KEYWORDS product design complexity coordination agent-based model simulation

THE IMPACT OF COMPLEXITY DURING PRODUCT DESIGN

Xiaoqi Zhang
Mechanical Engineering,
McGill University
xiaoqi.zhang@mail.mcgill.ca

Vince Thomson
Mechanical Engineering,
McGill University
vince.thomson@mcgill.ca

ABSTRACT

Ever increasing product complexity is an obstacle to effective product development. This paper introduces an agent-based model to study the impact of complexity during product design. Product was modeled as a set of functions that require knowledge, design effort and integration, and designers were modeled as agents who applied knowledge to function development tasks and communicated with each other. Simulation experiments were conducted to study the impact of variables, such as: designers' knowledge level, designers' experience, coordination efficiency and organizational structure under different levels of product complexity. The results suggest that an increase in complexity increases effort and span time exponentially. Thus, for the development of complex products, more effective coordination mechanisms should be applied when a project has very high levels of complexity and innovation. Having more knowledgeable and experienced designers also helps to lessen design effort and shorten span time. No assertion could be made as to whether a team or matrix organizational structure was superior.

INTRODUCTION

Fast-paced technology and globalization have created a very competitive marketplace. As companies respond to these challenges by increasing product diversity, introducing new technology, and establishing partnerships, they are generally faced with continuously increasing complexity. Complexity, if not well managed, is a drag on team performance and can lead to project failure. During product design, greater risk of budget overruns, schedule slippage, and flawed quality has been observed as complexity rises (Braun & Lindemann, 2008). There is an urgent question about how to better manage projects given ever-increasing complexity.

Complexity in product development has the characteristics of multiplicity, interaction and uncertainty; multiple functions are developed and integrated into a whole product, and during the process, designers coordinate activities to resolve interfaces and to reduce uncertainty by acquiring more product information. The essence of product design is to perform successive iterations of possible solutions to a design problem, and then, to converge to a final, single solution. During the process, it is important to limit the number of tasks to a manageable level as a project unfolds in order to contain the instantaneous complexity. It is important to organize tasks such that the design solution evolves quickly without having the instantaneous complexity become so great that the design problem cannot converge and the project cannot end.

To manage increasing complexity, companies apply different coordination mechanisms or change organizational structure to reduce cost and cycle time. Gokpinar et al. (2010) studied coordination problems in product development by constructing and analyzing networks of product architectures and the organization of designers. The statistical study of vehicle programs showed a positive association between coordination deficit and quality problems. It suggested that lack of communication probably caused quality problems in products. Erhardt (2011) investigated teamwork in product development with a different nature of problems and team composition. During the case study, the author observed that teams with homogenous knowledge did not communicate much when dealing with well-structured problems, but needed a lot of teamwork when faced with ill-structured problems; teams with heterogeneous knowledge worked in a modular fashion in the situation of well-structured problems, but had extensive interaction when coping with ill-structured problems. The study implied possible impact of organizational structure and team knowledge on the ability of handling complexity. Although the above literature is of managerial interest, it did not give deeper insight into why organizational structure and coordination mechanisms affected project performance in the observed way. Also, the studies did not give any suggestion on how to improve team performance in projects with different levels of complexity.

Since the case-based approach is limited in exploration of various scenarios, is time-consuming, and involves too many random variables, many researchers used modeling and simulation to study product development. There are mainly three paradigms used in most computer models in the literature: System Dynamics, Discrete Event and Agent Based. System Dynamics requires a clear definition of cause and effect relationships between variables. Le et al. (2012) developed a System Dynamics model to study the impact of management levers on project lead time, where certain cause and effect relationships of the overall project were assumed. However, in studying the effect of product complexity on designer interactions, many of the cause-effect relationships are unknown. In fact, this is what the research in this paper tries to find out. Discrete Event modelling is process-centric. It represents the system as a sequence of operations being performed on entities. Suss (2011) modeled product development as processes that consisted of a series of tasks exchanging information and showed that appropriate communication levels were important to reduce span time and effort. Although Suss was able to model the communication between designers, the process-centric paradigm was limited in its flexibility to model individual interactions, which is important when studying different aspects of coordination. In order to provide a tool to determine which team characteristics best fit certain projects, Rojas-Villafane (2010) developed a team coordination model using the Agent-Based paradigm. Tasks were modeled as an activity network and team members were agents working on tasks. The Agent-Based paradigm proved to be useful in modeling the coordination between team members. Although Rojas-Villafane applied the tool well to optimize the team for a sailboat race, the author did not investigate the relationship of coordination mechanisms and product complexity.

Therefore, with the goal of investigating what coordination mechanisms and team structure to adopt in a specific, complex situation, the research in this paper addresses the following research questions:

- Given a certain level of complexity, how can different coordination mechanisms affect team performance?
- Given a design team using certain coordination mechanisms, how much complexity can be handled?
- When product complexity increases due to design change, how should coordination mechanisms be

changed, so the design team can still achieve project objectives?

