Exploring the effect of experience on team behavior: A computational approach

Marija Majda Perišić[1], Mario Štorga[1,2] and John S. Gero[3,4]
[1]*University of Zagreb, Croatia*
[2]*Luleå University of Technology, Sweden*
[3]*University of North Carolina at Charlotte, USA*
[4]*George Mason University, USA*

The paper presents the results of research aimed at contributing to a better understanding of the effect of team experience and learning on the performance of a design team. An agent-based model of the design team was developed, and computational simulations were utilized to study how agent's knowledge changes by its use and what are the effects of such changes on the team behavior. Particularly, hypotheses stating that as team members work together, they become more efficient and explore less have been studied. Computational experiments demonstrated the positive impact of learning and prior experience on team efficiency and indicated that team experience has a strong influence on the breadth of solutions examined by a team in a constructive task. However, results also suggest that increased knowledge grounding could have detrimental effects on a team's performance when teams are faced with tasks which do not fall into team's expertise.

Introduction

Learning, as a process of acquiring knowledge and skills obtained from participating in events and activities [1], is a fundamental process for achieving progress in any field of human actions. By developing and reusing well-grounded beliefs, humans can solve problems more efficiently, as well as develop generalizations which could be used in novel situations. Cognitive scientists, however, emphasize that learning and the resulting knowledge necessarily depends on the situation [2]. Situated cognition views learning as a constructive process where each new experience is shaped by previous experiences and understanding of the current situation, while new experiences, in turn, re-shape the understanding of previous experiences. Particularly, the theory of situated learning states that learning cannot be separated from the social context in which it occurs, therefore emphasizing the role of social interactions in the formation of new knowledge.

Designers, while working in teams, are necessarily influencing each other's learning processes. It has been recognized in the literature that designers are affected by their prior experience and learned patterns [3]. How the experience of each team member impacts the performance of the design team and how does the team behavior change as team member's work together are questions which are still not well addressed.

To contribute to a better understanding of the effect of team experience and learning on the performance of a design team, this work presents the results of set of computational experiments performed utilizing an agent-based framework. These experiments aimed to explore the effect of knowledge grounding and experience on the efficiency of a stable design team by studying how the time needed to find a solution, as well as team's learning and solution-sharing behavior change over time. The goal of this paper is to show that, by modeling and implementing agents whose behavior is based on cognitive theories, computational simulations can increase the understanding of the behavior of the individuals and teams in constructive tasks, and produce results which match propositions based on cognitive theories.

The remainder of the paper is structured as follows: first, a short overview of related work in the fields of cognitive behavior of individuals, learning in teams and effect of team experience and tenure on design team performance is presented, as well as a short review of computational simulators used in the studies of team behavior in engineering design. Next, the aims of this work are explicated in the form of hypotheses derived from the literature review. The methods used to test the hypotheses are described in Section 4 where the computational framework, measures, and case studies are presented in detail. In Section 5, results of the experiments are given and are further analyzed and discussed in Section 6. Finally, the paper concludes with a discussion of the limitations of the study and an outline of possible future work.

Related work

Cognitive behavior of designer

When developing a model of situated reasoning and learning in design, several authors [1], [4] have built on a descriptive model proposed by Gero and Fujii [5]. Maher and Gero [6] distinguish three modes of reasoning: reflexive, reactive and reflective. Reflexive reasoning is defined as producing a reaction without overt reasoning, a reflex. Reactive reasoning includes perceiving a situation and choosing actions which will lead towards the desired goal. In addition to perceiving the situation, reflective reasoning comprises hypothesizing on possible outcomes and determining desired states, therefore enabling the concepts to change, consequently changing the goals and experiences. As knowledge is being successfully used, it becomes more grounded and requires less cognitive effort to process, i.e., it becomes reflexive [1]. This theory is complementary to the Kahneman's notion of System 1 and System 2 thinking, presented in his book *Thinking, Fast and Slow* [7]. Kahneman posits that there are two modes of thinking: System 1, which is fast, automatic, reflexive and subconscious, and System 2, which is effortful, conscious and based on logic. Further, he offers several pieces of evidence of environmental impact on the thought process of an individual, particularly emphasizing the anchoring, a cognitive bias directed towards first-seen proposals. By building on these theories, one can describe and model phenomena such as knowledge grounding and create experiments which have the potential to further enrich our understanding of some of the known design behaviors such as fixation or formation of design patterns [8].

Design team experience and its effect on team learning and performance

In a detailed meta-analytic review of the determinants of team performance in new product development, Sivasubramaniam et al. [9] listed team tenure, internal and external communication, team ability, functional diversity, group cohesiveness, goal clarity, and leadership as critical factors influencing team performance. Team ability was defined as having the knowledge and experience needed to deal with complex design tasks and was hypothesized to have a positive impact on team performance. Strong support for this hypothesis has been found, thus confirming and extending the findings from a study by Carbonell and Rodriguez [10], which posited that team experience is positively related to speed to market. Sivasubramaniam et al. further found strong support for a hypothesis that team tenure is positively associated with speed to market.

