Rodrigo, M. M. T. et al. (Eds.) (2021). Proceedings of the 29th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Fostering Conceptual Change in Software Design

Lakshmi T G[*] & Sridhar IYER
IDP in Educational Technology, Indian Institute of Technology Bombay, India
*tglakshmi@iitb.ac.in

Abstract: Novices and experts exhibit differences in understanding and creating software conceptual design. Experts are known to build integrated software solutions that fulfill the requirements of real-world design problems. Novices have specific difficulties such as fixation and lack of integration while creating software conceptual design. Software design teaching-learning approaches have been directed towards software methodologies, processes, and tools. However, teaching-learning interventions for specific novice difficulties and disciplinary practices of software conceptual design are still not available. We created a function-behaviour-structure (FBS) based learning environment, 'think & link' to alleviate the novice difficulties and disciplinary practices of integrated software solution creation. The aim of the study in this paper is to examine the change in novices' outcome and understanding of software conceptual design after having completed all activities in 'think & link'. The study was conducted with final year undergraduate computer and information technology students (n=20) from an engineering college in Mumbai, India. There is no gestalt shift in the pre-post solutions to design problems. However, learners' conceptions of software conceptual design indicate conceptual change. The learners' refined their understanding and developed perspectives about software conceptual design.

Keywords: Conceptual change, software engineering disciplinary practices, function-behaviour-structure design framework, technology enhanced learning environment

1. Introduction

Conceptual design activity is described as a process in which the functional requirements of the design problem are extracted and transformed into descriptions of solution concepts (Chakrabarti & Bligh, 2001). The conceptual phase of design is significant, as designers tend to develop numerous early ideas and solutions in this phase. Conceptual design is inherently hard and needs to be supported (Chakrabarti & Bligh, 2001). The characteristics of software such as intangibility and dynamicity add to the complexity of software conceptual design (Petre et al, 2010).

In the context of software conceptual design (SCD) it is a standard practice to create various representations of unified modeling language (UML) to represent the solution design. However most of the designs created using UML describe system in different notations from different points of view and at different levels of abstraction. A SCD is described by integrating the various UML representations. There is a lack of a single representation to represent all views (Niepostyn & Bluemke, 2012). In the formal curricula of computer engineering and information technology, students learn about syntax, semantics and processes to create the formal (UML) representations. However when students encounter open-ended real world problems they are unable to utilize the formal representations or create meaningful SCD (Eckerdal et al, 2006). Novices are unable to utilize multiple integrated representations in UML for a given design problem (Eckerdal et al., 2006; Lakshmi & Iyer, 2018).

To alleviate these novice difficulties while creating SCD we designed and developed a function-behaviour-structure (FBS) design framework (Gero & Kannengiesser, 2014) based learning environment - 'think & link'. In 'think & link' FBS framework manifests as a manipulable graph. This paper, describes the pedagogical design and features of 'think & link'. The evaluation of the effect of 'think and link' is examined using the conceptual change lens. We examine the learners' pre-port SCD designs and their understanding of SCD after they have completed activities in 'think & link'.

2. Background

Conceptual change research begins with addressing learners' understanding of a given topic and the change in understanding as a result of instruction (von Aufschnaiter & Rogge, 2015). In the context of computer science educators it assists in understanding novices' conception and creating specific instruction to address alternate conceptions (Qian & Lehman, 2017).

Prior studies on novice difficulties in SCD (Eckerdal et al, 2006) (Thomas et al, 2014) (Chren et al, 2019) indicate that novices (i) only rewrite problem statements during design phase, (ii) are unable to utilize formal representations of UML to model SCD and (iii) are unable to utilize multiple UML diagrams for integrated view of solution. Our studies confirmed these findings (Lakshmi & Iyer, 2018). Our studies have further informed us that novices face the difficulties of fixation and lack of integration during the SCD task. For example, given the design problem of mood based music player our study with novices' (Lakshmi & Iyer, 2018) found that they mostly focused on mood detection. The fixation was towards a specific function, structure or behaviour. In contrast expert software designers create comprehensive and cohesive SCD by understanding the problem and creating integrated solution view (Ball et al, 2010). Creating integrated solution designs involves– i) utilization of multiple UML representations that are linked and ii) addressing both problem and solution aspects of the given design problem.