A research model of product development was created to study the above questions. It was built with the Agent-Based paradigm, as it allowed variations in designer characteristics and in the interactions between designers. It also permitted the study of the effect of different coordination mechanisms on individuals. This model can track instantaneous complexity and dynamic effort during the design process. If dynamic complexity grows at too great a rate, the design problem cannot converge, or if the total effort to date is excessive, project objectives may not be achieved or the project could be cancelled. Simulation provides insight into determining the reasons that complexity is growing and what mechanisms can be used to improve project performance.

The paper is organized as follows: section 1 describes the model; section 2 describes the design of simulation scenarios and the results; section 3 discusses the results; section 4 summarizes the paper and gives conclusions.

1. Modeling Product Development

The model is designed with knowledge as the bridge between product, development process and organizational structure. The product is composed of functions, which require knowledge and integration. On the other side, designers should have the corresponding knowledge to design these functions, such that designers apply their knowledge to develop functions and communicate with others to resolve integration issues. How designers are grouped is determined by organizational structure.

Product can be represented as a function tree. In the model, functions are agents requiring knowledge and effort to create or integrate them. A function agent neither makes decisions nor acts on his own initiative; instead, a function agent can only respond passively to designer agents' actions. Designers are modelled as agents with different knowledge and experience aimed at finishing tasks such as function design, rework and functional integration. They have the autonomy to decide when to start, change or end activities such as the development of a function, consultation or coordination with another designer. As complexity grows, designers spend more time on consultation, coordination and rework, and if the rate of growth of complexity is too great, the amount of work continues to grow, and complexity inhibits a project from finishing.

Different organizational structure, designers' experience, and application of coordination mechanisms influence the effectiveness and quality of communication between designers, which further determines a designer's decisions. Following the assumptions and decision strategies, each designer works on assigned functions and communicates with others. This generates a designer's global behaviour during product development, where change in effort and project duration versus different product complexity, organizational structure, knowledge intensity, and integration complexity can be analyzed.

1.1 Characterization of Complexity

Product development is substantially a knowledge-intensive process. Nissen and Levitt (2002) reflected on product development: "all of the requisite knowledge must come together and be integrated before the knowledge-intensive work can be completed". Tomiyama et al. (2007) found knowledge structure to be the fundamental source of complexity. They claimed that modern products were multi-disciplinary whereas designers were usually expert in one discipline, which increased product and process complexity. Knowledge is the link among product, process and organization, as knowledge is embedded into product functions, which require designers with corresponding knowledge to create them. Complex processes further force organizations to make adaptations for better coordination. Thus, product complexity can be quantified in the knowledge perspective.

In the model in this paper, the set of knowledge used in design projects is classified where the levels of knowledge required for designing a function are defined with a scale: none, simple, medium and high. Correspondingly, designers' level of ability to apply knowledge is also defined with the scale: none, basic, professional and experienced. A numerical value is used for each level where the lower limit is 1 and the upper limit is *r*, which is determined by the user.

Two types of complexity, function complexity and integration complexity, are defined. Function complexity is the complexity of developing a certain function in the product. It is assumed that the more knowledge required for a function, the more complex the design. Thus, function complexity is calculated as the root mean square of the values of the knowledge involved in a function. Since the knowledge requirement was quantified as in range [1, r], the function complexity is in the range [1, r]. Integration complexity is the complexity of integrating functions into a whole product. It is assumed that the integration of two functions becomes more difficult with more interfaces and with the involvement of different knowledge. Thus, the integration complexity between two functions is calculated as the product of the number of interfaces and knowledge difference. To quantify knowledge difference, we represented a function as a vector in the knowledge space and used the intersection angle θ of two knowledge vectors as an indicator of their difference regarding knowledge content. To keep the value consistent with function complexity in terms of the order of magnitude, we used $r^{\sin\theta_{ij}}$ to measure the knowledge difference between function *i* and function *j*. The knowledge difference is in range [1, *r*] and the number of interfaces is in range [0, +∞]; hence, as the product, the integration complexity is in range [0, +∞].

Therefore, when product is represented as a functional tree, functions can be characterized from the knowledge perspective in terms of the function complexity, knowledge difference and integration complexity.

1.2 Designers Developing Functions

Designer agents are assigned function development tasks and they select one of the assigned tasks that have all the inputs ready. More experienced designers are generally more efficient doing technical work. For example, experienced designers develop simple functions faster and newbies develop difficult functions more slowly. So, it is assumed that a designer's working efficiency increases with his level of knowledge and decreases with a function's requirement for knowledge. The efficiency of designer *p* developing function *i* is determined by Equation 1.

$$\mu_{p,i} = \frac{1}{m_{p,i}} \sum_{k \in M_{p,i}} \frac{KA_{p,k}}{KR_{i,k}} \qquad \text{Equation 1}$$

– $KA_{p,k}$ is the ability level of engineer *p* applying knowledge *k*;

– $KR_{i,k}$ is the requirement level of function *i* for knowledge *k*;

– $M_{p,i}$ is the set of knowledge that engineer *p* is able to apply to function *i*;

– $m_{p,i}$ is the number of elements in the set.