Similar findings were presented by Akgun and Lynn [11] in their study of consequences of team stability (i.e., tenure) on the team performance. Akgun and Lynn [11] posited that team tenure impacts team learning and team experience, as stability promotes learning about team tasks and about the team itself. By creating a tacit task-related knowledge of “know how” and “know why,” and team related knowledge of “who knows what," team members can better coordinate their efforts, and consequently achieve higher speed to market. These studies indicate that experienced teams create products in a timely manner, which is likely influenced by the similarity of team members’ mental models and preferences developed through their experience which allow them to reach consensus faster.

However, several studies [12], [13] emphasize a curvilinear relationship between design team tenure and performance. Researchers [11], [13] state that in turbulent environments, team mental models can become obsolete and introducing new members can bring beneficial new knowledge and promote success. Similarly, Choi and Thompson [14] stress that, due to the tenure’s impact on an increase in homogeneity of members’ mental models, stable teams may struggle to produce creative results in innovative tasks. These authors presented a study which shows that a number of non-redundant ideas are significantly higher in groups which suffered membership change. Interestingly, findings presented by Gibson and Gibbs [15] indicate that team innovativeness is lower in less stable teams. Possible explanations for such contradictory findings were proposed by Hirst [16], who suggests that timing of turnover matters, while Badke-Schaub et al. [17] stress the importance of balance between knowledge overlap and distributed knowledge. Nevertheless, more research is needed to understand positive and negative impacts of team member’s experience, in relation to team environment regarding the nature of team task, organizational and market factors, and temporal factors such as the part of the product lifecycle the team is working on.

Computational studies of team behavior in design

Several computational models used to study different aspects of team behavior in product design have been developed. For example, McComb et al. [18] developed an agent-based model used to study the problem-solving behavior of design teams. In their model, the authors have defined and implemented agents based on research on the behavior of designers: agents share a common goal, they learn, interact at irregular intervals, have a bias in favor of their solutions, and focus on promising alternatives. In their work, an agent's search for a solution is implemented as a simulated annealing process.

Ambler [19] presented a model where each design concept is represented as a distinct point in Kauffman's NK model. Agents roam such space and perceive their relative fitness and current team situation, and can decide on their next steps. In addition to exploring neighboring locations, agents can provide jumping-off locations for future newcomers and can jump to previously found distant concepts. Agents communicate their relative fitness through collaborative linkages.

Models which focused on specific types of team learning and team expertise formation are found in Singh et al. [20] and Gero and Kannengiesser [21]. Singh et al. [20] focused on learning about "who knows what" in design teams. They explored how transactive memory is formed in flat, distributed and functional teams and further studied its impact on activity coordination and team effectiveness. Gero and Kannengiesser [21] studied the formation of team expertise in temporary design teams where agents are frequently required to adapt to new team formation.

Team exploration of solution space has been studied by Sosa and Gero [22], who developed an idea-agent-social context framework within which agents combine known geometric shapes to create as many diverse forms as they can. This model has been used to simulate ideation task, but no cognitive mechanisms of ideation were implemented in their agents.

Aims and hypotheses

Building on theoretical studies of the cognitive behavior of individuals and research findings on team learning, the current work aims to create a computational model of a design team and utilize it to gain experimentally-based insights into how experience gained by working together impacts the behavior of the team. More specifically, this work is aimed at testing two hypotheses:

As team members collaborate, they learn about each other and from each other. In other words, interactions between team members enable the creation, modification, and reinforcement of a team mental model [23]. Shared mental models, in turn, reduce the coordination cost and support reaching consensus. Experience gained and the development of shared mental models result in higher efficiency of the team, in terms of less time required to find a solution. Therefore, the first hypothesis is formulated as:

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

As team members work together, similar knowledge gets reused and becomes more grounded. As a result, teams develop a preference for particular solutions which proved to be suitable in previous projects, and reuse them at the expense of other solutions. Therefore, the second hypothesis is formulated as:

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

Model

An agent-based model of the design team is conceptualized, implemented and utilized to test these two hypotheses. Each team member is represented as an agent whose architecture is derived from previously published work [24].

Agent's cognitive behavior

The main driver of an agent's behavior is its mental model which is represented as a directed network. To contextualize the agents and tasks the FBS ontology was selected [3]. Building on those two approaches, nodes in the agent's mental model represent design task's decomposition and abstraction to the function, behavior or structure domain that the agent is familiar with. The agent's experience is presented as weighted, directed links between nodes. The agent's mental model (i.e. agent's knowledge network) consists of nodes of three types: function, behavior and structure nodes, which are connected by directed links whose weight indicates the perceived association between function and behavior, and behavior and structures. There are no links between nodes of the same type, and structure nodes can be reached from function nodes only through the behavior nodes [3].

Following Kahneman's theory of cognitive behavior and theory presented in [7], the agents have two ways of reasoning. Reflexive reasoning (System1) is implemented as spreading activation through the agent's mental model. The activation which spreads from an activated node to an adjacent node is proportional to the weight of the link connecting them, thus indicating the strength of their association. To model System2 thinking agents had to be equipped with additional reasoning mechanisms. Thus, instead of modeling structure nodes as simple nodes, each structure node is associated with a randomly created network. Behavior nodes are associated with selected network measures: diameter, a number of clusters, betweenness, closeness and degree centralities. By modeling behavior and structure nodes in such a manner, the natural association between structure nodes (i.e., networks) and behavior nodes (i.e., network measures) is created. The representation of the derived agent's mental model based on the FBS ontology and dynamic networks is presented on the Figure 1.