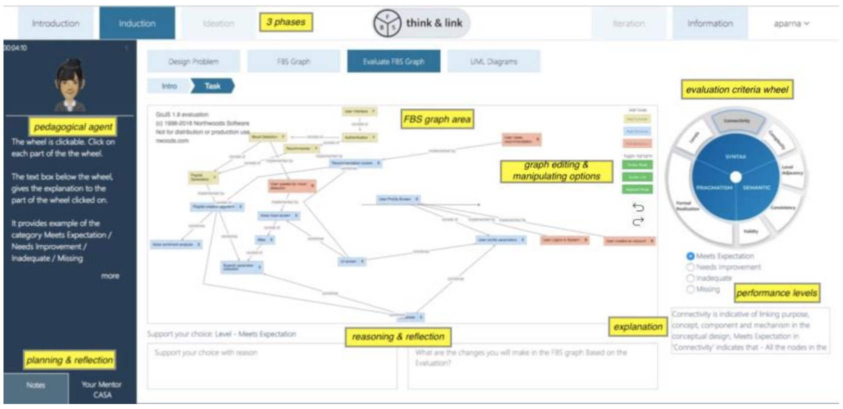
The FBS framework (Gero & Kannengiesser, 2014) models designing in terms of the design elements: function (F), behaviour (B), structure (S), expected behaviour (Be) and behaviour derived from structure (Bs). Functions, describe what the design is for; behaviours, describe what it does; and structures, describe what it is. Along with FBS elements the framework has 2 sets of behaviours, expected behaviour (Be) and behaviour derived from structure (Bs). One may distinguish two main fields of application, '*as a theoretical vehicle for understanding design, and as a conceptual basis for computerized tools intended to support practicing designers* '(Galle, 2009). The function-behaviour-structure (FBS) design framework (Gero & Kannengiesser, 2014) can be considered an appropriate framework to alleviate novices' difficulties of integrating representations in SCD.

3. Design of 'think & link

'think & link' is a web-based, self-paced, FBS framework based learning environment for teaching-learning of SCD. The FBS framework manifests as a FBS graph in the learning environment. The FBS graph is a representation through which learners can symbolize function, structure, behaviour and establish the relationship between them. The FBS graph pedagogy includes creation and manipulation of a representation. 'think &link' consists of scaffolds for learners to create, modify and evaluate a FBS graph for design problems. There are three phases in 'think & link' and the following features:

- FBS graph manipulator and editor – This feature is present throughout the three phases. However in the first phase alone the editor options are not provided to the learners. The graph manipulator displays the FBS graph for the problem with color-coded nodes. The clickable options on the right panel help the learner to display similar nodes, links and adjacent nodes (see figure. 1). In the edit mode the right panel extends clickable options to add function, structure and behaviour nodes. Dragging the cursor from the source node and placing it on destination node creates the link. The link can be annotated with tags - 'implemented by, consists of, and combines' by right clicking on the link. Using the activity, manipulator and clickable options in phase 1 learners build FBS conceptual model. This conceptual model helps the learners link F/B/S together. By editing FBS graphs learners build strategies to create and establish links between F/B/S for a given design problem.
- FBS graph evaluator – This feature is present in the phase 2 and 3. It aids in the self-evaluation of FBS graph based on criterion of syntactic, semantic and pragmatic categories. The criteria of conceptual model were adapted from Lindland et al. (1994). The categories include properties like connectivity, complexity, consistency, validity, consistency, levels and formal realization. All these parameters are adapted and presented in the context of the FBS graph. The evaluation categories are presented as a clickable wheel (see figure. 1). Clicking on an evaluation criteria the performance levels (meets expectation, needs improvement, inadequate, missing) are presented as radio buttons.

The explanation of the criteria and the respective selected performance level is presented to the learner. The learner has to select the performance level after evaluating the FBS graph. The learner needs to support the performance level choice with reason and state the corresponding changes in FBS graph that the learner would make. The learners' self evaluate the categories of SCD in the context of FBS graph. Learners are also required to provide reasoning for the evaluation and reflect on the changes in FBS graph.



The wheel is clickable. Click on each part of the the wheel.

The text box below the wheel, gives the explanation to the part of the wheel clicked on.

It provides example of the category Meets Expectation / Needs Improvement / Inadequate / Missing

[✓ Meets Expectation] Needs improvement Inadequate Missing

Connectivity is indicative of linking purpose, concept, component and mechanism in the conceptual design. Meets Expectation in 'Connectivity' indicates that - All the nodes in the

Support your choice: Level - Meets Expectation

Support your choice with reason

What are the changes you will make in the FBS graph based on the Evaluation?