When designers work on a function, they are turning "work to do" into "work completed". At the beginning of a function, "work to do" has all the requirements and the estimated effort. Once there are designers working on a function, the effort flows from "work to do" to "work completed". Flow is the sum of the efficiencies of all designers who are developing a function ($\sum_p \mu_{p,i}$, where $\mu_{p,i}$ is determined by Equation 1). When all effort has flowed from "work to do" to "work completed" and there is no unfinished rework, the function is considered completed and is ready for integration.

The designer keeps developing a selected function until

1. the function is completed;
2. or a designer wants to perform another task; this models the situation where a designer is assigned to several, different function development tasks. In this situation, a designer does not stick to one task until it is finished, but switches between different tasks as new information becomes available.
3. or a designer wants to communicate with another designer; this models the situation where a designer needs to consult with regard to the design of a function or needs help to coordinate activities.

1.3 Communication Methods

In the model, design agents may communicate for coordination or consultation. The communication method is either synchronous or asynchronous. For easier explanation, here we call the designer who starts the communication the asker and the designer who responds, the answerer. The asker makes decisions on the communication method (asynchronous or synchronous) and communication content (coordination or consultation). The answerer's action varies with the communication method used by the asker. It is assumed that one can only respond to a synchronous request with a synchronous answer, and an asynchronous request with an asynchronous answer.

An asynchronous communication is modeled as one round of question and answer. The asker creates an asynchronous question message, which is put into the answerer's message list. When the answerer checks the message list, the answerer reads the asker's message and creates an asynchronous answer, which is put into the asker's message list. When the asker checks for messages, the asker spends time reading the answer's message and finishes this round of communication. Examples of asynchronous communication are emails, voice messages, etc.

Different from asynchronous communication, which does not interrupt the answerer's work, synchronous communication causes the answerer to stop in order to respond immediately, except if the answerer is involved in another synchronous communication. The asker and the answerer simultaneously begin and end a synchronous communication. Examples of synchronous communication are face-to-face communication, telephone, etc.

1.4 Consultation

Consultation in the model is defined as communication between two designers concerning certain knowledge involved in technical work. It is assumed that when the knowledge requirement of a task is beyond a designer's ability, the designer consults with a more experienced designer about required knowledge. The decision strategy is determined by the knowledge gap of the requirement, the expert's ability and the designer's ability. Once the designer devotes enough effort towards consultation, the knowledge gap is closed and the designer does the technical work with higher efficiency.

1.4.1 Decision Making of Consultation

It is assumed that if a function's requirement for knowledge is beyond a designer's ability, the designer tends to go and consult, and the less adequate a designer's knowledge, the more likely consultation occurs.

The difficulty of a function to a designer is quantified by the concept of knowledge gap. For each knowledge item, there is a knowledge gap between the function and the designer. The knowledge gap is determined by Equation 2.

$$Gap_{pik} = \frac{R_{i,k}}{A_{p,k}} \quad \text{Equation 2}$$

– $R_{i,k}$ is the requirement level of function i for knowledge k;

– $A_{p,k}$ is the ability level of engineer p to apply knowledge k.

If the gap is less than 1, it means this knowledge item required in the function is easy for the designer; if the gap equals 1, it means the knowledge required in the function is just right for the designer; if the gap is greater than 1, it means the knowledge required in the function is difficult for the designer.

The likelihood to consult is quantified by the concept of knowledge gap tolerance (*TK*). The decision to consult is triggered only when a designer's knowledge gap in the function exceeds *TK*, i.e., the condition, $Gap_{pik} > TK$ is true.

After a designer decides to consult on knowledge *k*, the designer searches for consultant designers whose knowledge *k* is higher than or equal to the requirement of the function ($A_{cnslt,k} \geq R_{i,k}$). If there are many consultants, the designer randomly picks one of them. If no one satisfies the condition, the designer stops looking for a consultant, and does the work at the designer's own pace.

1.4.2 Effect of Consultation

It is assumed that a designer's ability to apply certain knowledge improves after consulting a more experienced designer. Generally speaking, the effect of consultation is better when more consultation effort is given and communication is more efficient. However, consultation efficiency can be impaired by the complexity of the function. Although a designer's ability to apply knowledge improves, the improvement is constrained by the consultant's knowledge level, i.e., a designer's improved ability cannot exceed the consultant's level of knowledge. Thus, a designer's improved ability can be quantified by Equation 3.

$$A_{p,k}^{New} = A_{cnslt,k} - (A_{cnslt,k} - A_{p,k}^{Old}) \cdot e^{-\left(\frac{efficiency}{Complexity_i}\times effort\right)} \quad \text{Equation 3}$$

– $A_{p,k}$ is the designer's ability of knowledge k; $A_{p,k}^{New}$ is the improved ability after the consultation; $A_{p,k}^{Old}$ is the ability before the consultation;

– $A_{cnslt,k}$ is the consultant's ability of knowledge *k*;

– *effort* is the consultation effort;

– $Complexity_i$ is the knowledge that is calculated as the root mean square of function *i*'s knowledge requirement;

– *efficiency* is the consultation efficiency and can be interpreted as the knowledge understood per hour.