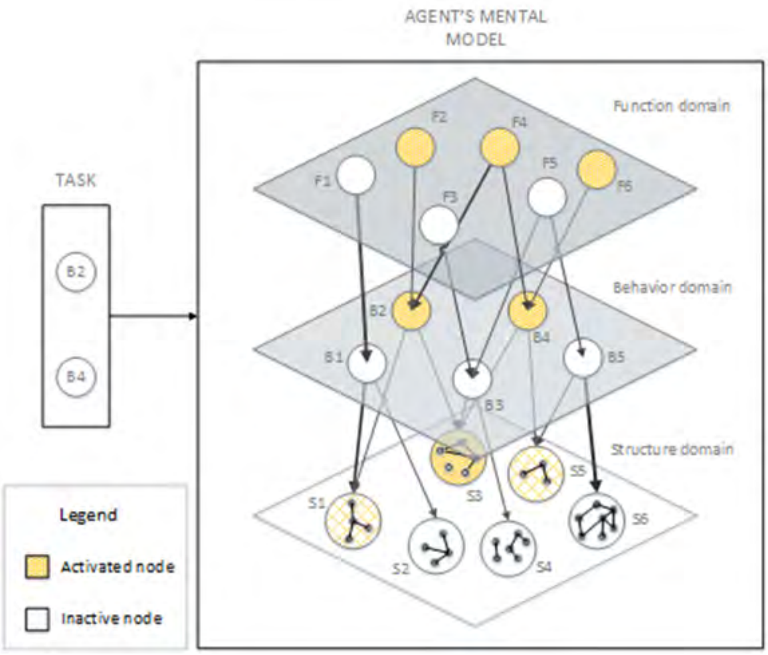

Fig1. The representation of agent's mental model

When a structure domain node's activation exceeds the analysis threshold, the agent analyzes the node (i.e., node's associated network) and creates, or further grounds, relationships to behavior domain nodes as determined by calculating the associated network's properties. As links increase in weight, they may exceed the reflexive threshold and agent will process them without extended reasoning i.e., reflexively. This results in the activations passing through a node and enabling the activation to reach in a single step nodes which are two or more links apart. If a link is not being used, its weight is decremented. i.e., if the knowledge is not used, agents slowly forget.

Agents can also expand their knowledge space by creating new structure nodes: if several structure nodes are activated, an agent can combine the networks associated with the activated structure nodes, thus producing a new network which is then named as a new structure node. Therefore, agents can create new solutions which display a novel combination of network properties, i.e., they can create structure nodes which present previously unseen combination of behaviors. Over time, these processes change the agent's mental model, Figure 2.

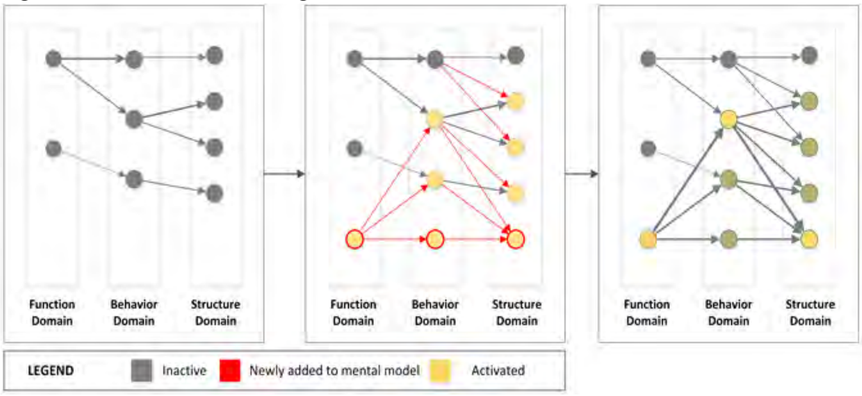

Fig2. The change of agent's mental model over time

Task processing

A design task is given to the team of agents in the form of a list of behaviors that a structure node is required to meet to be considered as a valid solution for the task. As the task is received an activation is generated. Since an activation can be passed to required behavior nodes only through function domain nodes, the agents also learn which function nodes are relevant (i.e., connected to desired behavior) and whether to activate them when a specific task is introduced. If a function domain node is sufficiently activated, or no progress has been made and activation accumulates in the task (i.e. it is not passed to function nodes), an agent can analyze the task or a function and determine whether a function node should be activated for a given task.

Through spreading activation and analysis, the agents search for the structure domain nodes which produce the required behaviors. Once a structure node is sufficiently activated, its associated network is evaluated against required behaviors, and if the structure node (more precisely, its associated network) is deemed as satisfying it will be proposed to others as the final solution. If, however, the structure domain node does not meet the required behavior, an agent can direct its actions by sending an activation to the function nodes which are connected to the unmet requirement.

To implement the previously presented theoretical background, the agent's learning is affected by interactions between team members. An "idea" consists either of a structure domain node whose activation level exceeds the sharing threshold, or of a knowledge link (a relation between a task and a function node, function and behavior node, or behavior and structure node) whose weight is sufficient and the respective nodes' activation exceeds the activation threshold. As agents encounter either structure domain nodes, or knowledge links worth sharing, they can propose them to others. Other agent's chain of actions is interrupted not just by additional activations sent to nodes the proposing agent has activated, but also by the creation of novel links and increase in weight of existing links which are being activated.

