Behavior. Springer International Publishing, Cham, 2016.

[61] Kozlowski, S. W. J., Chao, G. T., Grand, J. A., Braun, M. T., Kuljanin, G., "Capturing the multilevel dynamics of emergence: Computational modeling, simulation, and virtual experimentation", Organizational Psychology Review, Vol. 6, No. 1, Feb. 2016, pp. 3–33, doi: 10.1177/2041386614547955.

[62] Dionne, S. D., Sayama, H., Hao, C., Bush, B. J., "The role of leadership in shared mental model convergence and team performance improvement: An agent-based computational model", Leadership Quart., Vol. 21, No. 6, 2010, pp. 1035–1049, doi: 10.1016/j.leaqua.2010.10.007.

[63] McComb, S. A., "Mental Model Convergence: The Shift from being an Individual to being a Team Member", in Multi-Level Issues in Organizations and Time (Research in Multi-Level Issues, Vol. 6), Dansereau, F., Yammarino, F., (eds.) Emerald Group Publishing Limited, Bingley, 2007, pp. 95–147.

[64] Hughes, H. P. N., Clegg, C. W., Robinson, M. A., Crowder, R. M., "Agent-based modelling and simulation: The potential contribution to organizational psychology", Journal of Occupational and Organizational Psychology, Vol. 85, No. 3, 2012, pp. 487–502, doi: 10.1111/j.2044-8325.2012.02053.x.

[65] Schreiber, C., Singh, S., Carley, K. M., "Construct-a multi-agent network model for the co-evolution of agents and socio-cultural environments", 2004.

[66] Schreiber, C., Carley, K. M., "Validating agent interactions in construct against empirical

communication networks using the calibrated grounding technique", IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43, No. 1, 2013, pp. 208–214.

[67] Hong, L., Page, S. E., "Groups of diverse problem solvers can outperform groups of high-ability problem solvers", Proceedings of the National Academy of Sciences of the United States of America, Vol. 101, No. 46, 2004, pp. 16385–16389, doi: 10.1073/pnas.0403723101.

[68] LiCalzi, M., Surucu, O., "The power of diversity over large solution spaces", Manage. Sci., Vol. 58, No. 7, 2012, pp. 1408–1421, doi: 10.1287/mnsc.1110.1495.

[69] Siebers, P.-O., Macal, C. M., Garnett, J., Buxton, D., Pidd, M., "Discrete-event simulation is dead, long live agent-based simulation!", J. Simul., Vol. 4, No. 3, 2010, pp. 204–210, doi: 10.1057/jos.2010.14.

[70] Brabazon, A., Silva, A., Sousa, T. F. S., Matthews, R., O'Neill, M., "Simulating product invention using InventSim", in Handbook of Research on Nature-Inspired Computing for Economics and Management, IGI Global, 2007, pp. 379–394.

[71] Sosa, R., Gero, J. S., "A computational study of creativity in design: The role of society", AIEDAM, Vol. 19, No. 4, 2005, pp. 229–244, doi: 10.1017/S089006040505016X.

[72] Zhong, X., Ozdemir, S. Z., "Structure, learning, and the speed of innovating: a two-phase model of collective innovation using agent based modeling", Ind. Corp. Change., Vol. 19, No. 5, 2010, pp. 1459–1492, doi: 10.1093/icc/dtq020.

[73] Phillips, J., "Team-RUP: An Agent-Based Simulation Study of Team Behavior In Software Development Organizations", Auburn University, 2006.

[74] Songhori, M. J., Jalali, M. S., Terano, T., "The effects of teams' initial characterizations of interactions on product development performance", in *2017 Winter Simulation Conference (WSC)*, 2017, pp. 1383–1394, doi: 10.1109/WSC.2017.8247882.

[75] Singh, V., Casakin, H., "Developing a computational framework to study the effects of use of analogy in design on team cohesion and team collaboration", in *Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 11: Human Behaviour in Design, Design Education*, 2015, pp. 101–110.

[76] Singh, V., Dong, A., Gero, J. S., "Social learning in design teams: The importance of direct and indirect communications", AIEDAM, Vol. 27, No. 2, 2013, pp. 167–182, doi: 10.1017/S0890060413000061.

[77] Dehkordi, F. M., Thompson, A., Larsson, T., "Impacts of Project-Overload on Innovation inside Organizations: Agent-Based Modeling", International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, Vol. 6, No. 11, 2012, pp. 2808–2813.

[78] Sosa, R., Gero, J. S., "Brainstorming in Solitude and Teams: The Role of Group Influence", in *Proceedings of the Third International Conference on Computational Creativity*, 2012, pp. 188–194.

[79] Heath, B., Hill, R., Ciarallo, F., "A Survey of Agent-Based Modeling Practices (January 1998 to July 2008)", Journal of Artificial Societies and Social Simulation, Vol. 12, No. 4, 2009, p. 9.

[80] Jin, Y., Levitt, R. E., "The Virtual Design Team : A Computational Model of Project Organizations", Computational & Mathematical Organization Theory, Vol. 2, No. 3, 1996, pp. 171–196, doi: 10.1007/BF00127273.

[81] Christian, A. D., "Simulation of information flow in design", Massachusetts Institute of Technology, 1995.

[82] Gonçalves, B., Lopes, A., Esteves, S., Botelho, L., "Cognitive agents based simulation for decisions regarding human team composition", in *Proceedings of AAMAS '06*, 2006, pp. 65–72, doi: 10.1145/1160633.1160641.

[83] Martínez-Miranda, J., Aldea, A., Bañares-Alcántara, R., Alvarado, M., "TEAKS: Simulation of human performance at work to support team configuration", in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 114–116, doi: 10.1145/1160633.1160649.

[84] Zhang, X., Luo, L., Yang, Y., Li, Y., Schlick, C. M., Grandt, M., "A simulation approach for evaluation and improvement of organisational planning in collaborative product development projects", International Journal of Production Research, Vol. 47, No. 13, 2009, pp. 3471–3501, doi: 10.1080/00207540802356770.

[85] Crowder, R. M., Robinson, M. A., Hughes, H. P. N., Sim, Y.-W., "The development of an agent-based modeling framework for simulating engineering team work", IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, Vol. 42, No. 6, 2012, pp. 1425–1439.

[86] Du, J., El-Gafy, M., "Virtual organizational imitation for construction enterprises

(VOICE): Managing business complexity using agent based modeling", in *Construction Research Congress 2010: Innovation for Reshaping Construction Practice - Proceedings of the 2010 Construction Research Congress*, 2010, pp. 398–408, doi: 10.1061/41109(373)40.

[87] Levitt, R. E., "The Virtual Design Team: Designing project organizations as engineers design bridges", J. Organiz. Des., Vol. 1, No. 2, 2012, pp. 14–41, doi: 10.7146/jod.6345.

[88] Jin, Y., Levitt, R. E., Christiansen, T., Kunz, J. C., "The“ Virtual Design Team”: A Computational Model of Engineering Design Teams", in *AAAI'94 Spring Symposium on Computational Organization Design*, 1994, pp. 99–105.

[89] EPM™, "SimVision® User Guide", 2005.

[90] Le, Q., Panchal, J. H., "Modeling the effect of product architecture on mass-collaborative processes", J. Comput. Inf. Sci. Eng., Vol. 11, No. 1, 2011, p. 11003, doi: 10.1115/1.3563054.

[91] Levine, S. S., Prietula, M. J., "Open collaboration for innovation: principles and performance", Organ. Sci., Vol. 25, No. 5, 2013, pp. 1414–1433, doi: 10.1287/orsc.2013.0872.

[92] Zhang, S., Li, Y., Zhang, X., "Agent Behavior-Based Simulation Study on Mass Collaborative Product Development Process", Math. Probl. Eng., Vol. 2015, 2015, doi: 10.1155/2015/689383.

[93] Zhou, H., Hu, Y., Zhang, X., Zhao, D., "Influence study of participants' characteristics on project evolution in open source community based on agent-based simulation", in *Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on*, 2016, pp. 527–531, doi: 10.1109/CSCWD.2016.7566045.

[94] Olson, J., Cagan, J., Kotovsky, K., "Unlocking organizational potential: a computational platform for investigating structural interdependence in design", J. Mech. Des., Vol. 131, No. 3, 2009, pp. 31001–31013, doi: 10.1115/1.3066501.

[95] Saoud, N. B.-B., Mark, G., "Complexity theory and collaboration: An agent-based simulator for a space mission design team", Comput. Math. Organiz. Theor., Vol. 13, No. 2, 2007, pp. 113–146, doi: 10.1007/s10588-006-9002-7.

[96] Blau, B., Hildenbrand, T., Armbruster, M., Fassunge, M. G., Xu, Y., Knapper, R., "Incentives and Performance in Large-scale Lean Software Development-An Agent-

based Simulation Approach.", in *6th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, 2011, pp. 26–37.

[97] Farhangian, M., Purvis, M., Purvis, M., Savarimuthu, T. B. R., "Agent-Based Modeling of Resource Allocation in Software Projects Based on Personality and Skill", in Advances in Social Computing and Multiagent Systems, Springer International Publishing, 2015, pp. 130–146.

[98] Li, Y., Zhang, X., Zhang, S., "Arbitration strategy analysis of resource conflict in collaborative product development process using agent-based simulation", in *Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on*, 2011, pp. 324–328, doi: 10.1109/CSCWD.2011.5960093.

[99] Xia, N., Hu, B., Jiang, F., "An exploration for knowledge evolution affected by task assignment in a research and development team: perspectives of learning obtained through practice and communication", Simulation, Vol. 92, No. 7, 2016, pp. 649–668, doi: 10.1177/0037549716655609.

[100] Du, J., El-Gafy, M., "Modeling organizational behaviors of construction enterprises: An agent based modeling approach", in *Proceedings of the 2014 Winter Simulation Conference*, 2014, pp. 3341–3352, doi: 10.1109/WSC.2014.7020168.

[101] Son, J., Rojas, E. M., Shin, S.-W., "Application of agent-based modeling and simulation to understanding complex management problems in CEM research", J. Civ. Eng. Manag., Vol. 21, No. 8, 2015, pp. 998–1013, doi: 10.3846/13923730.2014.893916.

[102] Hsu, S.-C., Weng, K.-W., Cui, Q., Rand, W., "Understanding the complexity of project team member selection through agent-based modeling", Int. J. Proj. Manag., Vol. 34, No. 1, 2016, pp. 82–93, doi: 10.1016/j.ijproman.2015.10.001.

[103] Son, J., Rojas, E. M., "Evolution of Collaboration in Temporary Project Teams: An Agent-Based Modeling and Simulation Approach", Journal of Construction Engineering and Management, Vol. 137, No. 8, 2011, pp. 619–628, doi: 10.1061/(ASCE)CO.1943-7862.0000331.

[104] Ambler, N. P., "Design in the Modern Age: Investigating the Role of Complexity in the Performance of Collaborative Engineering Design Teams", Virginia Polytechnic Institute and State University, 2015.

[105] Bott, M., Mesmer, B., "Agent-Based Simulation of Hardware-Intensive Design Teams

Using the Function–Behavior–Structure Framework", Systems, Vol. 7, No. 3, Jul. 2019, p. 37, doi: 10.3390/systems7030037.

[106] Singh, H., Cascini, G., Casakin, H., Singh, V., "A Computational Framework for Exploring the Socio-Cognitive Features of Teams and their Influence on Design Outcomes", in *Proceedings of the 22nd International Conference on Engineering Design*, 2019, doi: 10.1017/dsi.2019.3.

[107] Kauffman, S. A., The origins of order: Self-organization and selection in evolution. Oxford University Press, USA, 1993.

[108] Martynov, A., Abdelzaher, D., "Iterative problem solving in teams: insights from an agent-based simulation", Team Perform. Manag., Vol. 22, No. 1/2, 2016, pp. 2–21, doi: 10.1108/TPM-04-2015-0023.

[109] Jafari Songhori, M., Tavana, M., Terano, T., "Product development team formation: effects of organizational- and product-related factors", Computational and Mathematical Organization Theory, Nov. 2019, doi: 10.1007/s10588-019-09302-8.

[110] Mihm, J., Loch, C., Huchzermeier, A., "Problem--solving oscillations in complex engineering projects", Manage. Sci., Vol. 49, No. 6, 2003, pp. 733–750, doi: 10.1287/mnsc.49.6.733.16021.

[111] Lapp, S., Jablokow, K., McComb, C., "KABOOM: an agent-based model for simulating cognitive style in team problem solving", Design Science, Vol. 5, Aug. 2019, p. e13, doi: 10.1017/dsj.2019.12.

[112] McComb, C., Cagan, J., Kotovsky, K., "Lifting the Veil: Drawing insights about design teams from cognitively-inspired computational model", Design Studies, Vol. 40, 2015, pp. 119–142.

[113] Hulse, D., Tumer, K., Hoyle, C., Tumer, I., "Modeling multidisciplinary design with multiagent learning", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 33, No. 1, Feb. 2019, pp. 85–99, doi: 10.1017/S0890060418000161.

[114] Du, J., El-Gafy, M., "Virtual organizational imitation for construction enterprises: Agent-based simulation framework for exploring human and organizational implications in construction management", J. Comput. Civil Eng., Vol. 26, No. 3, 2012, pp. 282–297, doi: 10.1061/(ASCE)CP.1943-5487.0000122.

[115] Dutta, P., Pepe, C., Xi, H., "Analyzing Organization Structures and Performance through Agent-Based Socio-technical Modeling", in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 2015, Vol. 2, pp. 39–53, doi: 10.1007/978-3-319-13356-0_4.

[116] Zhang, X., Thomson, V., "Modelling the development of complex products using a knowledge perspective", Research in Engineering Design, Vol. 30, No. 2, Apr. 2019, pp. 203–226, doi: 10.1007/s00163-017-0274-3.

[117] Hauser, J., Clausing, D., "The house of quality", *Harvard Business Review*, pp. 63–73, May-1988.

[118] Eppinger, S. D., Browning, T. R., Design structure matrix methods and applications. MIT Press, Cambridge, 2012.

[119] Christiansen, R. T., "Modeling Efficienty and Effectiveness of Coordination in Engineering Design Teams", Stanford University, 1993.

[120] Zhang, X., Zhang, S., Li, Y., Schlick, C., "Task scheduling behaviour in agent-based product development process simulation", Int. J. Comput. Integ. M., Vol. 25, No. 10, 2012, pp. 914–923, doi: 10.1080/0951192X.2011.596223.