Therefore, a designer's knowledge improves faster with a more experienced consultant, and a designer cannot surpass the consultant in terms of the consulted knowledge. Also, if a designer already has a high level of knowledge before consultation, it takes less effort to reach a certain improved level. The direct effect of consultation is the improvement of a designer's ability using knowledge *k*. As a result, a designer works more efficiently on functions that require the knowledge.

1.5 Coordination

Coordination in the model is defined as the communication between two designers who are developing different functions that are to be integrated. It is assumed that when designers are aware of a function's dependencies, they coordinate with other designers who are developing relevant functions. The decision strategy is determined by the integration complexity. The probability of rework for functions that have been coordinated is reduced by the effort spent on coordination.

1.5.1 Decision Making of Coordination

It is assumed that designers tend to underestimate the interdependency among functions, especially when designers do not have much experience in the project. Thus, a parameter *Experience* is defined to model a designer's understanding of function interdependency. *Experience* determines how well designers are aware of the integration complexity between functions. *Experience* increases with more coordination, which is discussed in detail later. A designer's perceived integration complexity between two functions is a fraction of the true integration complexity and is determined by Equation 4.

$$pcvI_{p,ij} = Experience_{p,ij} \cdot I_{ij} \quad \text{Equation 4}$$

– $pcvI_{p,ij}$ is designer p's perceived integration complexity of function *i* and function *j*;

– $Experience_{p,ij}$ is designer *p*'s certainty of the integration required by function *i* and function *j*, range [0, 1];

– I_{ij} is the true integration complexity of function *i* and function *j*; it is the product of the knowledge difference and the number of interfaces between the two functions as introduced in section 1.1.

If a designer doubts that a function can be integrated well with another function, a designer has a tendency to seek coordination. The higher the integration complexity of two functions, the more likely it is that a designer seeks coordination. The more effort designers spend coordinating two functions, the more confident the designers are about the integration; thus, there is less likelihood of seeking more coordination. A designer's doubt of handling integration complexity well is quantified as Equation 5.

$$doubt_{p,ij} = pcvI_{p,ij} - effort_{p,ij} \cdot efficiency \quad \text{Equation 5}$$

– $pcvI_{p,ij}$ is the integration complexity between function i and function j that designer p perceives;

– $effort_{p,ij}$ is the effort that designer p has spent on coordinating function i and function j in hours;

– *efficiency* is the integration complexity understood per hour.

The likelihood of seeking coordination is quantified by the concept of integration complexity tolerance (*TI*). The decision to seek coordination is triggered only when a designer's doubt about integration exceeds *TI*, i.e., the condition $doubt_{p,ij} > TI$ is true. After designer *p* decides to coordinate with function *j*, the designer that is assigned to function *j* is randomly picked.

1.5.2 Effect of Coordination

It is assumed that a designer's understanding of the integration complexity of two functions, i.e., experience, improves after coordinating the two functions. Similar to consultation, the improvement is better when more coordination effort is given and the communication is more efficient. Coordination efficiency is impaired by the integration complexity between the two functions. A designer's experience of understanding two functions is determined by Equation 6.

$$Experience_{p,ij} = 1 - (1 - expInitial_{p,ij}) \cdot e^{-\left(\frac{efficiency}{I_{ij}}\times effort_{p,ij}\right)} \quad \text{Equation 6}$$

– $expInitial_{p,ij}$ is a designer's initial understanding (experience) of function *i* and function *j*, range [0, 1];

– *efficiency* is the coordination efficiency and is interpreted as the integration complexity understood per hour;

– $effort_{p,ij}$ is the effort that designer p has spent on coordinating function *i* and function *j* in hours;

– I_{ij} is the true integration complexity of function *i* and function *j*.

Therefore, a designer's experience improves faster with greater communication efficiency. Also, if a designer already has high experience before consultation, it takes less effort to reach a certain improved level. A designer's experience improves faster with lower integration complexity. The direct effect of coordination is the improvement of a designer's understanding of the integration complexity between functions. As a result, a designer's perceived integration complexity is more accurate, and the designer is able to conduct coordination in a more timely way.

1.6 Rework Generation

When designers coordinate or integrate two functions, they may find incompatibility between the two functions, which generates rework. The percentage of work that needs to be revised decreases with better understanding of the integration complexity between functions. The average experience of all designers involved in the development of the two functions is used as the indicator of the global understanding of the integration complexity between the two functions. Thus, the percentage of work that needs to be redone is determined by Equation 7.

$$rework_{ij} = 1 - \frac{1}{d_{ij}} \cdot \sum_{p\in D_{ij}} Experience_{p,ij} \quad \text{Equation 7}$$

– $Experience_{p,ij}$ is designer *p*'s experience in function *i* and function *j*;

– D_{ij} is the set of designers that are involved in the development of function *i* and function *j*;

– d_{ij} is the number of designers in the set D_{ij}.