If a solution is proposed, every team member evaluates it against its mental model and produces a score. Each score is weighted with the factor indicating the agent's reputation (i.e., average trust in the agent). If agents rate the solution as sufficient, the simulation is over. If, however, the solution is rejected, agents continue their search until a new solution is found or time runs out.

Additional agent's behavioral mechanisms

To account for the other elements of the agent's architecture presented in [24], the implemented agent's cognitive mechanisms are biased by its:

1. *personality factors*, which influence the acceptance of novel ideas (based on agreeableness and openness to novel experiences factors), and sharing behavior (based on extraversion factor);
2. *trust* it holds for the others, which influences the impact other agent's proposals will have on the agent's mental model;
3. *frustration (affect),* which represents agent's affective state and guides agent's behavior in the following manner: as time passes and no solution has been found, or sufficient time passes with no progress, the agent's frustration increases and causes the weaker links to temporarily increase in weight, therefore enabling a wider search of the knowledge space. In the extreme, frustration causes all of the knowledge link weights to be regarded as similar. If a team experiences progress, i.e. solutions better than previous maximum has been found, the frustration drops and the agent focuses on the part of the knowledge where improvement has been found.

Description of a simulation run

Agents and a task are initialized at the commencement of a simulation run. Agents' knowledge networks are created. However, they are neither exhaustive (i.e. none of the agents is familiar with all of tasks, links, or function, behavior and structure domain nodes known to others), nor completely accurate (i.e. several incorrect links between behavior and structure domain nodes are introduced in an agent's model). Additionally, trust between agents is computed based on their initial knowledge related to the task. Trust others hold for an agent is proportional to the extent the agent is familiar with the task (i.e., with related functions and behaviors).

At the simulation start, each agent is presented with the task, which generates an activation, thus triggering a spreading activation mechanism. Through the cognitive mechanisms described above, agents search for a structure domain node which satisfies the requirements. If, either an existing or newly generated, structure domain node is deemed as suitable, it is stored in the agent's working memory and can be shared with others. Otherwise, if two nodes are both active and connected with a link of sufficient weight, the knowledge link is stored in the agent's working memory, and an agent can share it. An agent's working memory is constrained and can store a limited number of different links, and structure nodes. At any time step, an element stored in agent's working memory can be shared, but the content of working memory changes as additional structure domain nodes and knowledge links are activated. If in any time step two or more agents decide to share a structure domain node or a link with others, the dominant one is chosen randomly. As previously described, each agent is affected by what is shared as new activations are sent either to proposed structure domain nodes or to nodes activated when a link is shared.

The simulation is over either when agents agree on the solution or the time limit is exceeded, where time is treated as a scarce resource.

Design of the experiments

To test the two hypotheses presented, the agent-based model has been run with two types of task sequences, where *A* and *B* are two separate tasks:

T1. "$A\ B_1\ B_2\ \ldots\ B_n\ A$", where n = 1, 3, 5 or 10, and B_m is not equal to *A* for any *m*. Any two B_i and B_j can, but do not have to resemble each other, i.e. share some of the requirements.

T2. "*A*" *vs* "*B A*" *vs* "*B B (…) B A*", where *B* has been repeated 3, 5 or 10 times. Task *B* is related to task *A* as they require at least one common function, but are not the same, i.e. at least one function is different.

In the first type of task sequences (T1), tasks *Bi* have been chosen randomly, and may or may not have some functions or required behaviors in common with *A* or with each other. Cases range from simple tasks where multiple solutions for the tasks are possible and known to the team, to cases where no solution is known for the any given task. The experiment with the second task type (T2) is performed to test whether relevant, yet narrow experience on the related task impacts the performance on the task *A*. Following the hypotheses H1 and H2, it is expected that in both cases, where the team gained the diverse experience (T1) as well as restricted, but relevant experience (T2), will result in fewer simulation steps and less exploration of the solution space needed to find the solution.

To evaluate each simulation run, the following measurements have been collected:

M1. solution score as determined by the team (subjective, as determined by the agents)
M2. number of steps required to reach the solution, or max number of steps if session is over and no solutions are found
M3. number of new structure nodes created, i.e. number of distinct networks obtained through combining networks associated with existing structure domain nodes (overall, throughout session)
M4. number of new links learnt (overall) during the session
M5. number of distinct structure nodes proposed during the session (either new or existing)
M6. number of distinct links communicated during the session
M7. actual (i.e., objective) score, calculated as a percentage of required behaviors that a given solution satisfies.

Measures M1 and M2 provide the basis for whether teams become more efficient measured by the number of steps needed to find a solution and change in a score determined by the team, i.e. they are related to hypothesis H1. Steps needed (M2) and the score as determined by the team (M6) are related as teams terminate their search if the solution whose score is determined to be high enough is found. The number of new structures created (M3), number of new links learned (M4), number of distinct structures proposed (M5), and number of links shared (M6), provide the basis for whether a team explores less, i.e. they provide the basis for insights related to the hypothesis H2. Following the hypotheses, it is expected that scores for measures M3-M6 will be lower as more tasks have been performed by the team. Similarly, it is expected that the number of steps will decrease with team experience and that a score as determined by the team will be higher to show that a team reaches consensus quicker. The last measure, the objective (i.e., actual) score of proposed solution (M7) measures a team's effectiveness (i.e., an objective success), rather than efficiency (i.e., meeting the schedule). It is provided to gain better insights into positive and negative aspects of team experience on team performance.