[121] Zhang, S., Li, Y., "Multi-agent simulation of partner selection behaviour based on matching degree in collaborative product development process", Computer Modelling and New Technologies, Vol. 18, No. 10, 2014, pp. 25–30.

[122] Herrmann, J. W., "Predicting the Performance of a Design Team Using a Markov Chain Model", IEEE T. Eng. Manage., Vol. 62, No. 4, 2015, pp. 507–516, doi: 10.1109/TEM.2015.2456833.

[123] Ramsey, M., Levitt, R. E., "A Computational Framework for Experimentation with Edge Organizations", in *Proceedings of the 10th International Command and Control Research and Technology Symposium*, 2005.

[124] MacKinnon, D. J., "How Individual Skill Growth and Decay Affect the Performance of Project Organizations", Stanford University, 2007.

[125] Faas, P., Swindler, S., Lyons, J., Levitt, R., Ramsey, M., Vincent, P., "Organizational modeling and simulation in a planning organization", 2009.

[126] Yang, Y., Song, L., Zhang, X., "Organization-oriented simulation of collaborative

product development process based on designer's agent model", in *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, 2007, pp. 309–314, doi: 10.1109/CSCWD.2007.4281453.

[127] Lapp, S., Jablokow, K., McComb, C., "Collaborating with style: Using an agent-based model to simulate cognitive style diversity in problem solving teams", in *Proceedings of the ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2019.

[128] Gero, J. S., "Design Prototypes: A Knowledge Representation Schema for Design", AI Magazine, Vol. 11, No. 4, 1990, pp. 26–36.

[129] Bott, M. J., Mesmer, B., "Determination of Function-Behavior-Structure Model Transition Probabilities from Real-World Data", in *AIAA Scitech 2019 Forum*, 2019, doi: 10.2514/6.2019-1030.

[130] Martínez-Miranda, J., Pavón, J., "Modeling the influence of trust on work team performance", Simulation, Vol. 88, No. 4, 2012, pp. 408–436, doi: 10.1177/0037549711404714.

[131] Singh, V., Dong, A., Gero, J. S., "Developing a computational model to understand the contributions of social learning modes to task coordination in teams", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 27, No. 1, 2013, pp. 3–17, doi: 10.1017/S0890060412000340.

[132] Singh, V., Gero, J. S., "Computational modeling of temporary design teams", in *Proceedings of CAADRIA07*, 2007, pp. 49–56.

[133] Kirton, M. J., Adaption-Innovation: In the context of diversity and change. Routledge, 2004.

[134] Briggs Myers, I., Myers, P., Gifts Differing: Understanding Personality Type. Davies-Black Publishing, Mountain View, CA, 1995.

[135] Belbin, R. M., Team roles at work. Routledge, 2012.

[136] Watts, D. J., "Networks, Dynamics, and the Small- World Phenomenon", American Journal of Sociology, Vol. 105, No. 2, Sep. 1999, pp. 493–527, doi: 10.1086/210318.

[137] Sim, Y.-W., Crowder, R., Robinson, M., Hughes, H., "An agent-based approach to modeling integrated product teams undertaking a design activity", in *ASME 2009*

International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2009, pp. 227–236.

[138] Crowder, R. M., Robinson, M. A., Hughes, H. P. N., Sim, Y. W., "The development of an agent-based modeling framework for simulating engineering team work", IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, Vol. 42, No. 6, 2012, pp. 1425–1439, doi: 10.1109/TSMCA.2012.2199304.

[139] Singh, V., Dong, A., Gero, J. S., "Effects of social learning and team familiarity on team performance", in *Proceedings of the 2009 Spring Simulation Multiconference*, 2009, p. 6.

[140] Salas, E., Cooke, N. J., Rosen, M. A., "On Teams, Teamwork, and Team Performance: Discoveries and Developments", Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 50, No. 3, Jun. 2008, pp. 540–547, doi: 10.1518/001872008X288457.

[141] Perišić, M. M., Štorga, M., Gero, J. S., "Building a computational laboratory for the study of team behaviour in product development", in *Proceedings of the 21st International Conference on Engineering Design (ICED 17)*, 2017, pp. 189–198.

[142] Kennedy, W. G., "Modelling Human Behaviour in Agent-Based Models", in Agent-Based Models of Geographical Systems, Springer Netherlands, Dordrecht, 2012, pp. 167–179.

[143] Balke, T., Gilbert, N., "How Do Agents Make Decisions? A Survey", Journal of Artificial Societies and Social Simulation, Vol. 17, No. 4, 2014, doi: 10.18564/jasss.2687.

[144] Rao, A. S., Georgeff, M. P., "BDI Agents: From Theory and Practice", in *Proceedings of the First International Conference on Multiagent Systems - ICMAS-95*, 1995, pp. 312–319.

[145] Pereira, D., Oliveira, E., Moreira, N., Sarmento, L., "Towards an Architecture for Emotional BDI Agents", in *2005 Purtuguese Conference on Artificial Intelligence*, 2005, pp. 40–46, doi: 10.1109/EPIA.2005.341262.

[146] Jiang, H., Vidal, J. M., Huhns, M. N., "EBDI: an architecture for emotional agents", in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*, 2007, p. 1, doi: 10.1145/1329125.1329139.

[147] Sánchez, Y., Coma, T., Aguelo, A., Cerezo, E., "ABC-EBDI: An affective framework for BDI agents", Cognitive Systems Research, Vol. 58, Dec. 2019, pp. 195–216, doi: 10.1016/j.cogsys.2019.07.002.

[148] Schmidt, B., The Modelling of Human Behaviour. SCS Publications, Erlangen, 2000.

[149] Elkosantini, S., "Toward a new generic behavior model for human centered system simulation", Simulat. Modell. Pract. Theory, Vol. 52, 2015, pp. 108–122, doi: 10.1016/j.simpat.2014.12.007.

[150] Carley, K. M., Newell, A., "The nature of the social agent", J. Math. Sociol., Vol. 19, No. 4, 1994, pp. 221–262, doi: 10.1080/0022250X.1994.9990145.

[151] Lazarus, R. S., "Progress on a cognitive-motivational-relational theory of emotion.", American Psychologist, Vol. 46, No. 8, 1991, pp. 819–834, doi: 10.1037/0003-066X.46.8.819.

[152] Peng, W., Gero, J. S., "Situated concept formation from interactions: An implementable constructive memory model", 2015.

[153] Peng, W., Gero, J. S., "Using a Constructive Interactive Activation and Competition Neural Network to Construct a Situated Agent’s Experience", in Lecture Notes in Computer Science PRICAI 2006: Trends in Artificial Intelligence, Springer-Verlag, Berlin/Heidelberg, 2006, pp. 21–30.

[154] Gero, J. S., Peng, W., "Understanding behaviors of a constructive memory agent: A markov chain analysis", Knowledge-Based Systems, Vol. 22, No. 8, Dec. 2009, pp. 610–621, doi: 10.1016/j.knosys.2009.05.006.

[155] Liew, P.-S., Gero, J. S., "An implementation model of constructive memory for a design agent", in Agents in Design, Gero, J. S., Brazier, F. M. T., (eds.) University of Sydney, 2002, pp. 257–276.

[156] Gero, J. S., Smith, G. J., "A computational framework for concept formation for a situated design agent: Gero and Fujii revisited", 2009.

[157] Gero, J. S., Fujii, H., "A computational framework for concept formation for a situated design agent", Knowledge-Based Systems, Vol. 13, No. 6, Nov. 2000, pp. 361–368, doi: 10.1016/S0950-7051(00)00076-9.

[158] Gero, J. S., "Conceptual designing as a sequence of situated acts", in Artificial

Intelligence in Structural Engineering, Springer, Berlin, 1998, pp. 165–177.

[159] Lindblom, J., Ziemke, T., "Social Situatedness of Natural and Artificial Intelligence: Vygotsky and Beyond", Adaptive Behavior, Vol. 11, No. 2, Jun. 2003, pp. 79–96, doi: 10.1177/10597123030112002.

[160] Bartlett, F. C., Remembering - A study in experimental and social psychology. Cambridge University Press, London, 1977.

[161] Riegler, A., "Constructive memory", Kybernetes, Vol. 34, No. 1/2, 2005, pp. 89–104.

[162] Schacter, D. L., "Adaptive constructive processes and the future of memory.", American Psychologist, Vol. 67, No. 8, 2012, pp. 603–613, doi: 10.1037/a0029869.

[163] Gero, J. S., Smith, G. J., "A computational framework for concept formation for a situated design agent: Constructive memory", 2009.

[164] Thomas, R., Gero, J., "Social Interactions in Post-Design Phases in Product Development and Consumption: Computational Social Science Modeling", in Principia Designae — Pre-Design, Design, and Post-Design, Taura, T., (ed.) Springer Japan, Tokyo, 2015, pp. 147–173.

[165] Burke, C. S., Stagl, K. C., Salas, E., Pierce, L., Kendall, D., "Understanding team adaptation: a conceptual analysis and model.", The Journal of Applied Psychology, Vol. 91, No. 6, 2006, pp. 1189–207, doi: 10.1037/0021-9010.91.6.1189.

[166] Edmondson, A., "Psychological Safety and Learning Behavior in Work Teams", Administrative Science Quarterly, Vol. 44, No. 2, Jun. 1999, p. 350, doi: 10.2307/2666999.

[167] Kozlowski, S. W. J., Bell, B. S., "Team learning, development, and adaptation", in Work group learning: Understanding, improving and assessing how groups learn in organizations, Sessa, V. I., London, M., (eds.) Taylor & Francis Group, 2008, pp. 15–44.

[168] Endsley, M. R., "Toward a Theory of Situation Awareness in Dynamic Systems", Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 37, No. 1, 1995, pp. 32–64, doi: 10.1518/001872095779049543.

[169] Kotseruba, I., Tsotsos, J. K., "40 years of cognitive architectures: core cognitive abilities and practical applications", Artificial Intelligence Review, Jul. 2018, doi:

10.1007/s10462-018-9646-y.

[170] Anderson, J. R., How Can the Human Mind Occur in the Physical Universe? Oxford University Press, 2007.

[171] Laird, J. E., The Soar Cognitive Architecture. The MIT Press, 2012.

[172] Rosenbloom, P. S., Demski, A., Ustun, V., "The Sigma Cognitive Architecture and System: Towards Functionally Elegant Grand Unification", Journal of Artificial General Intelligence, Vol. 7, No. 1, Dec. 2016, pp. 1–103, doi: 10.1515/jagi-2016-0001.

[173] Laird, J. E., Lebiere, C., Rosenbloom, P. S., "A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics", AI Magazine, Vol. 38, No. 4, Dec. 2017, pp. 13–26, doi: 10.1609/aimag.v38i4.2744.

[174] Sun, R., Anatomy of the Mind. Oxford University Press, 2016.

[175] Miyake, A., Shah, P., "Toward Unified Theories of Working Memory: Emerging General Consensus, Unresolved Theoretical Issues, and Future Research Directions", in Models of Working Memory, Miyake, A., Shah, P., (eds.) Cambridge University Press, Cambridge, 1999, pp. 442–482.

[176] Kintsch, W., Healy, A. F., Hegarty, M., Pennington, B. F., Salthouse, T. A., "Models of Working Memory: Eight Questions and Some General Issues", in Models of Working Memory, Miyake, A., Shah, P., (eds.) Cambridge University Press, Cambridge, 1999, pp. 412–441.

[177] Lovett, M. C., Reder, L. M., Lebiere, C., "Modeling Working Memory in a Unified Architecture: An ACT-R Perspective", in Models of Working Memory, Miyake, A., Shah, P., (eds.) Cambridge University Press, Cambridge, 1999, pp. 135–182.

[178] Cowan, N., "An Embedded-Processes Model of Working Memory", in Models of Working Memory, Miyake, A., Shah, P., (eds.) Cambridge University Press, Cambridge, 1999, pp. 62–101.

[179] Miyake, A., Shah, P., Models of Working Memory. Cambridge University Press, Cambridge, 1999.

[180] Wilhelm, O., Hildebrandt, A., Oberauer, K., "What is working memory capacity, and how can we measure it?", Frontiers in Psychology, Vol. 4, No. JUL, 2013, doi:

10.3389/fpsyg.2013.00433.

[181] Ma, W. J., Husain, M., Bays, P. M., "Changing concepts of working memory", Nature Neuroscience, Vol. 17, No. 3, Mar. 2014, pp. 347–356, doi: 10.1038/nn.3655.

[182] Just, M. A., Carpenter, P. A., "A Capacity Theory of Comprehension : Individual Differences in Working Memory", Psychological Review, Vol. 99, No. 1, 1992, pp. 122–149, doi: 10.1037/0033-295x.99.1.122.

[183] Cowan, N., "The Magical Mystery Four", Current Directions in Psychological Science, Vol. 19, No. 1, Feb. 2010, pp. 51–57, doi: 10.1177/0963721409359277.

[184] Miller, G. A., "The magical number seven, plus or minus two: some limits on our capacity for processing information.", Psychological Review, Vol. 63, No. 2, 1956, pp. 81–97, doi: 10.1037/h0043158.

[185] Chase, W. G., Simon, H. A., "Perception in chess", Cognitive Psychology, Vol. 4, No. 1, Jan. 1973, pp. 55–81, doi: 10.1016/0010-0285(73)90004-2.

[186] Gobet, F., Clarkson, G., "Chunks in expert memory: Evidence for the magical number four … or is it two?", Memory, Vol. 12, No. 6, Nov. 2004, pp. 732–747, doi: 10.1080/09658210344000530.

[187] Anderson, J. R., "A spreading activation theory of memory", Journal of Verbal Learning and Verbal Behavior, Vol. 22, No. 3, Jun. 1983, pp. 261–295, doi: 10.1016/S0022-5371(83)90201-3.

[188] Jones, S. J., Wandzel, A. R., Laird, J. E., "Efficient Computation of Spreading Activation Using Lazy Evaluation", in *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016)*, 2016, pp. 182–187.

[189] Thomson, R., Lebiere, C., "A Balanced Hebbian Algorithm for Associative Learning in ACT-R", in *Proceedings of the 12th International Conference on Cognitive Modeling (ICCM 2013)*, 2013, pp. 71–76.

[190] Hebb, D. O., The Organization of Behavior: A Neuropsychological Theory. Wiley, Oxford, England, 1949.

[191] Tsotsos, J. K., A Computational Perspective on Visual Attention. The MIT Press, 2011.