2. Simulation Experiments

A series of experiments using the product development model revealed that when complexity became too great as dynamic effort or instantaneous complexity increased, a change in coordination mechanisms could reduce the growth rate of instantaneous complexity such that a project was brought back under control. Level of knowledge, organizational structure and coordination were mitigating factors, which could improve performance.

2.1 Data

2.1.1 Product Data

The functional data about a hydro-electric generator from GE Hydro was obtained from Bashir and Thomson (2004). The product was decomposed into five major functions – control environment, provide housing, provide monitoring, provide safety and control power. The five major functions were further decomposed into sub-functions, which gave 57 functions in total.

The knowledge involved in the product design was obtained by consulting the personnel who worked on the project. Ten types of knowledge were identified, including HVAC (heating, ventilating and air conditioning), air circulation, water circulation, heat transfer, electric-heat generation, control, mechanical engineering, sensor technology, physics and electrical engineering. Three levels were assigned to the knowledge items: minimum (numerical value = 1), general (numerical value = 2) and intense (numerical value = 3). The knowledge requirement for each function was known. For example, the environment control system required knowledge related to heat transfer; additionally, in order to provide housing knowledge of mechanical, control and heat transfer was required.

Using the knowledge required by each function, we were able to calculate the function complexity for each function. For example, the function "remove heat", a sub-function of "environment control", requires general knowledge of six knowledge items including HVAC, air circulation, water circulation, heat transfer, electric-heat generation, and control as well as minimal knowledge on the other four types of knowledge. Thus, function complexity is calculated as

$$Complexity_{remove\ heat} = \sqrt{\frac{2^2+2^2+2^2+2^2+2^2+2^2+1^2+1^2+1^2+1^2}{10}} \approx 1.7$$

The function complexity was an input to the model (Equation 3). Effort required to complete the technical work of each function was estimated based on the function complexity.

The integration complexity was also calculated. For example, it was known that the function "cool air" and the function "circulate air" were to be integrated. "Cool air" required general knowledge of electric-heat generation, control, mechanical engineering and sensor technology and minimum knowledge of the others. "Circulate air" required general knowledge of heat transfer, control, mechanical engineering and sensor technology and minimum knowledge of the others. Thus, using the method introduced in section 2.1, the intersection angle was calculated as

$$\sin\theta_{cool,circulate} = \sqrt{1 - \frac{(1,1,1,1,2,2,2,2,1,1)\cdot(1,1,1,2,1,2,2,2,1,1)}{\|(1,1,1,1,2,2,2,2,1,1)\|\cdot\|(1,1,1,2,1,2,2,2,1,1)\|}} \approx 0.2132$$

Since the upper limit of the knowledge scale was 3, the knowledge difference between the two functions was calculated as $3^{\sin\theta_{cool,circulate}} \approx 1.3$. It was known that there was one interface between the two functions; so, the integration complexity was $I_{cool,circulate}$. The integration complexity was an input to the model (Equation 6).

Therefore, the information of product functional tree, knowledge requirement, effort estimation and function complexity for each function, as well as integration complexity between each pair of functions, was obtained and provided as inputs to the model.

2.1.2 Designer Data

The data concerning the designers in the project was unknown; consequently, several different scenarios were designed and experimented. Twenty designer agents were created in the model to develop the product. Each designer was expert (numerical value = 3), proficient (numerical value = 2) or unskilled (numerical value = 1) for a certain knowledge. As a baseline, the designers were initially set as having sufficient knowledge to complete the development tasks, i.e., their knowledge matched well with the knowledge requirement of the functions. Also, they were initially identified as having medium experience (70%) about the project. The assignment of tasks varied with different organizational structures.

2.2 Simulation Scenarios

The objective of the simulations was to investigate the performance of projects when different mechanisms were applied under different product complexity. By examining the chart of dynamic effort obtained from each simulation, we were able to tell how the complexity unfolded with time. The chart also gave the total effort and project span time, which indicated project performance.

In order to observe project performance under different levels of complexity, we set the complexity level obtained from product data as the baseline and obtained a higher level of complexity by increasing the integration complexity by 50% and a lower level of complexity by decreasing the integration complexity by 50%. For example, the integration complexity between "cool air" and "circulate air" was calculated as 1.3 from the product data; so, their integration complexity was 1.3 in the baseline situation, and became 2 in the high complexity situation and 0.7 in the low complexity situation.

Mechanisms of interest included: a) knowledge level of designers, b) experience level of designers, c) coordination efficiency, and d) organizational structure. Three levels were set for the first three mechanisms and two levels for organizational structure.

A. **Designers' knowledge level.** The level obtained from designer data was set at medium. A high level of designers' knowledge was obtained by increasing designers' knowledge level by 0.5 (but not exceeding 3, the upper limit) and a low level of designers' knowledge was obtained by decreasing designers' knowledge level by 0.5 (but not lower than 1, the lower limit).