The model is implemented in the multi-agent simulation environment MASON. Simulations were run 50 times for each of the task types (T1 and T2). The maximum number of steps ($M2_{max}$) was set to 300, the number of agents within a team was set to three, and there were 30 initial structure domain nodes (i.e., distinct networks). A network associated with a single structure domain node consists of 4 to 10 nodes, with links randomly distributed between them. Nodes in the networks associated with structure domain nodes (i.e., nodes from "structure node networks") were chosen from a set of 50 distinct nodes. The number of functions was set to 15, while the number of behavior domain nodes varied from 10 to 50.

Simulation results and hypothesis testing

The average values for each of the measures M1-M7 in task sequences of types T1 and T2 are listed in the Tables 1 and 2 respectively. The data for the first task sequence type (T1) consists of the details on team performances on the first task *A* and on the last task *A* (i.e., second-time task *A* was performed). The data for the second task sequence type (T2) consists of the details on team's performance of task *A* if no tasks *B* were performed before task *A*, and on the team performance on task *A* after multiple performances of task *B*.

Table 1 Mean values and standard deviations (in brackets) for the results from T1

Task type	**M1: Score rated by agents**	**M2: Steps**	**M3: New structures**	**M4: New links**	**M5: Proposed structures**	**M6: Communicated links**	**M7: Score**
AB_1A	0.6732 (0.26)	217.8 (103.6)	72.9 (63.4)	514.1 (371.1)	27.2 (20.3)	61.8 (44.7)	0.7553 (0.27)
AB_1**A**	0.7246 (0.24)	165.6 (140.3)	21.9 (39.5)	154.9 (224.6)	20.8 (21.7)	45.3 (45.7)	0.773 (0.26)
A$B_{1,2,3}$A	0.5371 (0.29)	267.3 (72.8)	82.1 (54.4)	576.5 (306.2)	27.8 (15.9)	63.5 (35.2)	0.6150 (0.33)

$AB_{1,2,3}\underline{\mathbf{A}}$	0.6224 (0.30)	202.9 (121.7)	21.8 (29.5)	151.8 (176.3)	20.3 (17.8)	45.0 (39.3)	0.6547 (0.34)
$\underline{\mathbf{A}}B_{1,..,5}A$	0.6156 (0.25)	254.9 (80.8)	95.7 (56.3)	661.6 (324.5)	33.4 (18.0)	77.5 (39.5)	0.7173 (0.28)
$AB_{1,..,5}\underline{\mathbf{A}}$	0.6979 (0.29)	178.6 (118.7)	11.7 (21.6)	90.5 (133.1)	16.9 (19.1)	36.1 (39.3)	0.6927 (0.3)
$\underline{\mathbf{A}}B_{1,..10}A$	0.5542 (0.29)	215.8 (114.7)	71.2 (49.5)	495.6 (314.5)	26.9 (18.3)	61.8 (41.5)	0.6483 (0.31)
$AB_{1,..10}\underline{\mathbf{A}}$	0.6685 (0.30)	208.9 (104.5)	16.1 (24.9)	94.7 (132.9)	14.1 (14.5)	34.3 (32.2)	0.6294 (0.31)

Table 2 Mean values and standard deviations (in brackets) for the results from T2

Task type	**M1: Score rated by agents**	**M2: Steps**	**M3: New structures**	**M4: New links**	**M5: Proposed structures**	**M6: Communicated links**	**M7: Score**
$\underline{\mathbf{A}}$	0.6520 (0.24)	252.3 (78.5)	93.7 (60.5)	642.4 (348.7)	29.2 (17.0)	69.0 (36.2)	0.7362 (0.26)
$B\underline{\mathbf{A}}$	0.6881 (0.23)	233.6 (94.9)	61.7 (54.8)	419.1 (308.0)	27.0 (18.0)	62.9 (37.3)	0.7438 (0.25)
$B^3\underline{\mathbf{A}}$	0.6951 (0.26)	221.1 (102.0)	33.7 (43.4)	245.5 (257.4)	23.2 (18.8)	51.1 (38.2)	0.7472 (0.24)
$B^5\underline{\mathbf{A}}$	0.7555 (0.24)	210.3 (101.6)	28.7 (41.1)	209.9 (238.4)	16.6 (14.8)	40.9 (32.8)	0.7310 (0.25)
$B^{10}\underline{\mathbf{A}}$	0.6772 (0.29)	220.6 (100.5)	16.3 (27.8)	129.5 (158.3)	12.1 (13.0)	30.9 (25.6)	0.6459 (0.30)

To test hypotheses H1 and H2 significance testing was carried out. The simulation results were tested for normality and were shown not to be normally distributed. As a consequence the Wilcoxon's signed-ranked test was utilized on the simulation results. More detailed hypotheses formulated and tested on the task sequences of the type T1 were as follows:

1. $dM1_{First} < dM1_{Second}$
2. $dM2_{First} > dM2_{Second}$
3. $dM3_{First} > dM3_{Second}$
4. $dM4_{First} > dM4_{Second}$
5. $dM5_{First} > dM5_{Second}$
6. $dM6_{First} > dM6_{Second}$
7. $dM7_{First} < dM7_{Second}$

where dMi_{First} and dMi_{Second} for $i = 1, \ldots, 7$, represent the distributions of the collected data by measuring M1-M7 on the first and the second task *A* respectively.