Figure 1. Features of 'Think & Link'.

- Pedagogical agent – On the left side of 'think & link' (see figure. 1) a vertical column is dedicated to the pedagogical agent CASA (conceptual design assistant). CASA is present all through the phases. CASA provides procedural prompts related to the task that the learner is currently performing. The learner's task progress is monitored and CASA provides appropriate scaffolds to complete the task at the desired level. CASA also provides cognitive prompts, which aid the learner in creating the FBS design elements and linking them.
- Resources for Information - At each and every step of tasks in 'think &link' there are videos, which contain task specific knowledge required to complete the task (see figure. 1). Additionally there is also 'Information' page in the learning environment that includes a collection of videos about the context of the learning environment like SCD, FBS framework etc.
- Planning questions – It is important for the learner to reflect, evaluate and monitor their process during design. By doing this learners will be able to imbibe the process of SCD along with the strategies. As mentioned earlier the learners are taken through the three planes of cognition – doing, evaluation and synthesis. In the planning activity the learners are required to reflect on the task ahead and plan (see figure. 1). Example questions that they encounter are – *What will you do in this phase? How will this task be useful for creating software conceptual designs?*

4. Research Method

The broad research question guiding this study is - What is the nature of conceptual change in software conceptual design after learners' initial interaction with the learning environment 'think & link'?

4.1 Research Design, Participants and Study Procedure

The study was conducted as a hands-on one-day workshop in an urban private engineering institute. The participants for the workshop were selected via purposive sampling. Only participants in their final year of computer engineering (CS) and information technology (IT) were considered for the workshop. 20 students registered for the workshop (CS=15, IT=5: male=16, female=4). The study participants are representative of Indian urban engineering students.

In the workshop registration form along with personal details, participants answered an open-ended questionnaire aimed to capture their prior conception of SCD. First author of the paper along with a colleague were present during the workshop. Participants solved a pre-test at start. They individually created a SCD on pen and paper for the design problem - 'Design a mood based automatic music player'. They were free to use the internet for this task. After completing this task, participants utilized the individual desktop to access ‘think & link’. ‘think & link’ has three phases which the participants completed in ~4.5 hours. After completing all activities in ‘think &link’, participants for an hour solved another SCD for – ‘Design a finger print ATM system’. The pre and post SCD problems can be considered equivalent, as they are similar in terms of complexity and time taken to solve. After completing the post-test, participants were asked to respond to questions about understanding of term software conceptual design and a focus group semi structured interview was conducted. Participants spent 5 hours in the workshop.

4.2 Data Sources and Analysis

The online registration form for the workshop contained prior conception open-ended question to capture participants’ understanding of the term software conceptual design. The question that participants were asked was, “What is your understanding of 'software conceptual design'?” The sketches and notes for the pre and post paper based activities were collated as artifacts of the design activity. At the end participants’ conceptions about software conceptual design, usability and usefulness of ‘think & link’ was also captured as responses to online questionnaires. Table 1 below maps the measure, data source and the analysis technique.

Table 1. *Mapping of Measure, Data Source and Analysis Technique*

Measure	Data Source	Analysis
Pre-post change in SCD	Artifacts	Category evaluation [(Thomas et al., 2014)
Pre and post change in understanding of SCD	Pre-post open ended responses	Inductive thematic analysis (Clarke & Braun, 2016)

The categories of software conceptual design by Thomas et al. (2014) were utilized to analyze the pre-post design artifacts. These categories were utilized to classify the pre and post SCD of participants. Once the participants’ artifacts were classified, there emerged a need to explicate the participants’ transition across the categories in pre and post-test. The tool interactive stratified attribute tracking, iSAT (Majumdar & Iyer, 2016) was used to analyze the pre-post categories transition.The visualization from the tool is shown in figure. 5 and elaborated in the next section.

To analyze the open-ended responses to the questionnaire the guidelines provided by Clarke and Braun (2016) were followed. The unit of analysis was a sentence. The codes represented the relevant qualities, activities and outcome of SCD. The comparison between the pre-post responses is captured in sub-section 5.1 and 5.2 in results section.