[192] Anderson, J. R., "Human Symbol Manipulation Within an Integrated Cognitive Architecture", Cognitive Science, Vol. 29, No. 3, May 2005, pp. 313–341, doi:

10.1207/s15516709cog0000_22.

[193] Byrne, M. D., Kirlik, A., "Using Computational Cognitive Modeling to Diagnose Possible Sources of Aviation Error", The International Journal of Aviation Psychology, Vol. 15, No. 2, Apr. 2005, pp. 135–155, doi: 10.1207/s15327108ijap1502_2.

[194] Kahneman, D., Thinking, Fast and Slow. Farrar, Straus and Giroux, 2011.

[195] Kannengiesser, U., Gero, J. S., "Design thinking, fast and slow: A framework for Kahneman's dual-system theory in design", Design Science, Vol. 5, May 2019, p. e10, doi: 10.1017/dsj.2019.9.

[196] Maher, M. L., Gero, J. S., "Agent models of 3D virtual worlds", in *Proceedings of ACADIA 2002: Thresholds*, 2002, pp. 127–138.

[197] Norman, D. A., Ortony, A., Russell, D. M., "Affect and machine design: Lessons for the development of autonomous machines", IBM Systems Journal, Vol. 42, No. 1, 2003, pp. 38–44, doi: 10.1147/sj.421.0038.

[198] Štorga, M., Andreasen, M. M., Marjanović, D., "The design ontology: foundation for the design knowledge exchange and management", Journal of Engineering Design, Vol. 21, No. 4, Aug. 2010, pp. 427–454, doi: 10.1080/09544820802322557.

[199] Gero, J. S., Kannengiesser, U., "A function–behavior–structure ontology of processes", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 21, No. 4, Nov. 2007, pp. 379–391, doi: 10.1017/S0890060407000340.

[200] Gero, J. S., Kannengiesser, U., "The Function-Behaviour-Structure Ontology of Design", in An Anthology of Theories and Models of Design, Springer London, London, 2014, pp. 263–283.

[201] Jiang, H., "Understanding Senior Design Students' Product Conceptual Design Activities: A Comparison Between Industrial and Engineering Design Students", National University of Singapore, 2012.

[202] Stempfle, J., Badke-Schaub, P., "Thinking in design teams - an analysis of team communication", Design Studies, Vol. 23, No. 5, Sep. 2002, pp. 473–496, doi: 10.1016/S0142-694X(02)00004-2.

[203] Norman, D., "Emotion & design: attractive things work better", interactions, Vol. 9, No. 4, Jul. 2002, pp. 36–42, doi: 10.1145/543434.543435.

[204] Seif El-Nasr, M., Skubic, M., "A fuzzy emotional agent for decision-making in a mobile robot", in *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228)*, 1998, Vol. 1, pp. 135–140, doi: 10.1109/FUZZY.1998.687472.

[205] Fum, D., Stocco, A., "Memory, Emotion, and Rationality : An ACT-R interpretation for Gambling Task results", in *Proceedings of the Sixth International Conference on Cognitive Modeling*, 2000, pp. 106–111.

[206] Marinier, R. P., Laird, J. E., "Emotion-Driven Reinforcement Learning", Proceedings of the Annual Meeting of the Cognitive Science, Vol. 30, No. 30, 2008, pp. 115–120.

[207] Rodríguez, L.-F., Ramos, F., "Development of Computational Models of Emotions for Autonomous Agents: A Review", Cognitive Computation, Vol. 6, No. 3, Sep. 2014, pp. 351–375, doi: 10.1007/s12559-013-9244-x.

[208] Kowalczuk, Z., Czubenko, M., "Computational Approaches to Modeling Artificial Emotion – An Overview of the Proposed Solutions", Frontiers in Robotics and AI, Vol. 3, Apr. 2016, doi: 10.3389/frobt.2016.00021.

[209] Colby, B. N., Ortony, A., Clore, G. L., Collins, A., "The Cognitive Structure of Emotions", Contemporary Sociology, Vol. 18, No. 6, Nov. 1989, pp. 957–958, doi: 10.2307/2074241.

[210] Becker-Asano, C., Wachsmuth, I., "Affective computing with primary and secondary emotions in a virtual human", Autonomous Agents and Multi-Agent Systems, Vol. 20, No. 1, Jan. 2010, pp. 32–49, doi: 10.1007/s10458-009-9094-9.

[211] Ekman, P., "Basic Emotions", in Handbook of Cognition and Emotion, John Wiley & Sons, Ltd, Chichester, UK, 2005, pp. 45–60.

[212] Russell, J. A., "A circumplex model of affect", Journal of Personality and Social Psychology, Vol. 39, No. 6, 1980, pp. 1161–1178, doi: 10.1037/h0077714.

[213] Marsella, S., Gratch, J., "Computationally modeling human emotion", Communications of the ACM, Vol. 57, No. 12, Nov. 2014, pp. 56–67, doi: 10.1145/2631912.

[214] Hutchinson, A., Tracey, M. W., "Designers' own emotions and the practice of designing: a literature review and preliminary research agenda", Journal of Design Research, Vol. 15, No. 3/4, 2017, pp. 197–213, doi: 10.1504/JDR.2017.10011026.

[215] Hutchinson, A., "Designers, Emotions, And Ideas: How Graphic Designers Understand Their Emotional Experiences Around Ideation", Wayne State University, 2018.

[216] Sas, C., Zhang, C., "Investigating emotions in creative design", in *Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design - DESIRE ’10*, 2010, pp. 138–149.

[217] Chulvi, V., González-Cruz, M. C., "The influence of design methodology on a designer’s emotional parameters and on design results", DYNA, Vol. 83, No. 196, Apr. 2016, pp. 106–112, doi: 10.15446/dyna.v83n196.49783.

[218] Jeronimus, B. F., Laceulle, O. M., "Frustration", in Encyclopedia of Personality and Individual Differences, Springer International Publishing, Cham, 2017, pp. 1–5.

[219] Isen, A. M., "An Influence of Positive Affect on Decision Making in Complex Situations: Theoretical Issues With Practical Implications", Journal of Consumer Psychology, Vol. 11, No. 2, Jan. 2001, pp. 75–85, doi: 10.1207/S15327663JCP1102_01.

[220] Kounios, J., Beeman, M., "The Aha! Moment", Current Directions in Psychological Science, Vol. 18, No. 4, Aug. 2009, pp. 210–216, doi: 10.1111/j.1467-8721.2009.01638.x.

[221] McComb, C., Cagan, J., Kotovsky, K., "Studying Human Design Teams via Computational Teams of Simulated Annealing Agents", in *Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 7: 27th International Conference on Design Theory and Methodology*, 2015, doi: 10.1115/DETC2015-46545.

[222] Hulse, D., Gigous, B., Tumer, K., Hoyle, C., Tumer, I. Y., "Towards a Distributed Multiagent Learning-Based Design Optimization Method", in *Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2A: 43rd Design Automation Conference*, 2017, doi: 10.1115/DETC2017-68042.

[223] Gersick, C. J. G., "Time an transition in work teams: toward a new model of group development", Academy of Management Journal, Vol. 31, No. 1, Mar. 1988, pp. 9–41, doi: 10.2307/256496.

[224] Ones, D. S., Dilchert, S., Viswesvaran, C., "Cognitive Abilities", in The Oxford Handbook of Personnel Assessment and Selection, Schmitt, N., (ed.) Oxford University

Press, New York, NY, USA, 2012.

[225] Gottfredson, L. S., "Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography", Intelligence, Vol. 24, No. 1, Jan. 1997, pp. 13–23, doi: 10.1016/S0160-2896(97)90011-8.

[226] Ispas, D., Borman, W. C., "Personnel Selection, Psychology of", in International Encyclopedia of the Social & Behavioral Sciences, Second Edi., Vol. 17, Elsevier, 2015, pp. 936–940.

[227] Bowling, A. C., Mackenzie, B. D., "The relationship between speed of information processing and cognitive ability", Personality and Individual Differences, Vol. 20, No. 6, Jun. 1996, pp. 775–800, doi: 10.1016/0191-8869(96)00022-0.

[228] Gill, D., Prowse, V., "Cognitive Ability, Character Skills, and Learning to Play Equilibrium: A Level- k Analysis", Journal of Political Economy, Vol. 124, No. 6, Dec. 2016, pp. 1619–1676, doi: 10.1086/688849.

[229] Engle, R. W., "Working Memory Capacity as Executive Attention", Current Directions in Psychological Science, Vol. 11, No. 1, Feb. 2002, pp. 19–23, doi: 10.1111/1467-8721.00160.

[230] Rammstedt, B., Danner, D., Martin, S., "The association between personality and cognitive ability: Going beyond simple effects", Journal of Research in Personality, Vol. 62, 2016, pp. 39–44, doi: 10.1016/j.jrp.2016.03.005.

[231] McCrae, R. R., Costa Jr., P. T., "A five-factor theory of personality", in Handbook of personality: Theory and research, 2nd ed., Pervin, L. A., John, O. P., (eds.) Guilford, New York, NY, 1999, pp. 139–153.

[232] Barrick, M. R., Mount, M. K., "Yes, Personality Matters: Moving on to More Important Matters", Human Performance, Vol. 18, No. 4, Oct. 2005, pp. 359–372, doi: 10.1207/s15327043hup1804_3.

[233] Barrick, M. R., Stewart, G. L., Neubert, M. J., Mount, M. K., "Relating member ability and personality to work-team processes and team effectiveness.", Journal of Applied Psychology, Vol. 83, No. 3, 1998, pp. 377–391, doi: 10.1037/0021-9010.83.3.377.

[234] Kichuk, S. L., Wiesner, W. H., "The big five personality factors and team performance: implications for selecting successful product design teams", Journal of Engineering and Technology Management, Vol. 14, No. 3–4, Sep. 1997, pp. 195–221, doi:

10.1016/S0923-4748(97)00010-6.

[235] Sprehn, K. A., Macht, G. A., Kremer, G. E., Nembhard, D. A., "Mediating engineering design team performance through conscientiousness and cognitive style", in *Proceedings of the 19th International Conference on Engineering Design (ICED13)*, 2013, No. Vol.7: Human Behaviour in Design, pp. 179–188.

[236] Chang, C.-C., Peng, L.-P., Lin, J.-S., Liang, C., "Predicting the creativity of design majors based on the interaction of diverse personality traits", Innovations in Education and Teaching International, Vol. 52, No. 4, Jul. 2015, pp. 371–382, doi: 10.1080/14703297.2014.999697.

[237] Baer, M., Oldham, G. R., Jacobsohn, G. C., Hollingshead, A. B., "The Personality Composition of Teams and Creativity: The Moderating Role of Team Creative Confidence", The Journal of Creative Behavior, Vol. 42, No. 4, Dec. 2008, pp. 255–282, doi: 10.1002/j.2162-6057.2008.tb01299.x.

[238] Feist, G. J., "A Meta-Analysis of Personality in Scientific and Artistic Creativity", Personality and Social Psychology Review, Vol. 2, No. 4, Nov. 1998, pp. 290–309, doi: 10.1207/s15327957pspr0204_5.

[239] Schmitt, D. P., Allik, J., McCrae, R. R., Benet-Martínez, V., "The Geographic Distribution of Big Five Personality Traits: Patterns and Profiles of Human Self-Description Across 56 Nations", Journal of Cross-Cultural Psychology, Vol. 38, No. 2, Mar. 2007, pp. 173–212, doi: 10.1177/0022022106297299.

[240] Tomasello, M., The Cultural Origins of Human Cognition. Harvard University Press, 2009.

[241] Carley, K. M., Filonuk, D. T., Joseph, K., Kowalchuck, M., Lanham, M. J., Morgan, G. P., "Construct User Guide", Nov. 2012.

[242] Wegner, D. M., "Transactive Memory: A Contemporary Analysis of the Group Mind", in Theories of Group Behavior, Springer New York, New York, NY, 1987, pp. 185–208.

[243] Ren, Y., Argote, L., "Transactive Memory Systems 1985–2010: An Integrative Framework of Key Dimensions, Antecedents, and Consequences", Academy of Management Annals, Vol. 5, No. 1, Jun. 2011, pp. 189–229, doi: 10.5465/19416520.2011.590300.

[244] Miller, K. D., Pentland, B. T., Choi, S., "Dynamics of Performing and Remembering

Organizational Routines", Journal of Management Studies, Vol. 49, No. 8, Dec. 2012, pp. 1536–1558, doi: 10.1111/j.1467-6486.2012.01062.x.

[245] Robertson, P. J., "Transactive Memory Systems and Group Performance: A Simulation Study of the Role of Shared Cognition", Academy of Management Proceedings, Vol. 2008, No. 1, Aug. 2008, pp. 1–6, doi: 10.5465/ambpp.2008.33718594.

[246] Ren, Y., Carley, K. M., Argote, L., "The contingent effects of transactive memory: When is it more beneficial to know what others know?", Manage. Sci., Vol. 52, No. 5, 2006, pp. 671–682, doi: 10.1287/mnsc.1050.0496.

[247] Wijngaards, N. J. E., Boonstra, H. M., Brazier, F. M. T., "The role of trust in distributed design", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 18, No. 2, May 2004, pp. 199–209, doi: 10.1017/S0890060404040132.

[248] Mayer, R. C., Davis, J. H., Schoorman, F. D., "An Integrative Model of Organizational Trust", The Academy of Management Review, Vol. 20, No. 3, Jul. 1995, p. 709, doi: 10.2307/258792.

[249] Webber, S. S., "Development of Cognitive and Affective Trust in Teams", Small Group Research, Vol. 39, No. 6, Dec. 2008, pp. 746–769, doi: 10.1177/1046496408323569.

[250] Marsh, S. P., "Formalising Trust as a Computational Concept", University of Stirling, 1994.

[251] Fulmer, C. A., Gelfand, M. J., "At What Level (and in Whom) We Trust", Journal of Management, Vol. 38, No. 4, Jul. 2012, pp. 1167–1230, doi: 10.1177/0149206312439327.

[252] Yakovleva, M., Reilly, R. R., Werko, R., "Why do we trust? Moving beyond individual to dyadic perceptions.", Journal of Applied Psychology, Vol. 95, No. 1, 2010, pp. 79–91, doi: 10.1037/a0017102.

[253] Sicora, R. T., "Personality and Trust: A Qualitative Study on the Personality Styles/Traits of Leaders and Emplyees, and the Impact on Culture of Trust within Organisations", University of st. Thomas, 2015.