Similar hypotheses were formulated for the case where performance of a single task *A* was compared to the performance of task *A* after multiple completions of task *B* (T2). The results are presented in Tables 3 and 4. Further, to gain deeper insights in how the performance of multiple tasks *B* impacts the performance of task *A*, the data for task type T2 were separated into cases where (objective) solutions for the task *A* were found by the agents while performing task *B*, and where no solution for the task *A* was found while performing task *B*. The results for listed cases are presented in Tables 5 and 6.

Table 3 *p*-values for T1

Task type	M1: Score rated by agents	M2: Steps	M3: New structures	M4: New links	M5: Proposed structures	M6: Communicated links	M7: Score
AB_1A	0.0158*	0.0000*	0.0000*	0.0000*	0.0001*	0.0000*	0.0981
$AB_{1,2,3}A$	0.0003*	0.0001*	0.0000*	0.0000*	0.0001*	0.0000*	0.0960
$AB_{1,...,5}A$	0.0034*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.8281
$AB_{1,..10}A$	0.0031*	0.3773	0.0000*	0.0000*	0.0005*	0.0003*	0.7940

* p <0.05

Table 4 *p*-values for T2

Task type	M1: Score rated by agents	M2: Steps	M3: New structures	M4: New links	M5: Proposed structures	M6: Communicated links	M7: Score
BA	0.1025	0.0359*	0.0000*	0.0000*	0.2181	0.0927	0.2753
B^3A	0.0033*	0.0036*	0.0000*	0.0000*	0.0029*	0.0001*	0.3659
B^5A	0.0003*	0.0006*	0.0000*	0.0000*	0.0000*	0.0000*	0.6281
$B^{10}A$	0.0593	0.0072*	0.0000*	0.0000*	0.0000*	0.0000*	0.9867

* p <0.05

Table 5 p-values for cases of T2 where solution for task A were found while performing task B

Task type	M1: Score rated by agents	M2: Steps	M3: New structures	M4: New links	M5: Proposed structures	M6: Communicated links	M7: Score

BA	0.1007	0.0567	0.0020*	0.0005*	0.2110	0.1749	0.2853
B^3A	0.2858	0.0106*	0.0000*	0.0000*	0.0247*	0.0024*	0.4684
B^5A	0.0118*	0.0085*	0.0000*	0.0000*	0.0000*	0.0000*	0.7653
$B^{10}A$	0.7112	0.3785	0.0000*	0.0000*	0.0000*	0.0000*	0.9918

* p <0.05

Table 6 *p*-values for cases of T2 where no solutions for the task A were found while performing task B

Task type	M1: Score rated by agents	M2: Steps	M3: New structures	M4: New links	M5: Proposed structures	M6: Communicated links	M7: Score
BA	0.3306	0.0899	0.0004*	0.0002*	0.4004	0.1637	0.3932
B^3A	0.0154*	0.0544	0.0000*	0.0000*	0.0449*	0.0090*	0.2168
B^5A	0.0051*	0.0136*	0.0000*	0.0000*	0.0017*	0.0009*	0.2476
$B^{10}A$	0.0013*	0.0025*	0.0000*	0.0000*	0.0000*	0.0000*	0.4146

* p <0.05

Discussion

Analysis of performance on the task sequences of the type T1

As can be seen from the results presented in Tables 1 and 3, agents have rated solutions generated for task *A* significantly higher in the second turn comparing to the score proposed in the first turn, irrespective of the number of tasks performed in between turns. Furthermore, considerably fewer new structure domain nodes and links were learned, and significantly fewer structure domain nodes and links were shared. In other words, when they were faced with a familiar task, agents converged to the solutions faster, without the need to create novel structure domain nodes. Further, over the runs between two tasks *A,* more links between structure domain nodes and behavior domain nodes have been created. Some links to the proposed solutions became more grounded, i.e., agents became surer about proposed structure's behavior (i.e., associated network's properties), resulting in agents rating their solutions as better. However, when ten tasks were performed between two tasks *A*, the number of steps needed to find (and agree on) the solution was not significantly lower than in the first turn. Inspection of data revealed that this effect was caused by forgetting. Namely, while some of the links get grounded, the links which are not used in tasks between two tasks *A* decrease in weight, consequently resulting in more steps needed for the agents to activate relevant nodes and links.

It is interesting to note that a team's objective performance did not significantly increase. While the null hypothesis can be rejected at a significance level of 0.01 for the cases where there are one or three tasks performed in between tasks *A*, when five and ten tasks were performed in between, the null hypothesis cannot be rejected. From a closer inspection of the data, it was found that such behavior in some cases results from the acceptance of the structure domain nodes as the final solution, although they were not adequate solutions for the given task (i.e., solutions did not meet all of the requirements). In other words, while performing other tasks, the faulty links in agents' mental model became grounded, thus causing the agents to accept inadequate structure nodes and rate their performance highly. Such grounding originated from either two agents sharing the same wrong link and reassuring its correctness through communication, or if one trusted agent posited a wrong link, causing others to learn it.