5. Results

The goal of this paper is to explore the nature of conceptual change in SCD after learners completed activities in ‘think &link’. To analyze the transitions across the pre-post categories, we used the tool iSAT. The observations from analyzing the pre –post transitions (figure 2) are:

- none of the participant slid down to lower category in post test
- majority of the participants in both pre and post fell in the category 3 as they utilized the representation of flowchart
- 3 participants’ in the post intervention moved to the category 3 from category 0 (figure 4). The majority of the dynamic representations utilized by participants were flow chart, so that could explain the shift. A participant moved to category 3 from category 1. This movement could indicate that participants are able to create dynamic representations than the static formal representations.
- 2 participants’ in the post intervention moved to the category 4 from category 3 (figure 4). During the intervention the understanding that software conceptual design comprises of multiple artifacts depicting the functional, behavioral and structural view could be the reason for the shift.

There was no ‘gestalt shift’ in the participants’ artifacts, i.e. all the participants in post intervention did not move to category 4. However the slight shift as evident from drop in category 0, rise in category 3 and 4. Additionally to investigate the participant understanding of SCD the open-ended response thematic analysis was done. Two major themes of refinement of understand and perspective shift emerged. Under each subsection, the shift in understanding is discussed with example responses, their corresponding code and theme.

5.1 Refinement of Understanding about SCD

The participants' developed the understanding about SCD being a *'combination of all UML diagrams'* (code: drawings, theme: outcome) rather than just thinking about *'conceptual & schematic drawings'* (code: drawings, theme: outcome). There is a refinement in the outcome of SCD as an integrated solution from participants' responses such as- (i) *doing a conceptual design going deep into what actually the problem is and form connections to various solution parts actually*, (ii) *what will be the back end, what will be the front end, how will front end access back end*. The other observed refinement in participants' understanding about SCD is that *'.... need to understand intricacies for implementing minor details'* (code: details of solutions, theme: activities) whereas in pre response we see the response as *'creating design modules'* (code: module creation, theme: activities). Earlier participants' understanding about activities in SCD was 'documentation' whereas now they refine their understanding as SCD involving design and creation ('*in software engineering we didn't design anything, it was just documentation'*).

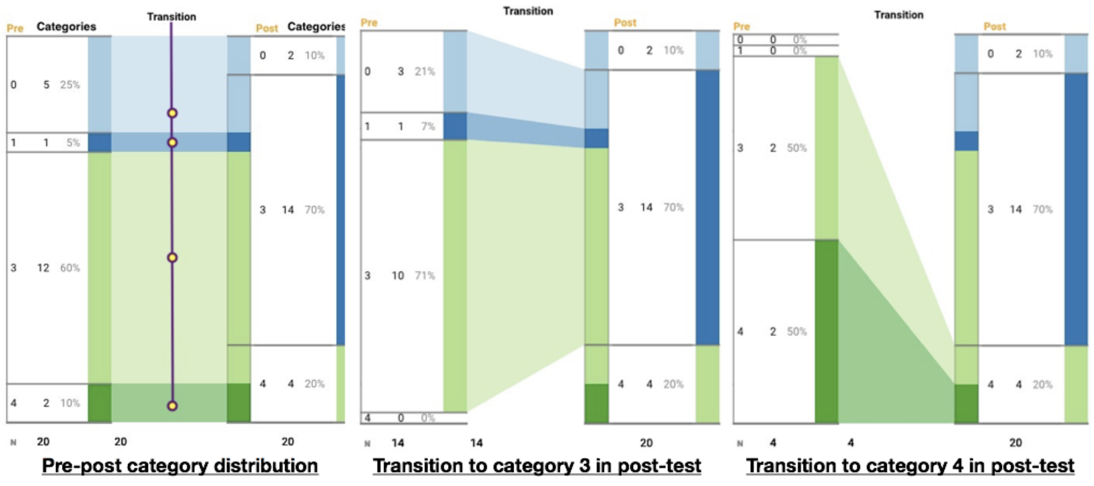


Figure 2. Transition of Pre-Post Artifact Category Classifications.

5.2 Perspective Shift

The participants' developed alternative views about SCD, which will be useful during solving design problems. The first shift in perspective is about SCD being a systematic approach – '*it is a systematic way instead of just throwing things on paper'* (systematic approach- activities). Pre-intervention participants' design for 'customer requirements in modules'. After intervention participants' acquire the perspective of designing for understanding of other '*designers as well as programmer or developer'*. The participants' also develop the perspective about the cognitive process involved in the activity – *'we need to first imagine how will end user use it, then create use cases, identify components'*. The participants' view SCD as a stage '*before coding'* where they are required to *'mention all steps so that it is as close to the real software',* whereas, in pre intervention response they view it as a '*phase extracting problem chara*cteristics'.