[254] Cho, J.-H., Cam, H., Oltramari, A., "Effect of personality traits on trust and risk to phishing vulnerability: Modeling and analysis", in *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2016, pp. 7–13, doi: 10.1109/COGSIMA.2016.7497779.

[255] Kanawattanachai, P., Yoo, Y., "The Impact of Knowledge Coordination on Virtual Team Performance over Time", MIS Quarterly, Vol. 31, No. 4, 2007, p. 783, doi: 10.2307/25148820.

[256] Akgün, A. E., Byrne, J., Keskin, H., Lynn, G. S., Imamoglu, S. Z., "Knowledge networks in new product development projects: A transactive memory perspective", Information & Management, Vol. 42, No. 8, Dec. 2005, pp. 1105–1120, doi: 10.1016/j.im.2005.01.001.

[257] Rau, D., "The Influence of Relationship Conflict and Trust on the Transactive Memory", Small Group Research, Vol. 36, No. 6, Dec. 2005, pp. 746–771, doi: 10.1177/1046496405281776.

[258] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G., "MASON: A Multiagent Simulation Environment", Simulation, Vol. 81, No. 7, Jul. 2005, pp. 517–527, doi: 10.1177/0037549705058073.

[259] Bharathy, G. K., Silverman, B., "Validating agent based social systems models", in *Proceedings of the 2010 Winter Simulation Conference*, 2010, No. 2002, pp. 441–453, doi: 10.1109/WSC.2010.5679142.

[260] Gero, J. S., Kannengiesser, U., "The situated function–behaviour–structure framework", Design Studies, Vol. 25, No. 4, Jul. 2004, pp. 373–391, doi: 10.1016/j.destud.2003.10.010.

[261] Perišić, M. M., Štorga, M., Gero, J. S., "Exploring the Effect of Experience on Team Behavior: A Computational Approach", in Design Computing and Cognition ’18, Springer International Publishing, Cham, 2019, pp. 595–612.

[262] Umeda, Y., Tomiyama, T., "FBS Modeling: Modeling Scheme of Function for Conceptual Design", in *Proceedings of the 9th international workshop on qualitative reasoning*, 1995, pp. 271–278.

[263] Sosa, M. E., Eppinger, S. D., Rowles, C. M., "A Network Approach to Define Modularity of Components in Complex Products", Journal of Mechanical Design, Vol. 129, No. 11, Nov. 2007, pp. 1118–1129, doi: 10.1115/1.2771182.

[264] Wang, M., Chen, W., Huang, Y., Contractor, N. S., Fu, Y., "Modeling customer preferences using multidimensional network analysis in engineering design", Design Science, Vol. 2, Nov. 2016, p. e11, doi: 10.1017/dsj.2016.11.

[265] O'Reilly, R. C., "Biologically Based Computational Models of High-Level Cognition", Science, Vol. 314, No. 5796, Oct. 2006, pp. 91–94, doi: 10.1126/science.1127242.

[266] Leibowitz, N., Baum, B., Enden, G., Karniel, A., "The exponential learning equation as a function of successful trials results in sigmoid performance", Journal of Mathematical Psychology, Vol. 54, No. 3, Jun. 2010, pp. 338–340, doi: 10.1016/j.jmp.2010.01.006.

[267] Kannengiesser, U., Gero, J. S., "Empirical Evidence for Kahneman's System 1 and System 2 Thinking in Design", in *Human Behaviour in Design - Proceedings of the 2nd SIG conference*, 2019, pp. 89–100.

[268] Moss, J., Cagan, J., Kotovsky, K., "Learning from design experience in an agent-based design system", Research in Engineering Design, Vol. 15, Jul. 2004, pp. 77–92, doi: 10.1007/s00163-003-0042-4.

[269] Sørensen, T. J., A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. I kommission hos E. Munksgaard, København, 1948.

[270] Dice, L. R., "Measures of the Amount of Ecologic Association Between Species", Ecology, Vol. 26, No. 3, Jul. 1945, pp. 297–302, doi: 10.2307/1932409.

[271] Levandowsky, M., Winter, D., "Distance between Sets", Nature, Vol. 234, No. 5323, Nov. 1971, pp. 34–35, doi: 10.1038/234034a0.

[272] Vijaymeena, M. K., Kavitha, K., "A Survey on Similarity Measures in Text Mining", Machine Learning and Applications: An International Journal, Vol. 3, No. 1, Mar. 2016, pp. 19–28, doi: 10.5121/mlaij.2016.3103.

[273] Goldschmidt, G., "The dialectics of sketching", Creativity Research Journal, Vol. 4, No. 2, Jan. 1991, pp. 123–143, doi: 10.1080/10400419109534381.

[274] Gero, J., Milovanovic, J., "The situated function-behavior-structure co-design model", CoDesign, Aug. 2019, pp. 1–26, doi: 10.1080/15710882.2019.1654524.

[275] Pinto, A. M., Barroso, L., "Principles of Regulated Activation Networks", in Lecture Notes in Computer Science ICCS 2014: Graph-Based Representation and Reasoning, Hernandez, N., Jäschke, R., Croitoru, M., (eds.) Springer, Cham, 2014, pp. 231–244.

[276] Pardilla-Delgado, E., Payne, J. D., "The Deese-Roediger-McDermott (DRM) Task: A Simple Cognitive Paradigm to Investigate False Memories in the Laboratory", Journal

of Visualized Experiments, No. 119, Jan. 2017, doi: 10.3791/54793.

[277] Hudlicka, E., "Beyond Cognition: Modeling Emotion in Cognitive Architectures", in *Proceedings of the International Conference on Cognitive Modeling (ICCM)*, 2004, pp. 118–123.

[278] Bourgais, M., Taillandier, P., Vercouter, L., Adam, C., "Emotion Modeling in Social Simulation: A Survey", Journal of Artificial Societies and Social Simulation, Vol. 21, No. 2, 2018, doi: 10.18564/jasss.3681.

[279] Hudlicka, E., "What are we modeling when we model emotion?", in *AAAI spring symposium: emotion, personality, and social behavior*, 2008, pp. 52–59.

[280] LePine, J. A., Colquitt, J. A., Erez, A., "Adaptability to changing task contexts: Effects of general cognitive ability, conscientiousness, and openness to experience", Personnel Psychology, Vol. 53, No. 3, 2000, pp. 563–593, doi: 10.1111/j.1744-6570.2000.tb00214.x.

[281] Saunders, R., "Curious Design Agents and Artificial Creativity: A Synthetic Approach to the Study of Creative Behaviour", University of Sydney, 2002.

[282] Driskell, J. E., Goodwin, G. F., Salas, E., O'Shea, P. G., "What makes a good team player? Personality and team effectiveness.", Group Dynamics: Theory, Research, and Practice, Vol. 10, No. 4, 2006, pp. 249–271, doi: 10.1037/1089-2699.10.4.249.

[283] Griffin, A. S., Guillette, L. M., Healy, S. D., "Cognition and personality: An analysis of an emerging field", Trends in Ecology and Evolution, Vol. 30, No. 4, 2015, pp. 207–214, doi: 10.1016/j.tree.2015.01.012.

[284] Anusic, I., Schimmack, U., Pinkus, R. T., Lockwood, P., "The nature and structure of correlations among Big Five ratings: The halo-alpha-beta model", Journal of Personality and Social Psychology, Vol. 97, No. 6, 2009, pp. 1142–1156, doi: 10.1037/a0017159.

[285] Curtis, R. G., Windsor, T. D., Soubelet, A., "The relationship between Big-5 personality traits and cognitive ability in older adults – a review", Aging, Neuropsychology, and Cognition, Vol. 22, No. 1, 2015, pp. 42–71, doi: 10.1080/13825585.2014.888392.

[286] Lippa, R. A., "Sex Differences in Personality Traits and Gender-Related Occupational Preferences across 53 Nations: Testing Evolutionary and Social-Environmental Theories", Archives of Sexual Behavior, Vol. 39, No. 3, Jun. 2010, pp. 619–636, doi: 10.1007/s10508-008-9380-7.

[287] Cobb-Clark, D. A., Schurer, S., "The stability of big-five personality traits", Economics Letters, Vol. 115, No. 1, Apr. 2012, pp. 11–15, doi: 10.1016/j.econlet.2011.11.015.

[288] Read, S. J., Smith, B. J., Droutman, V., Miller, L. C., "Virtual personalities: Using computational modeling to understand within-person variability", Journal of Research in Personality, Vol. 69, Aug. 2017, pp. 237–249, doi: 10.1016/j.jrp.2016.10.005.

[289] Hudlicka, E., "This time with feeling: Integrated model of trait and state effects on cognition and behavior", Applied Artificial Intelligence, Vol. 16, No. 7–8, Aug. 2002, pp. 611–641, doi: 10.1080/08339510290030417.

[290] Gebhard, P., "ALMA: a layered model of affect", in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05*, 2005, pp. 29–36, doi: 10.1145/1082473.1082478.

[291] Sense, F., Behrens, F., Meijer, R. R., van Rijn, H., "An Individual's Rate of Forgetting Is Stable Over Time but Differs Across Materials", Topics in Cognitive Science, Vol. 8, No. 1, Jan. 2016, pp. 305–321, doi: 10.1111/tops.12183.

[292] Zeffane, R., A Tipu, S., Ryan, J. C., "Communication, Commitment and Trust: Exploring the Triad", International Journal of Business and Management, Vol. 6, No. 6, Jun. 2011, doi: 10.5539/ijbm.v6n6p77.

[293] Lusher, D., Kremer, P., Robins, G., "Cooperative and Competitive Structures of Trust Relations in Teams", Small Group Research, Vol. 45, No. 1, Feb. 2014, pp. 3–36, doi: 10.1177/1046496413510362.

[294] Lewicki, R. J., McAllister, D. J., Bies, R. J., "Trust and Distrust: New Relationships and Realities", The Academy of Management Review, Vol. 23, No. 3, Jul. 1998, p. 438, doi: 10.2307/259288.

[295] Ellwardt, L., Labianca, G. (Joe), Wittek, R., "Who are the objects of positive and negative gossip at work?", Social Networks, Vol. 34, No. 2, May 2012, pp. 193–205, doi: 10.1016/j.socnet.2011.11.003.

[296] Guenter, H., van Emmerik, H., Schreurs, B., Kuypers, T., van Iterson, A., Notelaers, G., "When Task Conflict Becomes Personal", Small Group Research, Vol. 47, No. 5, Oct. 2016, pp. 569–604, doi: 10.1177/1046496416667816.

[297] Watkins, C. J. C. H., "Learning from Delayed Rewards", King's College, Cambridge, UK, 1989.

[298] Osborn, A. F., Applied Imagination - Principles and Procedures of Creative Thinking, 3rd Revise. Charles Scribner’s Sons, 1979.

[299] Wasserman, S., Faust, K., Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, 1994.

[300] White, S., O’Madadhain, J., "JUNG - Class DistanceStatistics", 2009. [Online]. Available: http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/shortestpath/DistanceStatistics.html#diameter(edu.uci.ics.jung.graph.Hypergraph).

[301] Freeman, L. C., "Centrality in social networks conceptual clarification", Social Networks, Vol. 1, No. 3, Jan. 1978, pp. 215–239, doi: 10.1016/0378-8733(78)90021-7.

[302] Reminga, J., Carley, K. M., "Measures for ORA (Organizational Risk Analyzer)", 2003.

[303] O’Madadhain, J., "JUNG - Class ClosenessCentrality", 2009. [Online]. Available: http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/scoring/ClosenessCentrality.html.

[304] White, S., "JUNG - Class WeakComponentClusterer", 2009. [Online]. Available: http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/cluster/WeakComponentClusterer.html.

[305] White, S., Nelson, T., "JUNG - Class BetweennessCentrality", 2009. [Online]. Available: http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/importance/BetweennessCentrality.html.

[306] The JUNG Framework Development Team, "JUNG - Clustering coefficient metric", 2009. [Online]. Available: http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/metrics/Metrics.html.

[307] Mones, E., Vicsek, L., Vicsek, T., "Hierarchy Measure for Complex Networks", PLoS ONE, Vol. 7, No. 3, Mar. 2012, p. e33799, doi: 10.1371/journal.pone.0033799.

[308] White, S., "JUNG - Class UnweightedShortestPath", 2009.

[309] McComb, C., Cagan, J., Kotovsky, K., "Rolling with the punches: An examination of team performance in a design task subject to drastic changes", Design Studies, Vol. 36, 2015, pp. 99–121, doi: 10.1016/j.destud.2014.10.001.

[310] Wynn, D. C., Eckert, C. M., "Perspectives on iteration in design and development", Research in Engineering Design, Vol. 28, No. 2, Apr. 2017, pp. 153–184, doi: 10.1007/s00163-016-0226-3.

[311] Wilensky, U., Rand, W., "Making Models Match: Replicating an Agent-Based Model", Journal of Artificial Societies and Social Simulation, Vol. 10, No. 4, 2007, p. 2.

[312] Balci, O., "Verification, validation, and testing", in Handbook of simulation, John Wiley and Sons, 1998, pp. 335--393.

[313] Klügl, F., "A validation methodology for agent-based simulations", in *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, 2008, pp. 39–43, doi: 10.1145/1363686.1363696.

[314] Carley, K. M., "Validating Computational Models", 2017.

[315] Sterman, J. D., Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill, 2000.

[316] Sargent, R. G., "Some approaches and paradigms for verifying and validating simulation models", in *Proceeding of the 2001 Winter Simulation Conference*, 2001, pp. 106–114, doi: 10.1109/WSC.2001.977251.

[317] Sargent, R. G., "Verification and validation of simulation models", in *2008 Winter Simulation Conference,* 2008, pp. 157–169, doi: 10.1109/WSC.2008.4736065.

[318] Ritter, F. E., Schooler, L. J., "Learning Curve, The", in International Encyclopedia of the Social & Behavioral Sciences, Elsevier, 2001, pp. 8602–8605.

[319] Schmidt, F. L., Hunter, J. E., "The validity and utility of selection methods in personnel psychology: Practical and theoretical implications of 85 years of research findings.", Psychological Bulletin, Vol. 124, No. 2, 1998, pp. 262–274, doi: 10.1037/0033-2909.124.2.262.

[320] Milovanovic, J., Gero, J. S., "Exploration of cognitive design behaviour during design critiques", in *Proceedings of the 15th International Design Conference (DESIGN 2018),* 2018, pp. 2099–2110, doi: 10.21278/idc.2018.0547.

