Exploring Novice Approach to Conceptual Design of Software

Lakshmi T.G., Indian Institute of Technology Bombay, tglakshmi@iitb.ac.in
Sridhar Iyer, Indian Institute of Technology Bombay, sri@iitb.ac.in

Abstract: Engineering graduates are expected to design solutions to solve real world problems. While experts have body of knowledge and heuristics to arrive at solutions, novices do not have such knowledge and hence find it difficult to design solutions. In order to determine the nature of difficulties faced by novices, we conducted a study where novices were required to come up with a conceptual design for a given software design problem. We used the Function-Structure-Behaviour (FSB) lens to analyze the artefacts and the processes. Results from the analysis show that novices were unsuccessful when they fixate to either one or all FSB elements initially. These results inform the pedagogical strategies for teaching learning of conceptual design in the context of software.

Keywords: software, conceptual design, novices, function-structure-behaviour

Introduction

Conceptual design is one of the critical steps in engineering design (Pahl & Beitz, 2013). Conceptual design activity is described as a process in which the functional requirements of the design problem are transformed into descriptions of solution concepts (Chakrabarti & Bligh, 2001). Software design even though has a lot of common activities with other design domains (Cross et al, 1996); the dynamic and intangible nature of software however, poses a lot of challenges. Novices often have limited knowledge structures and experience that professional software designers have gained by experience. Novices might find designing solutions for open-ended problems daunting. Novices' difficulties can drive design of interventions and pedagogic strategies. In this paper we report a study conducted to understand novice's difficulties during the conceptual design activity in the context of software design.

Theoretical lens

The overarching theoretical lens that we chose to understand the novice conceptual design processes is Function-Structure-Behavior (Gero, 1990; Goel et al 2009). Function-Structure-Behaviour (FSB) answers – what the system does (F), what are all required to achieve the function (S), how is the function achieved (B). We used the FSB framework to analyze artefacts and identify absence of cognitive strategies/mechanisms in novices such as mental simulation, abstraction, association. These cognitive strategies have been reported to be utilized by expert software designers (Ball et al, 2010).

Methodology

The broad research question guiding this study was, 'What are the difficulties novices face while creating a conceptual design? '.

Participants and problems

The study spanned over two days and had five final year computer-engineering under-graduates as participants. The participants (par1-5) were exposed to software design courses as a part of their curriculum. So they had pre-requisite knowledge for the task. The participants were given a choice of problems: (i) Design a finger print ATM system (par2), (ii) Design a mood based automatic player (par1 & par3), (iii) Design a finger print based payment system (par4 & par5) and (iv) Design a cooking recipe recommender system. All the problems are equivalent in terms of complexity of time and cognitive skills.

Data source and analysis

Each participant was required to select one of the above problems and create a conceptual design. To accomplish the task the participants were provided with laptops with internet connection, paper and pen. While the participant was on the task, video recording of the activity and screen capture of their interaction with laptop was captured. Post task the participant was also interviewed. The data source included the video recording of the activity, screen captures, participant generated artifacts and the interview transcripts.

To analyze the video recording the protocol analysis by Mc Neill et al, (Mc Neill et al, 1998) was adapted. We looked for the FSB elements and tagged the video with the timestamp and details. The unit of analysis for

function (F) was words/sentences, for structure (S) words and for behaviour (B) it was sentences. The coded video and the artifacts were used to extract the frequency of the FSB elements. The frequency and uniqueness would indicate the breadth and depth of conceptual design. A csv file was created with the FSB codes, details and timestamps. Process Mining Tool (PROM) was used to present in the data in the form of an event stream where the transitions across timelines was generated. The event stream provided the timeline and transitions of FSB elements indicating the cognitive processes

Results

Among the five participants two of them (par2 & par4) completed the task successfully. They both generated artefacts ranging from most basic formal representations like component diagram, use case diagram to the complex formal representations like sequence diagram and process diagram. These participants also had the higher frequency of unique function (F) elements. They also identified appropriate structures and generated behaviours, which lead to the solution design to have sufficient details. The event stream for these two participants indicate a cyclic process of simulating end user interactions (B), abstract them to features of solution (F) and associating it to components (S) in the system.

Par1 and par3 had higher frequency of function (F) elements, however the unique functions were very less. They both simulated behaviours (B) however kept going back to the same behaviour. They were unable to identify structures (S), which indicate the absence of association/abstraction process. Par5 on the other hand kept going back to the same function (F). The unsuccessful candidates were fixated to one of the FSB elements during conceptual design. The FSB framework helped to identify that fixation can happen at absence of certain cognitive processes like mental simulation, abstraction and association. All three unsuccessful participants were unable to utilize formal representation mechanism naturally even though they had the pre-requisite knowledge.

Conclusions and implications

The study task was time bound and we are unsure about the effect of additional time in hand on the participants' conceptual design. We would need to repeat the study for more novices to ascertain if there are any more cognitive processes that we could have missed.

This paper identifies the role of mental simulation, abstraction, association and representation in novices to create conceptual design in the context of software design. The study also identified that fixation can happen across function, structure and behaviour. Fixation could mean that novice couldn't simulate/associate/abstract. We need to explore mechanisms to trigger such processes and also observe experts' process. For teaching learning of conceptual design pedagogical strategies need to be developed which would avoid fixation.

References

Ball, L. J., Onarheim, B., & Christensen, B. T. (2010). Design requirements, epistemic uncertainty and solution development strategies in software design. *Design Studies*, 31(6), 567-589.

Chakrabarti, A., & Bligh, T. P. (2001). A scheme for functional reasoning in conceptual design. *Design Studies*, 22(6), 493-517

Cross, N., Christiaans, H., & Dorst, K. (Eds.). (1996). *Analysing design activity* (Vol. 11, p. 463). Chichester: Wiley.

Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design. *AI magazine*, 11(4), 26.

Goel, A. K., Rugaber, S., & Vattam, S. (2009). Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *Ai Edam*, 23(1), 23-35

Mc Neill, T., Gero, J. S., & Warren, J. (1998). Understanding conceptual electronic design using protocol analysis. *Research in Engineering Design*, 10(3), 129-140.

Pahl, G., & Beitz, W. (2013). *Engineering design: a systematic approach*. Springer Science & Business Media.