It is possible that this situation could be avoided if agents were presented with an objective evaluation of solutions after each task. However, in a closed system, agents developed faulty mental models. This serves to show that, apart from sharedness of the mental modes among team members, the accuracy of such models is important. As stated by Badke-Schaub et al. [17] "*All members of a team can share some identical knowledge, but all of them might be wrong*."

Analysis of performance on the task sequences of the type T2

In the second type of task sequences, task *B* was chosen to have an overlap with the task *A*, in terms of having some function and behavior domain nodes in common. The overall results show a similar trend as seen in the case of the first task sequence type. However, it can be noticed that the score and number of structure domain nodes and links shared is not significantly different when a single task was performed prior to performing task *A*. The number of steps, number of new structure domain nodes created and number of new links learnt were all significantly lower when task *A* was performed after a related task. This implies that, although agents learned links related to task *A* and created several structure domain nodes while performing the task *B*, the knowledge relevant for task *A* neither became grounded, nor has it significantly decreased in weight. Therefore, the number of links and solutions proposed did not significantly differ from the case where agents performed only task A. In all other cases, agents reach the conclusion faster in terms of steps needed, and accordingly, number of newly created structures, links learned, and solutions and links shared. However, the score as determined by the team was not significantly better when ten tasks B were performed prior to task A. Further, while the objective score was not significantly lower for any of the cases, for the case where there are ten tasks B performed, the hypothesis that an actual shift of the score is towards negative can be accepted at significance level of 0.01. Close inspection of the data showed that when multiple performances of task B occurred, the weights of the links relevant for the task A were decreasing, while links important for the task B gained weight, causing agents to more often chose solutions appropriate for the task B rather than for the task A.

To further inspect whether the overlap between task A and task B impacts the results for the second task sequence type, two datasets where extracted: cases where suitable (objective) solutions for a task A were found by the agents while performing task B, and cases were no real solutions for task A were found while performing any of the tasks B. The first contains instances where a solution for the task A was found in almost every run, and the second contains data on the simulations where no solutions for the task A were found in any of the runs.

It is expected that, in the case where no solution for task A has been found, objective solution scores do not vary significantly between cases. When no solution for task A has been found, the score as rated by agents is significantly higher after several runs of task B indicating that some partial solutions have been found during the performance of task B. Further, in several cases, wrong links caused the agents to finish their search, thus resulting in lower number of steps required. However, as expected, the objective score for the solutions found did not differ significantly between any of the cases.

Regarding the instances where solutions for the task A were found while performing task B, the simulations where ten tasks B were performed are the most interesting. For such simulations, the hypothesis that an actual shift of the score is towards the negative was found supported at a significance level of 0.05. In other words, performing task B a large number of times decreases the probability of finding a correct solution for the task A, even though several possible solutions have been encountered throughout completing the tasks. This example demonstrates the detrimental effects overspecialization of teams can have on the performance of the tasks which fall outside of team's expertise area. On the individual level, team member's overspecialization has been demonstrated to have a detrimental effect on team performance as it related to de-skilling of individuals and the reduction in the common ground between team members [25], [26]. Research in [25] and [26] deals with the case where team members' expertise areas have *too little* in common. The simulated case demonstrates the other side of the spectrum: the case where team member's expertise areas have *too much* in common, i.e., the case of overspecialization on the team level. After multiple performances of a single task *B*, agents developed and grounded a shared knowledge related to the task *B* which enabled them to reach the solution efficiently, but the rest of their metal models remained unused and declined in weight which resulted in team's incapacity to find a solution when presented with a novel task. Therefore, as concluded in [17], there is a need for balance in shared and distributed knowledge within the team.

Conclusion

Team performance is necessarily influenced by knowledge held by its members. However, the relationships between member's mental models, team interactions, and team performance are still not adequately understood. This paper contributes to knowledge about the effect team experience has on team performance by utilizing computational simulations to demonstrate the increase in team efficacy as team members collaborate on multiple tasks. Further, by modeling designers as cognitively rich agents, the model enabled some insights into the cognitive behaviors of designers. The process of knowledge grounding and its effect on the team's convergence to the solution was studied, and findings match the existing cognitive theories, thus displaying the capability of agent-based models to aid research on design teams. However, additional studies are needed to further enhance understanding of co-evolution of team members' mental models and the mutual impact of tasks performance and team experience. New simulation runs are needed to fully verify the model and to study how variability across factors influences results. Future studies will be aimed at running the model in different scenarios, therefore providing additional insights in each of the modelled aspects. Further, additional experiments will provide deeper insights into the scale independence of results (regarding task and team size) and will focus on detailed studies of the influence of various factors on the team performance. For example, questions such as how do the results vary in relation to the level of overlap between tasks performed, how does the diversity (among members) and scope of agent's initial knowledge influence the overall performance, how does cohesiveness of the group affect the results and how does the rate of forgetting affect performance. Finally, what is the effect of task uncertainty, goal clarity and environmental turbulence on team processes and behavior should be further explored. It is important to note that these experiments dealt with stable teams (i.e. with clear boundaries and with no change in membership) which are rare in product development domain. Therefore, future research will concentrate on expanding these experiments by including multiple teams working on different project, and by simulating team members leaving, entering or returning to a team. Finally, the limitations in the task representation should be tackled. Particularly, product development teams are often faced with the requirement of novelty and the currently implemented agents have limited capability in dealing with such a requirement. Future research will be aimed at providing a richer task and team representation and enabling more realistic simulations.