[321] Ellis, A. P. J., Hollenbeck, J. R., Ilgen, D. R., Porter, C., West, B., Moon, H., "Capacity, collaboration and commonality: a framework for understanding team learning", in *Proceedings of the 6th International command and control research and technology*

symposium, 2001, doi: 10.1.1.297.1942.

[322] Barry, B., Stewart, G. L., "Composition, process, and performance in self-managed groups: The role of personality.", Journal of Applied Psychology, Vol. 82, No. 1, 1997, pp. 62–78, doi: 10.1037/0021-9010.82.1.62.

[323] Grant, A. M., "Rethinking the Extraverted Sales Ideal", Psychological Science, Vol. 24, No. 6, Jun. 2013, pp. 1024–1030, doi: 10.1177/0956797612463706.

[324] Ahmed, S., Wallace, K. M., Blessing, L. T., "Understanding the differences between how novice and experienced designers approach design tasks", Research in Engineering Design, Vol. 14, No. 1, Feb. 2003, pp. 1–11, doi: 10.1007/s00163-002-0023-z.

[325] Perisic, M. M., Martinec, T., Storga, M., Gero, J. S., "A Computational Study of the Effect of Experience on Problem/Solution Space Exploration in Teams", Proceedings of the Design Society: International Conference on Engineering Design, Vol. 1, No. 1, Jul. 2019, pp. 11–20, doi: 10.1017/dsi.2019.4.

[326] Hurst, A., Nespoli, O. G., Abdellahi, S., Gero, J. S., "A Comparison of Design Activity of Academics and Practitioners Using the FBS Ontology: A Case Study", Proceedings of the Design Society: International Conference on Engineering Design, Vol. 1, No. 1, Jul. 2019, pp. 1323–1332, doi: 10.1017/dsi.2019.138.

[327] Sun, G., Yao, S., Carretero, J. A., "An experimental approach to understanding design problem structuring strategies", J. of Design Research, Vol. 14, No. 1, 2016, p. 94, doi: 10.1504/JDR.2016.074778.

[328] Cross, N., "Design cognition: results from protocol and other empirical studies of design activity", in Design Knowing and Learning: Cognition in Design Education, Newstatter, W., McCracken, M., (eds.) Elsevier, Oxford, UK, 2001, pp. 79–103.

[329] Björklund, T. A., "Initial mental representations of design problems: Differences between experts and novices", Design Studies, Vol. 34, No. 2, Mar. 2013, pp. 135–160, doi: 10.1016/j.destud.2012.08.005.

[330] Cross, N., "Expertise in design: an overview", Design Studies, Vol. 25, No. 5, Sep. 2004, pp. 427–441, doi: 10.1016/j.destud.2004.06.002.

[331] Lloyd, P., Scott, P., "Discovering the design problem", Design Studies, Vol. 15, No. 2, Apr. 1994, pp. 125–140, doi: 10.1016/0142-694X(94)90020-5.

[332] Liikkanen, L. A., Perttula, M., "Exploring problem decomposition in conceptual design among novice designers", Design Studies, Vol. 30, No. 1, Jan. 2009, pp. 38–59, doi: 10.1016/j.destud.2008.07.003.

[333] Perišić, M. M., Štorga, M., Gero, J. S., "Situated novelty in computational creativity studies", in *Proceedings of the Tenth International Conference on Computational Creativity ICCC 2019*, 2019, pp. 286–290.

[334] Boden, M. A., "What Is Creativity?", in Dimensions of Creativity, The MIT Press, Cambridge, MA, 1994, pp. 75–117.

[335] Amabile, T. M., Creativity in Context - Update To The Social Psychology Of Creativity, 1st Editio. Routledge, New York, New York, USA, 1996.

[336] Runco, M. A., Creativity: Theories and Themes: Research, Development and Practice. Elsevier, Academic Press, Amsterdam/Boston, 2007.

[337] Sarkar, P., Chakrabarti, A., "Assessing design creativity", Design Studies, Vol. 32, No. 4, Jul. 2011, pp. 348–383, doi: 10.1016/j.destud.2011.01.002.

[338] Besemer, S. P., Creating products in the age of design: How to improve your new product ideas! New Forums Press, Stillwater, OK, 2006.

[339] Maher, M. L., Brady, K., Fisher, D., "Computational models of surprise in evaluating creative design", in *Proceedings of the Fourth International Conference on Computational Creativity*, 2013, pp. 147–151.

[340] Grace, K., Maher, M. Lou, Fisher, D., Brady, K., "Data-intensive evaluation of design creativity using novelty, value, and surprise", International Journal of Design Creativity and Innovation, Vol. 3, No. 3–4, Oct. 2015, pp. 125–147, doi: 10.1080/21650349.2014.943295.

[341] Maher, M. L., Fisher, D., "Using AI to evaluate creative designs", in *Proceedings of the 2nd International Conference on Design Creativity*, 2012, pp. 45–54.

[342] Marsland, S., Nehmzow, U., Shapiro, J., "A real-time novelty detector for a mobile robot", in *Proceedings of European Advanced Robotics Systems Masterclass and Conference*, 2000.

[343] Pimentel, M. A. F., Clifton, D. A., Clifton, L., Tarassenko, L., "A review of novelty detection", Signal Processing, Vol. 99, Jun. 2014, pp. 215–249, doi:

10.1016/j.sigpro.2013.12.026.

[344] Ranjan, B. S. C., Siddharth, L., Chakrabarti, A., "A systematic approach to assessing novelty, requirement satisfaction, and creativity", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 32, No. 4, Nov. 2018, pp. 390–414, doi: 10.1017/S0890060418000148.

[345] Markou, M., Singh, S., "Novelty detection: a review—part 1: statistical approaches", Signal Processing, Vol. 83, No. 12, Dec. 2003, pp. 2481–2497, doi: 10.1016/j.sigpro.2003.07.018.

[346] Maguire, R., Maguire, P., Keane, M. T., "Making sense of surprise: An investigation of the factors influencing surprise judgments.", Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol. 37, No. 1, 2011, pp. 176–186, doi: 10.1037/a0021609.

[347] Brown, D. C., "Creativity, surprise & design: an introduction and investigation", in *Proceedings of the 2nd International Conference on Design Creativity*, 2012, pp. 75–84.

[348] Kelly, N., Gero, J. S., "Generate and situated transformation as a paradigm for models of computational creativity", International Journal of Design Creativity and Innovation, Vol. 5, No. 3–4, Oct. 2017, pp. 149–167, doi: 10.1080/21650349.2016.1203821.

[349] Bianchi, M., "Novelty, preferences, and fashion: when goods are unsettling", Journal of Economic Behavior & Organization, Vol. 47, No. 1, Jan. 2002, pp. 1–18, doi: 10.1016/S0167-2681(01)00162-7.

[350] Janssen, M., Stoopendaal, A. M. V., Putters, K., "Situated novelty: Introducing a process perspective on the study of innovation", Research Policy, Vol. 44, No. 10, Dec. 2015, pp. 1974–1984, doi: 10.1016/j.respol.2015.06.008.

[351] Likert, R., "The nature of highly effective groups", in Readings in organizational behavior, Gray, J. L., Starke, F. A., (eds.) Merrill, Columbus, OH, 1977, pp. 100–129.

[352] Martínez-Miranda, J., Aldea, A., Bañares-Alcántara, R., "Simulation of work teams using a multi-agent system", in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 1064–1065, doi: 10.1145/860575.860797.

[353] Brabazon, A., Silva, A., De Sousa, T. F., O’Neill, M., Matthews, R., Costa, E., "Agent-based modelling of product invention", in *Proceedings of the 7th annual conference on*

Genetic and evolutionary computation, 2005, pp. 129–136, doi: 10.1145/1068009.1068027.

[354] Wang, Y., Xu, Y., Zhang, X., "Optimization of product development process based on multi-agent simulation", in *International Conference on Cooperative Design, Visualization and Engineering*, 2009, pp. 351–358, doi: 10.1007/978-3-642-04265-2_53.

[355] Panchal, J. H., "Agent-based modeling of mass-collaborative product development processes", J. Comput. Inf. Sci. Eng., Vol. 9, No. 3, 2009, p. 31007, doi: 10.1115/1.3184605.

[356] Hsu, S.-C., "A complexity-based approach to intra-organizational team selection", University of Maryland, 2013.

[357] Fernandes, J. V., Henriques, E., Silva, A., Pimentel, C., "An Agent-based approach to support planning for change during early design", in *Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 3: Organisation and Management*, 2015, pp. 83–92.

[358] Raina, A., Cagan, J., McComb, C., "Transferring Design Strategies From Human to Computer and Across Design Problems", Journal of Mechanical Design, Vol. 141, No. 11, Nov. 2019, doi: 10.1115/1.4044258.

[359] Zhang, X., Thomson, V., "The Impact And Mitigation Of Complexity During Product Design", Int. J. Design & Nature and Ecodynamics, Vol. 11, No. 4, 2016, pp. 553–562, doi: 10.2495/DNE-V11-N4-553-562.

[360] Hulse, D., Tumer, K., Hoyle, C., Tumer, I., "Modeling Collaboration in Parameter Design Using Multiagent Learning", in Design Computing and Cognition ’18, Springer International Publishing, Cham, 2019, pp. 577–593.

APPENDIX A: SCOPE AND CHARACTERISTICS OF AGENT-BASED MODELS OF PRODUCT DEVELOPMENT TEAMWORK

Table A.1 Scope of agent-based models of product development teamwork

Model and references	Year of development	Purpose	Application area
Virtual Design Team - VDT [80], [87], [88] model	1994-2012	Estimation of project duration, production cost, coordination cost and process quality based on the simulation of project team's task execution and coordination.	Project teams performing routine engineering design or product development tasks. Later extensions enable simulation of teams executing non-routine tasks (e.g. health care delivery or equipment maintenance, agile software development teams).
Design Information-Flow Simulation (DiFS) [81] model	1995	Simulation of information exchange and coordination in teams during the design process.	Small design team (5 to 10 engineers) working on a 3 to 6 months long design process.
Mihm et al. [110] model	2003	Exploration of how project size (i.e. number of components) and team communication affect team performance.	Teams working on complex distributed design project.
TEAm Knowledge-based Structuring - TEAKS [83], [130], [352] model	2003-2011	Support of team configuration process.	Teams working on engineering projects for which tasks can be predefined.
Gero and Kannengiesser [44] model	2004	Exploration of how team expertise emerges in temporary design teams.	Temporary design teams.
Sosa and Gero [71] model	2005	Studies of impact the social context has on designer's creativity and creative behaviour.	Groups of designers, not necessarily a team in terms of cooperating agents (competing designers simulated).
InventSim [70], [353]	2005-2007	Simulation of product invention process to investigate the utility of different search heuristics.	Groups of designers, not necessarily a team in terms of cooperating agents (competing designers simulated).
Gonçalves et al. [82] model	2006	Support of team configuration process for a specific cooperative task (e.g. brainstorming).	Design team performing a collaborative task (e.g. brainstorming).
Singh and Gero [132] model	2007	Simulation of temporary design teams behaviour.	Temporary design teams.
Bellamine-Ben Saoud and Mark [95] model	2007	Support of cooperative processes planning by 1) providing a virtual collaboration environment for cooperation scenarios evaluation, and 2) enabling the study of group behaviour while monitoring an	Space mission design teams in aerospace organisations.

		information source and recovering errors.	
Zhang et al. models [84], [98], [120], [121], [126], [354]	2007-2014	Evaluation of human and organisational factors in collaborative product development process.	Collaborative product development teams.
Crowder et al. [137], [138] model	2008-2012	Examination of team performance sensitivity to changes in team configuration or processes.	Integrated product teams.
Olson et al. [94] model	2009	Theory building and research of interaction patterns arising during the collaborative design of complex systems.	Teams designing complex systems; modelled to imitate NASA's Jet Propulsion Laboratory design team
Le and Panchal [90], [355] model	2009-2011	Study of the co-evolution of the product and the community of developers.	Mass collaborative product development teams.
Singh et al. [131], [139] model	2009-2013	Study of the effect of team familiarity (i.e. transactive memory system) and social learning on team performance.	Design teams (either flat, distributed flat or functional)
Zhong and Ozdemir [72] model	2010	Study of the social structure influence on the speed of innovation.	Groups of designers, not necessarily a team in terms of cooperating agents (competing designers simulated).
Virtual Organisational Imitation for Construction Enterprises - VOICE [86], [100], [114] model	2010-2014	Assessing and managing the business complexity of construction enterprises.	Construction engineering and management (CEM) teams.
Blau et al. [96] model	2011	Study of how incentive schemes (individual or group) affect the teams' backlog dependency resolution behaviour.	Large-scale, lean software development team.
Son and Rojas [101], [103] model	2011-2015	Evolution of inter- and intra- team collaboration networks and their influence on team performance.	Temporary construction engineering and management teams.
Dehkordi et al. [77] model	2012	Study of the effect of project overload on innovation.	Advanced engineering teams.
Sosa and Gero [78] model	2012	Study of effect the team structure has on idea generation.	Design team working on a simple task of divergent reasoning.
Levine and Prietula [91] model	2013	Study on how the cooperativeness of participants, the diversity of their needs, and the degree to which the goods are rival (subtractable) affects the performance of open collaboration.	Teams working on open collaborative projects.
Hsu et al. [102], [356] model	2013-2016	Comparison of different team member selection methods in different market contexts.	Small, long-term construction design teams.
Ambler [104] model	2015	Examination of general methods for improving design in the form of 1) general strategies that incentivise beneficial collaborative team-	Large collaborative engineering design teams.

		formation dynamics, and 2) optimally structuring a design approach to minimise structural design complexity.	
Dutta et al. [115] model	2015	Estimation of project duration, production cost, coordination cost and process quality. Simulation of actions and interactions of team members to assess the emerging team phenomenon.	Teams participating in large engineering design projects (thousands of participants and tasks).
Farhangian et al. [97] model	2015	Evaluation of team performance based on skill competency and team members' personality composition.	Software development teams.
Herrmann [122] model	2015	Study of the effect of different problem-solving approaches on the team performance.	Product design teams. Problem-solving tasks simulated.
Singh and Casakin [75] model	2015	Study of effects use of analogy has on design team cohesion and collaboration.	Product design teams.
Zhang et al. [92] model	2015	Simulation of individual's behaviour in mass collaborative product development, with particular emphasis on help-seeking behaviour.	Mass collaborative product development teams.
Agent Model for Planning and rEsearch of eaRly dEsign - AMPERE [57], [357] model	2015-2017	Modelling early phases of complex design in order to provide planning support. Enables studies of team’s ability to deal with uncertainty and requirement change.	Product development teams working on a complex design. Early design phase simulated.
Cognitively-Inspired Simulated Annealing Teams - CISAT [112], [358] model	2015-2019	Simulation and analysis of design team's processes and performance achieved on a problem-solving task.	Small, flat design teams. Problem-solving tasks simulated.
Martynov and Abdelzaher [108] model	2016	Study of the effect that knowledge overlap, search width and problem complexity have on team performance on a problem-solving task.	Teams performing problem-solving activities.
Xia et al. [99] model	2016	Study of the impact that task-assignment strategies have on the learning process and evolution of knowledge.	Software R&D teams performing programming tasks (design tasks not modelled).
Zhang and Thomson [116], [359] model	2016-2019	Studies of the impact of knowledge and learning on a product design team performance and complexity mitigation.	Product design teams.
Zhou et al. [93] model	2016	Examining the impact of participants' characteristics (knowledge and cooperation willingness) on the evolution of the community project.	Teams working on open source design
Hulse et al. [113], [360] model	2018-2019	Studies of the effect of learning and collaboration on design outcomes in a distributed design task.	Teams performing complex engineering design tasks. Multidisciplinary design tasks simulated.
Bott and Mesmer [105] model	2019	Studies of effectiveness (in terms of team productivity) of waterfall and Agile processes in hardware-intensive systems	Hardware-intensive design teams. Every design phase modelled.