6. Discussion and Conclusion

The driving research question in this paper is to examine the nature of conceptual change in learners understanding of SCD. The pre-post open-ended responses indicate that learners' have undergone a shift in their understanding of SCD. The open-ended responses collated before the commencement of the workshop capture the conception participants had before the exposure to 'think & link'. Although the coding themes in pre-post almost remain the same, the responses themselves were contrasting in nature. This indicates a conceptual integration (Vosniadu, 2013), in which the practices and understanding from the activities in ‘think & link’ are combined with participants’ earlier conceptions. Although the pre-post change in categories of design solution observed at the end of the workshop was not significant (figure 2). The gradual change in the participants’ understanding about the outcomes and activities in SCD is indicative of conceptual change (Nussbaum, 1989). The design features of ‘think &link’ could be utilized by teachers/researchers who want to develop conceptual change in novices’ understanding of SCD. The results from this study have implication for the teaching learning of software conceptual design. The design features of ‘think &link’ could be utilized by teachers/researchers who want to develop conceptual change in novices’ understanding of scd. Some of the results that can be utilized for teaching learning of SCD are: (i) explicit linking of different aspects of scd, (ii) recognizing importance and deliberate practice of SCD, (iii) scaffolds for cognitive processes. We have identified two major limitations of the study. First, the sample size of the study is small (n=20) for generalizability. However the goal of this study is to explore the nature of conceptual change in the context of software conceptual design after learner interactions with the mediating tool ‘think and link’. Second, reconfirmation of conceptual change by participants after thematic analysis was not done.

Future studies have been planned to closely examine the changes. Additionally, longitudinal studies have been planned by providing access to the learners to interact with ‘think & link’ for longer periods of time and collect data pertaining to processes, practices and discourse.

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.

Chren, S., Buhnova, B., Macak, M., Daubner, L., & Rossi, B. (2019, May). Mistakes in UML diagrams: analysis of student projects in a software engineering course. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering Education and Training* (pp. 100-109). IEEE Press

Clarke, V. & Braun, V., 2016. Thematic analysis. *The Journal of Positive Psychology*, 12(3), pp.297-298

Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., & Zander, C. (2006). Can graduating students design software systems?. In *SIGCSE’06* (pp. 403-407). ACM.

Galle, P. (2009). The ontology of Gero's FBS model of designing. *Design Studies*, *30*(4), 321-339.

Gero, J. S., & Kannengiesser, U. (2014). The function-behaviour-structure ontology of design. In *An anthology of theories and models of design* (pp. 263-283). Springer, London.

Lakshmi, T. & Iyer, S. (2018). Exploring Novice Approach to Conceptual Design of Software. *In Kay, J. and Luckin, R. (Eds.) Rethinking Learning in the Digital Age: Making the Learning Sciences Count, 13th International Conference of the Learning Sciences (ICLS) 2018*, Volume 3. London, UK: International Society of the Learning Sciences

Lindland, O. I., Sindre, G., & Solvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE software*, *11*(2), 42-49.

Majumdar, R., & Iyer, S. (2016). iSAT: a visual learning analytics tool for instructors. *Research and practice in technology enhanced learning*, *11*(1), 16.

Niepostyn, S. J., & Bluemke, I. (2012, June). The Function-Behaviour-Structure Diagram for Modelling Workflow of Information Systems. In *International Conference on Advanced Information Systems Engineering* (pp. 425-439). Springer, Berlin, Heidelberg.

Nussbaum, J. (1989). Classroom conceptual change: philosophical perspectives. *International Journal of Science Education*, 11(5), 530-540.

Petre, M., van der Hoek, A. and Baker, A. (2010). Editorial. *Design Studies*, 31(6), pp.533-544.

Qian, Y., & Lehman, J. (2017). Students’ misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1.

Thomas, L., Eckerdal, A., McCartney, R., Moström, J. E., Sanders, K., & Zander, C. (2014, July). Graduating students' designs: through a phenomenographic lens. In *Proceedings of the tenth annual conference on International computing education research* (pp. 91-98). ACM.

Von Aufschnaiter, C., & Rogge, C. (2015). Conceptual change in learning. *Encyclopedia of science education*, 209-218.

Vosniadou S (ed) (2013) International handbook of research on conceptual change, 2nd edn. Routledge, New York/London