B. **Experience in project.** The baseline was 70%, which meant that designers were aware of about 70% of the integration knowledge between functions. High experience was set at 95%, which meant that designers were familiar with almost all the integration knowledge. Low experience was set at 45%, which meant that designers knew less than half of the integration knowledge.

C. **Coordination efficiency.** The baseline was set at 0.5, which meant that 0.5 units of integration complexity could be resolved in one hour of coordination. The high level was set at 0.8 and the low level at 0.2.

D. **Organizational structure.** Two organizational structures, team and matrix, were used for the designers. In the team structure, designers were divided into five teams, each in charge of a major function of the product. Designers could only work on the sub-functions of the major function that was assigned to their team. In the matrix structure, designers were grouped based on their capability of knowledge. Designers could work on many different functions as long as their knowledge satisfied the requirement. The team structure was set as the baseline.

Simulated scenarios are summarized in **Table 1**. The baseline was the scenario in which all mechanisms were at a medium level and designers were organized as teams. We studied the mechanisms one at a time, but not their combined effect; so, thirty scenarios were simulated.

Mechanisms		Product complexity: Low	Medium	High
Designers' knowledge	Low			
	Medium			
	High			
Designers' experience	Low			
	Medium			
	High			
Coordination efficiency	Low			
	Medium			
	High			
Organizational structure	Team			
	Matrix			

TABLE 1. *Simulation scenarios*

3. Result and Discussion

During the simulation, designer agents who were communicating or working on functions were considered to be giving effort, which the model captured, and then, generated the chart of instantaneous effort during the entire project.

Figure 1 shows the effort versus project time for the development of a product from GE Hydro, where complexity is low, medium or high. The horizontal axis is project time in working hours and the vertical axis is the number of designers contributing effort at any point of time. In the case of low complexity, dynamic effort reduces continuously, which indicates that complexity is continuously being reduced and that the design problem is converging quickly. Thus, the project finishes in a short time. In the case of medium complexity, dynamic effort decreases at first, increases significantly in the middle of the project, and then, decreases towards the end. When complexity is high, more frequent and large increases in effort can be observed. This causes the scenario to have a much greater span time than the low or medium complexity case. The increase in effort in the medium and high complexity cases is due to the greater effort required to resolve issues between interdependent functions.

Overall project performance can be indicated by total effort and span time. The area below a curve gives the total effort in person-hours and the point where the curve stops shows the span time in working hours.

Dynamic effort was compared when the levels of coordination mechanisms varied. **Figure 2** shows an example with three scenarios when mechanisms with different levels of efficiency are applied to designers who have moderate experience and knowledge, and are developing a product with high complexity. In the case of high coordination efficiency, dynamic effort decreases almost continuously during the project. When increases happen, they are brought back quickly. In the case of medium coordination efficiency, the project takes longer to converge and more, larger increases in effort happen. In the case of low coordination efficiency, dynamic effort decreases much more slowly and in many places increases for a period of time, especially in the late phases of the project, which indicates that the design problem may not converge within schedule.

Figure 3 compares the effect of different factors on effort and span time under different levels of product complexity. In the baseline situation, designers form teams to develop a product with medium complexity; they have moderate experience, sufficient knowledge and coordinate with medium efficiency. All the other scenarios are compared with the baseline situation

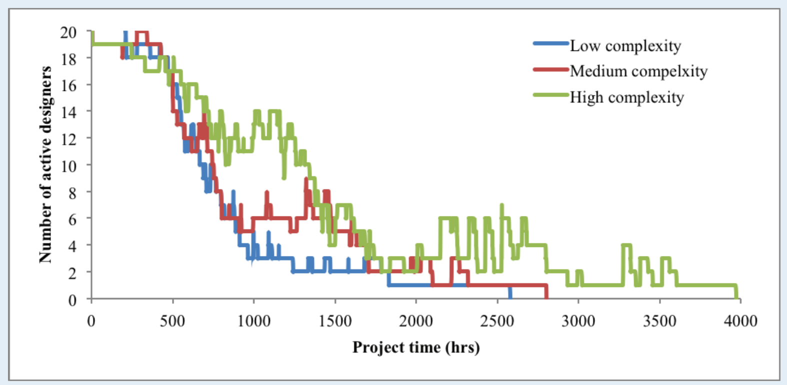

FIGURE 1. *Effort versus project time for the development of a complex product, where complexity is low, medium or high. The data were created by modelling the activity during a multiyear project at GE Hydro.*

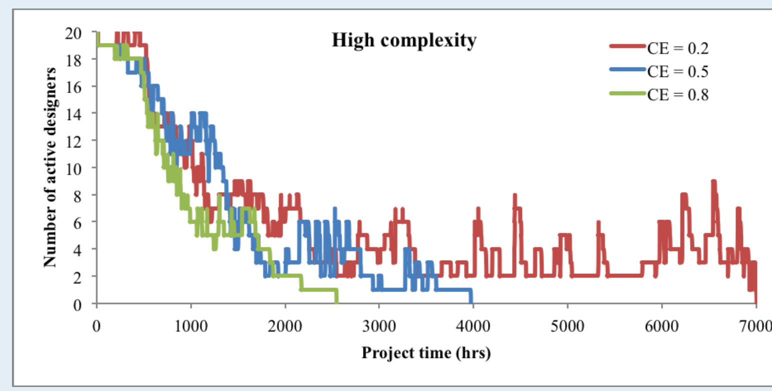

FIGURE 2. *Dynamic effort with different coordination efficiency (CE) for high product complexity*

regarding effort and span time. The first column shows the change of effort and the second column shows the change in span time. Different rows show the scenarios varying with the different factors.