Singh et al. [106]	2019	Studies of the impact of social and cognitive features (e.g. self-efficacy, conformity, confidence, mental inertia) on team performance and design outcomes.	Flat teams performing co-design activities in early design phase.
KAI Agent-Based Organisational Optimisation Model - KABOOM [111], [127] model	2019	Studies of the effect of team member's cognitive-style preferences, communication patterns and specialisation on team processes and performance	Design teams performing problem-solving tasks.
Jafari Songhori et al. [109] model	2019	Studies of the impact of team formation strategies (regarding diversity and communication patterns) on the product quality.	Product development teams.

Table A.2 Characteristics of agent-based models for product development teamwork simulation

Model and references	**Key characteristics, positive**	**Limitations**
Virtual Design Team - VDT [80], [87], [88] model	First versions extensively validated. Agent's skill and experience level are included, attention allocation and effect of workload are modelled. Later versions include modelling of a dynamic transactive memory system of each agent, learning, forgetting, and effect of trust and cultural differences. Uncertainty and complexity of tasks are included; rework, failures and exceptions are modelled. Various communication tools are modelled, noise in communication included, meetings modelled, help-seeking behaviour implemented.	Later versions not extensively validated. Versions enabling non-routine task simulations not extensively validated. Agent's personality traits such as social competencies or extraversion not modelled. Agents do not learn by observation or imitation of others. Knowledge grounding, conception and creativity not represented.
Design Information-Flow Simulation (DiFS) [81] model	Verified and validated; sufficiency, plausibility and predictive power tested by comparison with real-world data. Synchronous and asynchronous communication modelled. Informal communication modelled. Task dependencies modelled. Agent's work and communication efficacy, short-term memory, environmental awareness and knowledge of tasks and contacts modelled.	Only small teams and short projects can be simulated due to a great level of detail represented. Noise in communication not modelled. Exception handling behaviour not modelled. Agents do not learn. Trust, shared mental models and effect of familiarity not modelled. Static agent parameters.
Mihm et al. [110] model	Verified; results' robustness to parameter change is tested. Scheduled and spontaneous information exchange between agents modelled. Agents' cooperative style (cooperative or uncooperative) and communication frequency varied. Effect of communication of preliminary information and partially connected networks studied. Performance function of each component (i.e. agent) is influenced by other agent's	No empirical validation provided. Noise and effect of social factors (e.g. trust in others) on communication not modelled. Agents do not differ in cooperative style or problem-solving capability. Local performance functions have a single optimum and equally affect other components. All agents are assumed to have skills to reach a local optimum.

	performance. Local performance functions change throughout the simulation.	
TEAm Knowledge-based Structuring - TEAKS [83], [130], [352] model	Validated through face validity and comparison with historical data. Personality traits, motivational factors and emotions included in the model. Collaborative tasks modelled. Agent's social capability, trust and emotions influence the collaborative task performance.	Tasks need to be predefined and preassigned to agents. Rework and exception handling not modelled. Agents do not learn. Reputation not modelled. Help-seeking behaviour not simulated. Agents do not interact informally or form trust relationship by providing back-up behaviour.
Gero and Kannengiesser [44] model	Agents are forming generalised knowledge about the world and their peers. Knowledge is dynamic, and knowledge grounding is included in the model. Agents are situated and form situated views of others' expertise (in the form of their function, behaviour and structure.) Agents interact with others and learn through interaction.	No verification or validation reported. Other aspects of teamwork not modelled (e.g. formation of social ties, personality traits, or task performance). No implementation details provided.
Sosa and Gero [71] model	Verified; replicated to ensure internal validity. Designer agents learn through imitation of others and experimentation. They learn rules and form strategies. Agents differ in processing and synthetic abilities. Adopter's evaluation effect on designer's behaviour modelled.	No testing against empirical data reported. Designer agents do not form social relationships or exchange information. Their interaction is restricted to the imitation of one another.
InventSim [70], [353]	Verified; sensitivity analysis performed. Several real-world search heuristics embedded in the agents (e.g. anchoring, trial-and-error, copying) Market demands and influence of competition on product's success are modelled. Dependency between product components taken into account, and product performance represented as rugged landscape.	Validation not reported. Agents are not characterised by personality traits, cognitive abilities, affect or social behaviour. No interactions between producers. Producers do not learn through observations of others. Rugged landscape (NK model) does not change during simulation - no effect of market change simulated.
Gonçalves et al. [82] model	Agent behaviour based on the empirical data. Agent's cognition, goals and motivation included in the model. The quality of brainstorming session measured regarding time, the number of states, cooperation level, divergence, and levels of indecision.	No verification or validation performed. Initial phases of the project described. No implementation details provided. Impractical task representation due to a large number of possible brainstorming states.
Singh and Gero [132] model	Comprehensive agent model: agents are characterised by motivation, attitude, skills, expertise, memories, experience, and beliefs. Agents are adaptive, have constructive memory, interact with others, and form and update mental model of the team. By building on their past experiences, agents are forming	No verification or validation reported. Implementation details not provided. Interaction rules and task representation details not defined.

	generalisations and grounding their knowledge.	
Bellamine-Ben Saoud and Mark [95] model	Verified and validated; sensitivity analysis performed and simulation output compared to empirical data. Discussions between subgroups in a team modelled. Agent's attention allocation and hearing modelled. Agents can overhear sidebars (i.e. small group) discussions between other team members and decide whether to join. Noise affects information exchange and error detection. Sidebar discussions and information exchange enable error detection.	Large amounts of data are necessary to run the model. Only two sidebar-joining strategies modelled. Agents cannot leave discussion once joined (until it finishes). Noise level depends only on the number of current sidebar discussions. Occurrence and duration of sidebars predefined. Agents can detect errors only during sidebar discussions.
Zhang et al. models [84], [98], [120], [121], [126], [354]	Models [84], [120], [126], [354] validated by comparison with empirical data. Detail modelling of agent's tasks. Task scheduling behaviour modelled and can be interrupted and re-continued iteratively. Task importance, urgency and personal preferences included in the utility function. Memory recovery time, learning and forgetting taken into account. Rework, exception handling and conflict resolution are modelled. Collaborative behaviour modelled.	Models [98], [121] not reported to be validated. The impact of motivational, social or affective aspects on agent's performance not modelled. Social aspects, e.g. trust, not considered during partner selection. Shared mental models not modelled. A transactive memory system is predefined and fixed.
Crowder et al. [137], [138] model	Rules guiding agent's behaviour are based on empirical data. Face validation of simulation outcomes performed. Task difficulty and dependencies modelled. The inclusion of learning from resources and peers. When agents have insufficient competencies, they request help from others. Shared mental models, trust and motivation modelled.	Simulation not applied to a real-world problem. Tasks are predefined and prescribed to agents. Agents can work only on one task at the time and must finish a given task to perform the next one. Agents do not learn through observation or by doing (only through interactions between peers and resources). Agents do not have transactive memory and send help requests to all other agents.
Olson et al. [94] model	Verified. Agent's mental model implemented and trained on past projects. Resolution of a problem (in case team performance is not satisfactory) modelled, thus simulating the effect of discussions initiated by the team leader. Collaborative tasks modelled, iterative and direct negotiations modelled. Agents differ in progress rate. Agent's problem-solving strategies based on processes observed in real-world. Agents can autonomously decide what task to work on next.	No validation reported. Significant amount of work needed to extend the model's applicability. In particular, much effort is needed to develop rich domain models. The agent's implementation statistically tailored for particular domain.
Le and Panchal [90], [355] model	The first model of mass collaborative product development teamwork. Agents decide whether to join by reasoning on cost and benefits of participation. Developed product is represented as a network of modules.	No validation reported. Cost and benefits are constant throughout the simulation. The number of modules is fixed, and product structure does not change during the simulation. The growth rate of each module implemented in the model. No distinction in skills or

		expertise requirements between modules modelled. No modelling of errors and difficulties caused by incongruent agent's goals. Differences in agents' skills, behaviour, or experience not modelled. Agents do not learn.
Singh et al. [131], [139] model	Verified, validated based on theoretical predictions. Model of social learning. Different kinds of learning modelled: learning by direct communication, learning by observation of other member's interactions and learning by observation of team members performing tasks. Agents have a (not necessarily accurate) mental model of "who knows what" and update it through direct or indirect interactions. Effects of communication frequency, business and membership retention studied. Different team structures simulated and compared.	Not applied to a real design problem. To isolate effects of social learning, aspects such as noise in communication, task delegation due to workload (and not due to lack of competencies) or learning by doing not modelled. Other aspects of teamwork not modelled.
Zhong and Ozdemir [72] model	Producers are represented with genetic algorithms, and possess sets of potential and realised relationships with their peers, over which they can exchange information. Agents are characterised by learning capability and can learn new skills through interactions with high performing peers. Three learning mechanisms modelled: probabilistic change, uphill climbing and crossover mechanism. Production guided by rewards obtained (i.e. consumer satisfaction).	Validation not reported. Agent's transactive memory system is accurate. Other aspects (e.g. influence of collaboration frequency or competition on knowledge exchange) not modelled. Consumers' preference function has a single optimum. All consumers share the same idea of the ideal product. Consumers' preference function does not change throughout simulation (no market change).
Virtual Organisational Imitation for Construction Enterprises - VOICE [86], [100], [114] model	Verification and face validation performed. Task attributes such as priority, difficulty, workload and dependencies modelled. Agent's attributes related to task execution and error handling (e.g. capacity, quality preference, work quality) included. Meetings and communication due to increased workload or informational dependencies modelled.	No validation based on empirical data reported. Informal information exchange not included. Agents are not characterised by personality traits, motivational level, or emotional state. Agents do not learn, form mental models or expertise. Agents do not seek help from their peers, learn through observation or collaboration, or form trust and transactive memory system.
Blau et al. [96] model	Verified, sensitivity analysis performed. Face validity performed. Lean and agile principles are modelled. Agent's local fitness function and learning factor are modelled.	The model was not tested against empirical data. Backlog processing time fixed. Agents do not differ in learning factors. Agents have no social features, expertise, or skills. Noise in communication and failures not modelled.
Son and Rojas [101], [103] model	Verified; sensitivity analysis and debugging reported. Model assumptions grounded in theory.	No validation of model outcomes reported. Other factors such as work structure and complementarity of knowledge

	Simple. Familiarity between agents and sociality parameters used to model the probability of agent's encounter. Familiarity decreases if agents do not interact. Each agent decides whether to cooperate (share info), or not based on payoff function.	do not influence collaboration. Contact is based only on social elements, and goal-driven collaboration is not modelled. The number of relationships is limited to three. Effect of indirect relationships is not considered.
Dehkordi et al. [77] model	Agents characterised by knowledge, capacity, preferred workload (number of parallel projects), and motivation. Agent's motivation is influenced by workload, as well as with other team members' motivation levels and knowledge. Agent's creativity parameter included. Agent's knowledge increases during interaction with more knowledgeable agents.	Not validated (modelled behaviour not compared to the real-world data). Theoretical background of model assumptions not explicated.
Sosa and Gero [78] model	Brainstorming task is represented as shape generation task. Agents can explore by random shape drawing and transformation, evaluate existing shapes in order to create new concepts, or exploit by applying learned concepts. Group influence parameter determines the percentage of concepts shared between members.	No validation reported. Influence of motivation and personality factors on the number of ideas shared and evaluations not modelled. Effects of trust and leadership style not modelled.
Levine and Prietula [91] model	Verified, sensitivity analysis performed. Validated by comparison with literature-based predictions. Agent's cooperative type modelled (cooperator, free rider or reciprocator). Agents send requests for help based on their transactive memory system. Resources (knowledge) are characterised by rivalry which affects the cost that an agent bear when cooperating.	Not tested on a real-world data. Agents do not form trust relationships or organise in communities. Agents do not learn about others' cooperativeness. Transactive memory system is static and accurate. Effect of ostracism not modelled. Agents do not differ in skills. Agents only produce resources prescribed by their tasks.
Hsu et al. [356][102] model	Verified through extensive testing, code work through and debugging. Validated by comparison with empirical data. Detail representation of agent's technical competencies and work-related features (e.g. education, license, working experience, salary).	Team performance measured only through profit. Quality or differences in project completion time not simulated. Agent's social competencies, personality traits, affective states or cognition not modelled.
Ambler [104] model	Verified; sensitivity analysis performed. Agents are guided by their understanding of the situation and reason about their next steps, are forming a perception of others (i.e. of their fitness) and exchanging information with them. Design space represented as NK model. Builds on principles of design from Axiomatic design	No validation reported. Agents do not differ in terms of preferences or goals and have the same understanding of the fitness function. No personality traits or social or affective aspects (e.g. trust, social competences) modelled. Static design landscape representation. Economic, market, legislative, and regulatory constraints not included.