The results from the experiments reported here suggest there is a detrimental influence of knowledge inaccuracies, forgetting and overspecialization on team performance. Additional experiments should further explore how much of the knowledge overlap is beneficial for the team, as well as what knowledge should be shared [17]. The current model could serve as a valuable experimentation tool in such studies as it enables simulating multiple different scenarios, thus producing large amounts of data which can be further studied by deep learning algorithms and potentially reveal additional patterns of design behavior.

Acknowledgements

This paper reports on work funded by Ministry of Science, Education and Sports of the Republic of Croatia, and Croatian Science Foundation MInMED project (www.minmed.org). This research is supported by a grant from the National Science Foundation to the third author under Grant CMMI-1400466. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Gero JS, Peng W (2009) Understanding behaviours of a constructive memory agent: A markov chain analysis. Knowl-Based Syst 22:610-621.

2. Light B, Butterworth G (2016) Context and Cognition: ways of learning and knowing. Routlege, London.
3. Gero JS, Kannengiesser U (2004) The situated function-behavior-structure framework. Des Stud 25(4):373-391.
4. Kannengiesser U, Gero JS (2012) A Process Framework of Affordances in Design. Des Issues 28:50-62.
5. Gero JS, Fujii H (2000) A computational framework for concept formation in a situated design agent, Knowledge-Based Systems 13(6):361-368.
6. Maher ML, Gero JS (2002) Agent Models of 3D Virtual Worlds. ACADIA 2002: Thresholds, California State Polytechnic University, Pamona 127-138.
7. Kahneman D (2011) Thinking, Fast and Slow. Farrar, Straus and Giroux, NY.
8. Gero JS (2017) Whither Design Theory and Methods?. Back to the Future: The Next 50 Years, ANZAScA 563-572.
9. Sivasubramaniam N, Liebowitz SJ, Lackman CL (2012) Determinants of New Product Development Team Performance: A Meta-analytic Review. J Prod Innov Manag 29:803-820.
10. Carbonell P, Rodriguez AI (2006) Designing teams for speedy product development: The moderating effect of technological complexity. J Bus Res 59:225-232.
11. Akgun AE, Lynn GS (2002) Antecedents and consequences of learning in new product development teams. J Eng Technol Manage 5:57-72.
12. Gruenfeld DH, Fan ET (1999) What newcomers see and what oldtimers say: discontinuities in knowledge exchange. Shared Cognition in Organizations: The Management of knowledge, Lawrence Erlbaum, Mahwah, 245-268.
13. Levine JM, Moreland RL (1999) Knowledge transmission in work groups: helping newcomers succeed. Shared Cognition in Organizations: The Management of knowledge, Lawrence Erlbaum, Mahwah, 267-296.
14. Choi HS, Thomson L (2005) Old wine in a new bottle: Impact of membership change on group creativity. Organ Behav Hum Dec 98:121-132.
15. Gibson CB, Gibbs CL (2006) Unpacking the concept of virtuality: The effect of geographic dispersion, electronic dependance, dynamic structure, and national diversity on team innovation. Admin Sci Quart 51:451-495.
16. Hirst G (2009) Effects of membership change on open discussion and team performance: The moderating role of team tenure. EJWOP 18:231-249.
17. Badke-Schaub P, Neumann A, Lauche K, Mohammed S (2007) Mental models in design teams: a valid approach to performance in design collaboration? CoDesign 3:5-20.
18. McComb C, Cagan J, Kotovsky K (2015) Lifting the Veil: Drawing insights about design teams from a cognitively-inspired computational model. Des Stud 40:119-142.
19. Ambler NP (2015) Design in the Modern Age: Investigating the Role of Complexity on the Performance of Collaborative Engineering Design Teams. PhD Thesis, Virginia Polytechnic Institute and State University.
20. Singh V, Dong A, Gero JS (2013) Social learning in design teams: The importance of direct and indirect communications. AIEDAM 27:167-182.

21. Kannengiesser U, Gero JS (2004) Modelling Expertise of Temporary Design Teams. J Des Research 4:1-13.
22. Sosa R, Gero JS (2013) The creative values of bad ideas. Proceedings of CAADRIA 2013 853-862.
23. Van den Bossche P, Gijselaers W, Segers M, Woltjer G, Kirschner P (2011) Team learning: building shared mental models. Instr Sci 29:283-301.
24. Perisic MM, Storga, M, Gero JS (2017) Building a computational laboratory for the study of team behaviour in product development. Proceedings of ICED'17 8:189-198.
25. Carley K, Gasser L (1999) Computational Organization Theory, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, 299-330.
26. Anderson EG, Lewis K (2014) A Dynamic Model of Individual and Collaborative Learning Amid Disruption. Organ Sci 25:356-376.