A. Effect of designers' knowledge level. A higher level of designers' knowledge helps to reduce total effort and to shorten project span time, no matter whether the product complexity is low, medium or high. A lower level of designers' knowledge increases effort, but does not necessarily prolong the project. This can be explained by the fact that other constraints such as coordination efficiency and organizational structure have a large effect on project span time.

B. Effect of experience during the project. The factor – initial experience – shows the innovation level of the project. The chart indicates that the more innovative the project is, the more sensitive it is to increasing complexity. When developing a more complex product, a highly experienced design team can control effort and span time almost to the same level as for the development of a low complexity product.

C. Effect of coordination efficiency. In the case of low and medium complexity, higher coordination efficiency does not show obvious improvement in project performance. This implies that a medium level of coordination is sufficient for the development of a product with lower complexity. Compared to the situations with low and medium complexity, a product with high complexity is more sensitive to the effect of coordination efficiency: both effort and span time vary greatly with this factor. When coordination is very efficient, high product complexity is no longer a severe problem for designers. Effort and span time can be kept at the same level as developing a product with medium complexity.

D. Effect of organizational structure. In all three cases of complexity, a matrix organizational structure increases the total effort and slightly shortens span time. When designers are organized as teams, they are more focused on the set of functions that are assigned to them; therefore, less communication effort is required outside of the team. In a matrix organizational structure, since designers are usually assigned with multiple tasks, they tend to communicate more with other designers. There is also more switching between tasks, i.e., when a task is not ready for development because of the lack of input information, a designer does another task. Thus, more effort is spent, but shorter span time is observed when designers are organized as a matrix. However, we cannot assert which organizational structure is superior, as the effect of organizational structure also depends on product structure, i.e., whether the product is very modular or has highly interdependent functions.

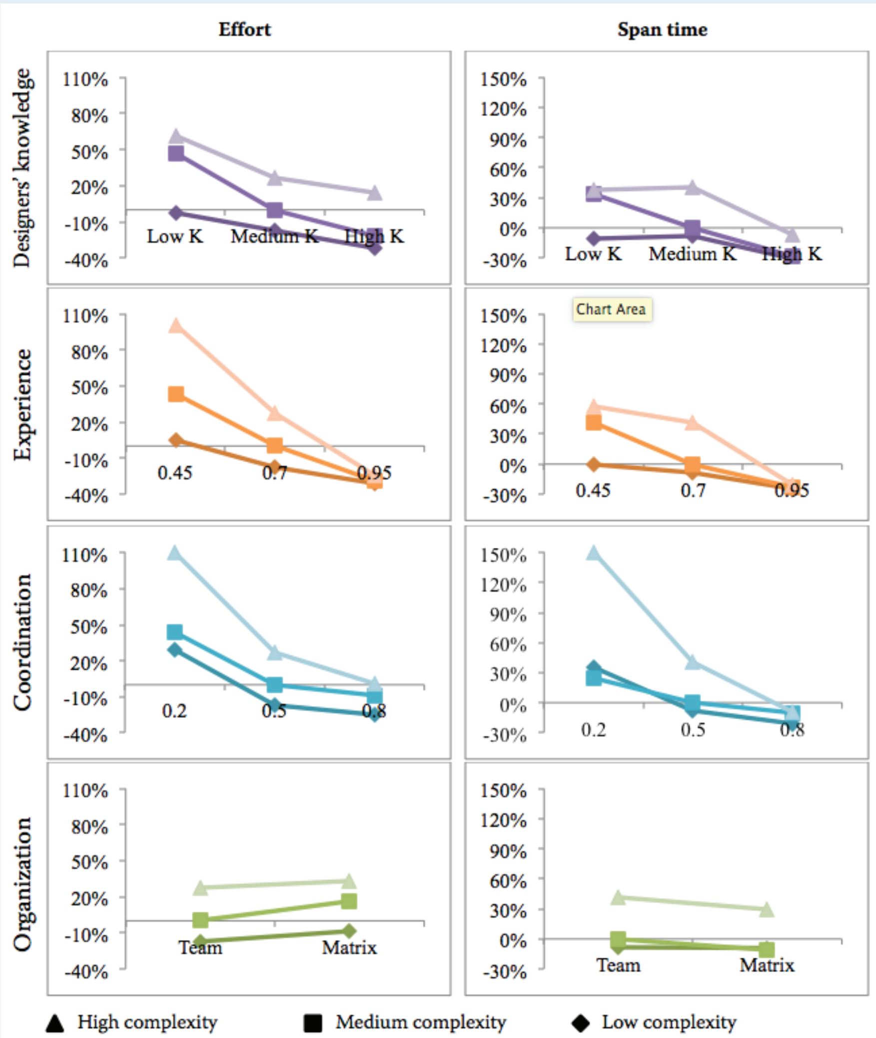

FIGURE 3. *The effect of different factors on effort and span time under different levels of product complexity. Effort and span time are y-axis results, and knowledge, experience, coordination and organization are x-axis variables.*

To summarize, increasing complexity requires more effort and prolongs the project, especially for innovative projects on which designers have little experience. More knowledgeable designers using efficient coordination save effort and shorten project span time. This effect is more obvious when developing a highly complex product.