Dutta et al. [115] model	Individuals can work on two tasks simultaneously, and several agents can work on the same task. Composite (team-level) measures introduced: composite capability, motivation and availability. Validated against data obtained from interviews.	Task iterations and rework not considered. Agents learn only from a leader or resource (i.e. no knowledge exchange between team members). Composite measures depend only on task-owners characteristics, and not on interactions between team members. Agents do not form an opinion about others (their knowledge, expertise, social competencies, trustworthiness, etc.).
Farhangian et al. [97] model	Agent's personality and skill competency are modelled. Team performance measure takes into account the estimation of how well team members get along in terms of matching of their personalities, creativity, roles capabilities and stress handling.	No validation of assumptions or results reported. Affective and motivational aspects not included. Demographic factors such as gender, age or social roles do not influence the relationships among agents.
Herrmann [122] model	Design problem-solving modelled as solution space exploration. Several collaboration (e.g. serial), problem-separation ("all at once" vs separating the problem to sub-problems) and search strategies ("hill climbing", “fixed effort” and “target values”) modelled and compared.	Validation not reported. Design space is static (does not address requirements change or market fluctuations). Expertise, experience, learning, and difference in agent's knowledge and skill not modelled.
Singh and Casakin [75] model	Study of the effect of the use of analogy on team performance. The distinction between analogy purposes (problem identification, function finding, solution generation or explanation) modelled. The expertise of team members and analogical distance are taken into account. Team members characterised by their knowledge and expertise domains. Each agent possesses a transactive memory.	No validation reported. Implementation details not provided (initial phase of research). The transactive memory system is fixed and accurate.
Zhang et al. [92] model	Modelling of individual's behaviour in terms of initiative, collaboration and autonomy. Uncertainty and complexity of processes taken into account. Three types of agents modelled: management, technical core and common development agent. Based on agent's type, its goals and behaviours differ. Agents can collaborate in off-line or on-line manner. If a problem encountered is difficult, not urgent and requires longer time, agents will collaborate in an off-line manner. If the problem is urgent, can be resolved quickly and is not difficult, agents will collaborate in an on-line manner.	No validation reported. No detail description of behaviours presented. Agents' interactions restricted to collaboration due to difficulties encountered. Agents do not form a transactive memory system.
Agent Model for Planning and rEsearch of eaRly dEsign - AMPERE [57], [357]	Belief-Desire-Intention (BDI) model of agency used. Customer, Project Lead and Design agents modelled. Design agents vary in the level of experience (Senior and Junior) which is reflected in their workload (and consequently time committed to a particular project), team	Performance not compared to the real-world data (i.e. no validation reported). Agents are characterised solely regarding time needed to complete a task, quality improvement, and actions they can perform. Personality (e.g. in response to

	influence, rate at which their work improves the solution, and work cost. Coordination of resources, change in requirements, risk evaluation, and solution submission to the customer modelled. These actions are directed by a project lead agent. Requirement change, iteration, rework and daily variability in individual performance included in the model.	stress) and interpersonal relations do not influence collaboration or performance. Adaptation to change modelled simply as a drop in solution quality. Approaching deadlines do not affect design agents, just the project lead agent which directs resources accordingly.
Cognitively-Inspired Simulated Annealing Teams - CISAT [112]	Characteristics of individuals are based on theory and are implemented to realistically represent human behaviour. Agents are biased in favour of their designs, interact in irregular intervals, and learn by doing. They tend to focus on most promising alternatives, avoid premature convergence by exploring different concepts, and employ different breadth- and depth-first search strategies, until a satisfying solution is found. Agents exchange ideas and evaluate each other's solutions. Problem-solving modelled as a search over rugged landscape. Agent's behaviour is based on simulated annealing and stochastic optimisation algorithms (which resemble problem-solving process employed by human designers). Verified and validated against empirical data.	Agent's characteristics like cognitive ability, personality traits or motivation not modelled. Agent's do not differ in their characteristics and do not form social relationships with others. The impact of conflicting goals or differences in understanding of design requirements not modelled. The rugged landscape does not change over the course of the simulation. Management or leader influence on agent's search not modelled.
Martynov and Abdelzaher [108] model	Design space modelled as a rugged landscape (NK model). Difference in agent’s expertise and knowledge taken into account: agents are capable of accurately rating only solution components which fall into their area of expertise. Noise in communication modelled.	Validation not reported. Design space is static. Agents do not increase their knowledge during the simulation. Agent's evaluations of the proposals not influenced by trust, agents are not self-biased. Exclusive use of majority rule to aggregate heterogeneous knowledge of team members.
Xia et al. [99] model	Verified, validated by comparison with theory-based predictions. Learning by doing and forgetting modelled as system dynamics model.	Agents never attempt to finish tasks (or learn) by the trial-and-error method. Agents do not differ in learning and forgetting rate. Effect of mutual trust, previous collaborative experiences or social capabilities on collaborative behaviour not modelled. Agents have a perfect knowledge of “who knows what”.
Zhang and Thomson [116], [359] model	Product complexity modelled. Technical and integration complexity differentiated. Rework, work efficiency and experience gain modelled. Consultation (i.e. meetings) and communication of the tasks outcomes among agents included.	Assumes that functional decomposition of the product is known in advance and static. Personality, affective and motivational aspects not included. Trust among agents not modelled. Static transactive memory system.

	Agents' performance on a task implemented as a system dynamics model. Learning depends on the technical complexity, agent's knowledge, consultation effort and helper's knowledge. Communication delay modelled. Verification, validation and sensitivity analysis reported.	Agents are learning only through consultation and communication (e.g. learning through doing not included). Forgetting not modelled.
Zhou et al. [93] model	Verified; sensitivity analysis performed. The inclusion of preferential attachment (collaboration is more likely between agents which already collaborated successfully). Agents learn through collaboration. Definition of agent's types based on participants' comments, information dissemination, and innovation ability.	No validation reported. Agents do not learn by performing a task in isolation, do not form expertise or preference towards certain tasks. Forgetting is not implemented. Agents do not form transactive memory. Incentives (e.g. committed effort, perceived value of the product, learning motivation, sense of group belonging) not modelled. Tasks selected randomly rather than based on preference, knowledge, urgency, or perceived importance.
Hulse et al. [113], [360] model	Control over design variables is distributed among agents to simulate collaboration on a complex engineered system. Agents are learning based on the received feedback. Agents adapt the level of solution space exploration (and exploitation) based on the performance of the past designs. Collaboration between agents (can be) included and varied. Comparison among three collaboration modes (collaboration, component asynchrony and variable asynchrony) presented.	The agent's behaviour not compared to the real-world scenarios (i.e. simulation validation not conducted). Communication biases (e.g. noise, delay, incomplete information received, misunderstandings) not included. Agents do not differ in their competence or personality (e.g. exploration is based solely on performance, and not influenced by preferences).
Bott and Mesmer [105] model	Agent's cognitive states are modelled as FBS processes (using first-order Markov process to model transitions from one state to the other). Verified. Calibrated and validated against real-world data. Coupling of design choices between design teams modelled.	Synthesis, analysis and evaluation collapsed into a single activity. Communication among agents modelled to be with zero time lag, and no noise or misunderstandings.
Singh et al. [106]	Agents' mental models change with experience (i.e. agents are capable of learning and forgetting). Agents differ in learning ability (or mental inertia) and forgetting rate. Agents are affected by others' ideas; conformity is included in the model. Self-efficacy parameter included and modelled to depend on motivation and experience.	Validation and verification not reported. No implementation details provided. Solution space exploration modelled as a search over a static surface. No noise in communication modelled. Transactive memory system and social factors such as homophily do not influence agent's decision to conform to other's solutions.

KAI Agent-Based Organisational Optimisation Model - KABOOM [111], [127]	Problem-solving simulated as a search over a rugged landscape. Agent's search influenced by simulated-annealing technique: temperature guides the probability of acceptance of other agent's solutions not improving the overall score. Agent's search strategy and perception of solution quality depend on agent's cognitive style. Agents have memory of past solutions, memory is biased to represent serial position effect. Agents are biased to move towards or away from other's solutions (influenced by a group conformity parameter). Communication among agents is modelled and occurs either in pairs (one-to-one), or in team meetings (all-to-all). Effect of cognitive gap on communication success included.	Acceptance of team's solution is based solely on solution's objective value (i.e. the solution scoring highest is accepted). Influence of authority, trust, conflicting goals or agent's bounded-rationality not taken into account. Interaction probability is a global constant. Communication pairs are chosen randomly (e.g. expertise-based, trust-based or homophily-based communication not modelled). Solution space modelled as static, symmetrical and bounded. Assumes perfect decomposition of problem variables along mutually exclusive dimensions.
Jafari Songhori et al. [109] model	Coordinated search process modelled as a search over a modified NK landscape (one parameter guides interactions among elements within the same subsystem, other parameter guides interactions among different subsystems). Agents have incomplete knowledge and domain of expertise. A pair of agents similar in expertise has a higher probability of successfully communicating solutions. Product complexity and product architecture modelled Teams grouped into clusters determining the rate of interactions among them. Teams differ in the level of diversity (regarding their domain of expertise) of their members.	No validation or verification reported. Landscape does not change during the simulation. Agents do not differ in capabilities or personality. Communication only in pairs and solely based on the organisational interactions. Designer's interactions are based (only) on the connected cavemen model. Product development processes executed simultaneously for all subsystems.

APPENDIX B: CONDITIONS FOR THE SOLUTION SPACE UNBOUNDEDNESS

An essential aspect of the developed model is the fact that agents are capable of generating new structures and expanding the solution space. Therefore, the necessary and sufficient conditions for the solution space unboundedness (in terms of the number of different structure nodes that can be generated) had to be explored.

If the generation mechanisms are union and concatenation as introduced in Chapter 5, it can easily be seen that an initial space composed of a network (or networks) with two links and a single intersecting node (i.e. an initial space composed of two links and three nodes) can serve to generate an infinite number of networks. These links can either be contained within two structure node networks, or form a single structure node network. Let each of the links form one structure node network (the analogous process can be applied in case of a single structure node): S_1 contains the first, and S_2 the second link. Since both of these networks contain elements not present in the other network, a union operation creates a new network different from both of them. Formally, let S denote a structure node network (regarded as a collection of nodes and links), and define $nodes(S)$ and $links(S)$ as functions returning the set of nodes, and set of links respectively. Let (a, b) denote an undirected and unweighted link connecting nodes a and b, thus $(a, b) = (b, a)$. It can be written:

$$nodes(S_1) = \{a, b\}, \qquad links(S_1) = \{(a, b)\}$$

$$nodes(S_2) = \{b, c\}, \qquad links(S_2) = \{(b, c)\}$$

$$S_1 \nsubseteq S_2 \wedge S_2 \nsubseteq S_1 \rightarrow \exists S_3 = S_1 \cup S_2 \bullet S_3 \neq S_1 \wedge S_3 \neq S_2$$

$$nodes(S_3) = \{a, b, c\}, \qquad links(S_3) = \{(a, b), (b, c)\}$$

The created network S_3 has two distinct links and three different nodes. Now, a concatenation operation can be applied to S_3 which will concatenate two of its nodes into a new node (e.g. link $b - c$), thus creating a new network. It is important to note that the newly generated network is necessarily different from the existing networks (since it contains a new node), but it also inevitably shares one node with either S_1 or S_2. Formally:

$$\#links(S_3) \geq 1 \rightarrow \exists S_3' \bullet S_3' \neq S_3$$

$$links(S_3') = \{(a, d)\}$$

$$nodes(S_3') = \{a, d\} \bullet a \in nodes(S_3) \wedge d \notin nodes(S_3)$$

Which can be written as:

$$a \in nodes(S_1 \cup \ S_2) \wedge d \notin nodes(S_1 \cup \ S_2)$$

$$(a \in nodes(S_1) \vee a \in nodes(S_2)\ ) \wedge d \notin nodes(S_1) \wedge d \notin nodes(S_2)$$

Thus, $S_3' \neq S_i \ \forall i \in \{1,2,3\}$, and network S_3' and one of the original structures (in this particular example, S_1) form an exactly the same situation as the starting one: two networks with intersecting links. Therefore, the described process (i.e. a union followed by a concatenation) can be reapplied to these two networks (S_3' and S_1), generating new networks S_4 and S_4'. It is easy to show that this procedure can be performed infinitely many times, each time generating a pair of new networks (albeit, with similar properties). The newly generated networks can be further combined with the existing ones to generate more diverse networks with different properties. Note, the described process generates an infinite number of networks even if concatenation of the two initial nodes results in the third (e.g. if $concatenate(a, b) = c$), the only difference being that the concatenation necessarily has to be applied to the other (i.e. $b - c$) link. Further, the described process can be directly applied to networks consisting of more nodes and links (as long as the two of the links present in the initial space share a node).

As described, the sufficiency of these conditions (i.e. initial space consisting of two intersecting links) to serve as a basis for an unbounded space can be proven by mathematical induction. The proof of necessity is similarly simple. The condition can be separated in two parts:

1. $\#links \geq 2$,
2. $\exists l_1 = (a, b),\ l_2 = (c, d) \in links \ \bullet \ l_1 \neq l_2 \wedge (a = c \vee a = d \vee b = c \vee b = d)$.

The necessity of the condition is discussed as follows:

- If an initial space consists of none or only one node, the proof of boundedness of the space is trivial as application of transformations does not change the initial space. A similar situation is one in which an initial space consists of two nodes and no links. Namely, if the nodes are separated in their respective networks, the only new structure which can be generated is the union of the two. If the initial space consist of a single network ($S_1 \bullet nodes(S_1) = \{a, b\},\ links(S_1) = \ \emptyset$), or of three networks ($S_1 \bullet nodes(S_1) = \{a, b\},\ links(S_1) = \ \emptyset$; $S_2 \bullet nodes(S_2) = \{a\},\ links(S_2) = \ \emptyset$; $S_3 \bullet nodes(S_3) = \{b\},\ links(S_3) = \ \emptyset$), no new networks can be generated as the union does not create a new network, and there are no links for concatenation to be applied. In short, if two nodes and no links form the initial space no concatenations can be performed, so irrespective of the initial space, the only way to expand the solution space is by performing unions. The largest number of transformations can be applied to the initial space in which each of the nodes forms one network. Taken the other way, the situation with two nodes and no links is equivalent to producing all non-empty subsets of a node-set $\{a, b\}$, meaning that the size of the space cannot exceed $2^2 - 1$. Similar reasoning can be applied in the case of an initial space containing *n* nodes and no links. Irrespective of the initial space, the possible space cannot exceed the size of $\sum_{i=1}^{n} \binom{n}{i} = 2^n - 1$.

- If there is just a single link (and two or more nodes) the similar bound can be found. Without loss of generality, let $\{a_i \mid \forall i \in \{1, \dots, n\}\}$ denote the n nodes, and (a_1, a_2) a link. It is important to note that, although concatenation (potentially) generates a new node $a_{n+1} = \ concatenate(a_{1,} a_2)$, in this case the new node is isolated – thus meaning that no further concatenations can be done. Let the initial space consists of $n + 1$ networks, n with a single node and one with only the link. One new network can be added by concatenation, (possibly) increasing the overall number of distinct nodes to $n + 1$. Now, this problem can be regarded as creation of all non-empty subsets of a set $\{a_i| \ \forall i \in \{1, \dots, n + 1\}\}$ for networks that do not contain the link, and all non-empty subsets of a set $\{a_i \middle| \ \forall i \in \{3, \dots, n + 1\}\}$ for networks that do contain the link. Overall, the number of networks which can be generated is: $2^{n+1} - 1$ of the networks which do not contain the link, and $2^{n-1} - 1$ of the networks that do contain the link. Since it is evident that other initial spaces would generate even smaller number of new networks then the case presented, it can be concluded that the generated space cannot exceed the size of $2^{n+1} - 1 + \ 2^{n-1} - 1 = \ 5 * \ 2^{n-1} - 2$.
- Finally, if the initial space consists of more than one link ($m \ \geq 2$), but these links have no intersections, following the same reasoning as in the case of one link and n nodes, it can be concluded that the space to be generated is constrained. Since there are no intersections, a finite number of concatenations (m) can be performed – therefore generating a finite set of elements (maximal possible number being $2m + n$). Now, the upper bound for the generated space size can again be obtained by calculating the number of all non-empty subsets.

For clarity, several points should be emphasised. First, concatenation is implemented as an injective function (i.e. each unordered pair of nodes is concatenated into a distinct node which cannot be obtained by concatenating any other two nodes). Similarly, concatenation of two particular nodes always returns the same node. Finally, no network can contain more than one instance of a particular node. In other words, if $S_1 \bullet nodes(S_1) = \{a, b\},\ links(S_1) = \{(a, b)\}$, and $concatenate(a, b) = c$, then:

$$S_1' \bullet nodes(S_1') = \{c\}, links(S_1') = \emptyset,$$

$$S_1 \cup S_1' = S_2 \bullet nodes(S_2) = \{a, b, c\}, links(S_2) = \{(a, b)\},$$

but if concatenation is applied to S_2 no new networks will be generated as $S_2' = S_1'$. Namely, concatenation would transform the link (a, b) into the node c ($concatenate\{a, b\} = c$), but a network cannot contain two instances of the same node, thus meaning that the only node left in the node-set of the S_2' network would be node c. Since S_2' and S_1' have the same node-set and link-set, they are the same network (i.e. generate the same structure node in the structure space).

To summarize, the space which can be generated by application of union and concatenation processes (as defined herein), is unbounded if and only if the initial space contains network(s) with at least two intersecting links.

APPENDIX C: LIST OF PARAMETER DESCRIPTIONS AND THEIR DEFAULT VALUES

Table C.1 presents a list of default parameter values as set in the studies conducted by the developed simulation tool. Any deviations from the overview presented herein are emphasised in the study description. The default parameter values were derived from the theories (e.g. [183], [231]), empirical studies ([223]) and existing computational models ([171]). Several parameters were derived from others to capture specific interdependencies: for example, the relation among sharing and activation thresholds (see Section 5.1.3 for details). Such interdependencies are clarified in Table C.1. Finally, several parameters guiding the agent's cognitive behaviour were set experimentally to tailor the simulation outcomes to fit the real-world behaviour (for example, so that agents tend to create and propose a credible number of structures [201], [326]). Nevertheless, due to deficiencies in the modelling of agents' communication (see Section 5.2), the formal calibration was not performed. The future work will experiment with the different parameter setups to study the impact of their change on the obtained results.

Table C.1 List of parameter descriptions and their default values

Parameter name	Parameter description	Range	Default value
activationThreshold	The level of node activation necessary for the node to become eligible for communicating and reasoning upon	[0,1]	0.50
analysisThreshold	The level of node activation necessary for the node to become eligible for detail analysis which permits new link creation	[0,1]	0.75
C	Link weight growth parameter; guides link grounding during the task performance	[0,10]	0.03
decay	Guides the decay of node's activation in between two tasks steps	[0,1]	0.20
reflexThreshold	Threshold guiding the difference between System 1 and System 2 thinking; links whose weight is	[0,1]	0.99

	above this threshold are processed at 'no cost'.		
forgettingRate	Parameter guiding how much of the prior-task link's weight is retained after the task completion if the link has not been used in the task	[0,1]	0.90
learningRate	Parameter guiding how much of the link's weight increase is retained after the task completion (if the link has been used)	[0,1]	0.30
Number of behaviour nodes within the agent's domain of expertise	An integer indicating how many behaviour nodes the agent can be an expert in	[1,5]	*Random uniform, unif(1,5)*
Number of past tasks remembered	An integer indicating how many of the past task the agent can remember. Used to set initial values for inhibition and base level activation (if enabled)	$\mathbb{N}_0$	10
maximumBLA	The maximal amount of activation which can be added to the node's (regular) activation due to relevance of the node in previous similar tasks	[0,1]	0.25 (*activation threshold * 0.50*)
maximumInitialnhibition	The maximal initial (i.e. at task start) amount of activation which can be subtracted from the node's regular activation due to negative associations tied to the node	[0,1]	0.25 (*activation threshold * 0.50*)
maximumInhibition	The maximal amount of activation which can be subtracted from the node's regular activation due to negative associations tied to the node	[0,1]	0.50 *(activation threshold)*
$t_{left\_\min}$	Percentage of time left when affect starts influencing agent's behaviour	[0,1]	0.50
$t_{left\_\max}$	Percentage of time left when affect reaches the *node analysis*	[0,1]	0.10

	threshold if none of the requirements is met		
cognitiveAbility	Parameter guiding agent's *focused capacity* and an initial weight of newly learnt links	[0,1]	0.50 or random from *Normal(0.5,0.166)*
focusCapacity	Parameter guiding the number of knowledge chunks which can be processed in one simulation step	{3,4,5}	4 (due to default values of *cognitive ability* parameter)
extraversion	Parameter influencing agent's *sharing threshold*.	[0,1]	0.50 or random from *Normal(0.5,0.166)*
sharingThreshold	The necessary link's activation (i.e. respective node's activation and link weight) or node's score threshold above which the agent wishes to communicate the knowledge element	[0,1]	0.75 *(analysis threshold)*
Maximal agreeableness influence on perception of remarks	Parameter influencing the perception of other agent's scores assigned to the proposed structure; a maximal value that can be added or subtracted from other agent's structure rating (it is scaled by the agent's agreeableness)	[0,1]	0.05
agreeableness	Parameter guiding initial trust value and perception of other agents' messages	[0,1]	0.50 or random from *Normal(0.5,0.166)*
maxTrust	The maximal amount of trust the agent has in a particular team member	[0,1]	0.75 (analysis threshold)
minTrust	The minimal amount of trust the agent has in a particular team member	[0,1]	0.25 (1 - analysis threshold)
trustChangeRate	Parameter guiding the update of trust in between two tasks	[0,1]	0.10
acceptanceThreshold	Parameter guiding the team's acceptance of a particular structure as a final solution. Team score assigned to a structure needs to exceed the *acceptance threshold* for the	[0,1]	0.95

	structure to be declared as a solution, thus terminating the simulation		
Number of agents	Number of agents forming a team	$\mathbb{N}$	3
Task duration (number of steps)	Number of simulation steps allocated for one task	$\mathbb{N}$	1,000

LIST OF FIGURES

LIST OF TABLES

BIOGRAPHY

Marija Majda Perišić was born in 1991, in Split, Croatia. In 2009, she enrolled in the study of Mathematics at the Faculty of Science, University of Zagreb. In 2012, she gained a bachelor's degree, and in 2014 a master's degree, specialising in Applied Mathematics.

In 2015, she accepted a position at the Chair of Design and Product Development, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb (UNIZAG-FSB), as a research assistant on a project funded by the Croatian Science Foundation (CSF) "Models and Methods of Innovation Management in the Development of Complex Engineering Systems - MInMED". The same year she enrolled in postgraduate doctoral studies at the Faculty of Electrical Engineering and Computing, University of Zagreb. She currently participates in CSF project "Team Adaptability for Innovation-Oriented Product Development – TAIDE".

During her PhD studies, Marija Majda attended an international summer school "European Agent Systems Summer School (EASSS)" at the University of Catania. In collaboration with dr.sc. Christopher Hoyle from Oregon State University, she has organised a scientific workshop titled "Agent-Based Modelling in Design", which preceded the "8th International Conference on Design Computing and Cognition –DCC18" held in Lecco. She has co-authored four journal papers and eight conference papers.

She is a member of the Design Society, and since 2016 she actively participates in the organisation of DESIGN conference.

.

BIBLIOGRAPHY

JOURNAL PAPERS

Perišić, M.M., Štorga, M., Podobnik, V. (2018). Agent-Based Modelling and Simulation of Product Development Teams. *Tehnički vjesnik : znanstveno-stručni časopis tehničkih fakulteta Sveučilišta u Osijeku*, 25(Supplement 2), 524-532. https://doi.org/10.17559/TV-20170912134413

Horvat, N., Škec, S., Perišić, M.M., Bojčetić, N. (2020). Relating Problem-Solving Approach to Turn-Taking in Communication of Design Teams. *Tehnički vjesnik : znanstveno-stručni časopis tehničkih fakulteta Sveučilišta u Osijeku*, 27(3). https://doi.org/10.17559/TV-20180911161642

Parraguez, P., Piccolo, S.A., Perišić, M.M., Štorga, M., Maier, A.M. (2019). Process Modularity Over Time: Modeling Process Execution as an Evolving Activity Network. *IEEE Transactions on Engineering Management*, 67, 1-13. https://doi.org/10.1109/tem.2019.2935932

Martinec, T., Škec, S., Savšek, T., Perišić, M. M. (2017). Work sampling for the production development: A case study of a supplier in European automotive industry. *Advances in Production Engineering & Management*, 12(4), 375-387. https://doi.org/10.14743/apem2017.4.265

CONFERENCE PAPERS

Perišić, M.M., Martinec, T., Štorga, M., Gero, J.S. (2019). A Computational Study of the Effect of Experience on Problem/Solution Space Exploration in Teams. In *Proceedings of the Design Society: International Conference on Engineering Design*, Delft, The Netherlands, May 5-8, 2019, Cambridge University Press, (pp. 11-20). https://doi.org/10.1017/dsi.2019.4

Perišić, M.M., Štorga, M., Gero, J.S. (2019). Situated novelty in computational creativity studies. In *Proceedings of the Tenth International Conference on Computational Creativity, June 17-21, 2019*, Charlotte, USA, Association for Computational Creativity, (pp. 286-290).

Horvat, N., Škec, S., Martinec, T., Lukačević, F., Perišić, M.M. (2019). Comparing Virtual Reality and Desktop Interface for Reviewing 3D CAD Models. In *Proceedings of the Design Society: International Conference on Engineering Design*, Delft, The Netherlands, May 5-8, 2019, Cambridge University Press, (pp. 1923-1932). https://doi.org/10.1017/dsi.2019.198

Miler, D., Perišić, M.M., Mašović, R., Žeželj, D. (2019). Predicting student academic performance in Machine elements course. *Advances in Mechanism and Machine*, Springer International Publishing, Cham (pp. 825-834). https://doi.org/10.1007/978-3-030-20131-9_82

Perišić, M.M., Štorga, M., Gero, J.S. (2019). Exploring the Effect of Experience on Team Behavior: A Computational Approach. In *Proceedings of the 8th International Conference on Design Computing and Cognition*, Lecco, Italy, July 2-4, 2018, Springer International Publishing, (pp. 595-612). https://doi.org/10.1007/978-3-030-05363-5_32

Perišić, M.M., Štorga, M., Gero, J.S. (2017). Building a computational laboratory for the study of team behaviour in product development. In *Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 8: Human Behaviour in Design*, Vancouver, Canada, The Design Society, (pp. 189-198).

Perišić, M.M., Martinec, T., Štorga, M., Kanduč, T. (2016). Agent-based Simulation Framework to Support Management of Teams Performing Development Activities. In *Proceedings of the 14th International Design Conference (DESIGN 2016*), Cavtat, Croatia, May 16-19, 2016, The Design Society, (pp. 1925-1936).

Štorga, M., Škec, S., Perišić, M.M., Martinec, T. (2016). Monitoring, Measurement and Simulation of Individual and Teamwork Performance in Collaborative Product Development and Design. In *Proceedings of the Human Life Design (HLD) International Symposium JAIST 2016*, Kanazawa, Japan.

ŽIVOTOPIS

Marija Majda Perišić je rođena u Splitu, 1991. godine. Studij matematike na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu (UNIZAG-PMF-MO) upisala je 2009. godine. Diplomu prvostupnice matematike je stekla 2012. Iste godine je upisala diplomski sveučilišni studij Primijenjene matematike na UNIZAG-PMF-MO. Zvanje magistre matematike stekla je 2014. godine.

Početkom 2015. godine zaposlila se na Katedri za konstruiranje i razvoj proizvoda Fakulteta strojarstva i brodogradnje Sveučilišta u Zagrebu (UNIZAG-FSB) kao stručni suradnik na projektu Hrvatske zaklade za znanost (HRZZ) "Modeli i metode upravljanja inovacijama u razvoju kompleksnih inženjerskih sustava". U ožujku iste godine upisala je poslijediplomski doktorski studij na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Od 2017. radi kao asistent na UNIZAG-FSB. Od 2018. sudjeluje u HRZZ projektu "Timska adaptabilnost u razvoju inovativnih proizvoda".

Usavršavala se na ljetnoj školi "European Agent Systems Summer School" na Sveučilištu u Cataniji. U suradnji s dr.sc. Christopherom Hoyleom s Državnog sveučilišta Oregon, organizirala je znanstvenu radionicu pod nazivom "Agent-Based Modelling in Design" koja je prethodila međunarodnoj konferenciji "Design Computing and Cognition" održanoj u Leccu. Objavila je 4 rada u znanstvenim časopisima te 8 konferencijskih znanstvenih radova.

Članica je zajednice Design Society. Od 2016. godine aktivno sudjeluje u organizaciji međunarodne DESIGN konferencije.