4. Conclusions

Product design was modeled using an agent-based paradigm. The model allowed variations among designer agents based on experience and knowledge. Interactions between designer agents where communication had different efficiencies were also modelled. By consulting, coordinating and performing technical work, designer agents contributed effort during assigned tasks. Instead of modeling the effect of management mechanisms on a project directly, the model looked at the effect knowledge, experience and coordination on individuals and determined the resulting effect on span time and effort. Using product data from GE Hydro, the paper simulated a series of scenarios to investigate the effect of designers' knowledge, experience, coordination efficiency and organizational structure on project performance. Simulation results suggested that when product complexity increased, effort and span time increased. In addition, at high complexity, effort and span time increased exponentially, where the increase in effort was great and occurred towards the middle and end of a project. This increase in effort was caused by interdependency between functions. This increase in effort may not be obvious at the beginning of the project, and may be a surprise to design teams when it occurs.

Results also showed that more efficient coordination mechanisms should be applied especially when a project was highly complex and innovative in order to reduce or control the potential large increase in effort and span time. It was seen that knowledgeable and experienced designers did help to save design effort and shorten span time.

A matrix organizational structure increased the total effort and slightly shortened span time compared to using a design team structure. The difference between using either organizational structure was small; therefore, no assertion could be made as to which organizational structure was superior in all situations. The relative effectiveness of either structure needs to be studied further since other factors can have a great effect on effort and span time.

In future work, research will focus on the following features.

- Communication network. In the current model, any designer agent can communicate with any other designer. However, designers usually have limited access to other designers. This has an impact on the investigation of organizational structure. The communication network of designer agents will be varied and more organizational structures will be studied.
- More types of communication. The current model only used one-to-one communication. However, besides one-to-one communication, designers use other communication methods such as group meetings, databases, etc. More types of communication will be added to the model; so, the effect of availability of certain communication channels will be studied.

authors

Xiaoqi Zhang is pursuing her Ph.D. in the Department of Mechanical Engineering at McGill University. Her thesis topic is the measurement and mitigation of complexity during product design. During her Masters degree at McGill University, she worked on a research project on the value optimization of engineering jobs at Pratt & Whitney Canada in order to efficiently schedule engineering tasks. She has also worked with Bombardier Aerospace to find effective solutions for shortening product development span time. In 2014, Xiaoqi Zhang was part of a prize-winning team for the redesign of a manufacturing process for Schluter Systems.

Vince Thomson is a Post-Retirement Professor in the Department of Mechanical Engineering, McGill University. He has been involved in manufacturing and knowledge process-related research for the past 35 years at McGill University and the National Research Council (Canada). His research interests include manufacturing, realtime control and process management. In process management, Dr. Thomson has developed fundamental principles of process, methods for process improvement, metrics for monitoring processes, and information technology tools to support process performance. Partners include a wide variety of industries including aerospace, telecommunication, steelmaking and healthcare.

references

Bashir, H. A., & Thomson, V. (2004). Estimating design effort for GE hydro projects. Computers & Industrial Engineering, 46, 195-204.

Braun, S. C., & Lindemann, U. (2008). The influence of structural complexity on product costs. IEEE International Conference on Industrial Engineering and Engineering Management.

Erhardt, N. (2011). Is it all about teamwork? Understanding processes in team-based knowledge work. Management Learning, 42, 87-112.

Gokpinar, B., Hopp, W. J., & Iravani, S. M. R. (2010). The impact of misalignment of organizational structure and product architecture on quality in complex product development. Management science, 56, 468-484.

Le, H. N., Wynn, D. C., & Clarkson, P. J. (2012). Impacts of concurrency, iteration, design review, and problem complexity on design project lead time and error generation. Concurrent Engineering, 20, 55-67.

Nissen, M., & Levitt, R. (2002). Toward simulation models of knowledge-intensive work processes. CIFE Working Paper, 77.

Rojas-Villafane, J. A. (2010). An agent-based model of team coordination and performance. Ph.D, Florida International University.

Suss, S. (2011). Coordination in complex product development. Doctor of Philosophy, McGill University.

Tomiyama, T., D'Amelio, V., Urbanic, J., & ElMaraghy, W. (2007). Complexity of multi-disciplinary design. CIRP Annals - Manufacturing Technology, 56, 185-188.