Ph.D. Defence Presentation

Fostering software conceptual design via the Function-Behaviour-Structure design framework

T.G.Lakshmi
154380002

under the guidance of Prof. Sridhar Iyer

05 July 2021, Ph.D. defence
Motivation
An example of software conceptual design (SCD) problem

Create a software conceptual design for a mood based music player system.

The system has following requirements:
- System needs to detect mood
- Play music automatically according to mood
- Provide secure authentication
- Remember user's choice of music
- Recommend music based on the history of user's choice

• Given this problem to undergraduate computer engineers:
  • Ideas
  • Software Engineering Course : Unified Modeling Language (UML) diagram
Software Conceptual Design is a critical design practice

From industry & academia

- ~60%* of the total product cost is fixed at the conceptual design phase
- Critical and important phase in design (Dym et al, 2005; Chakrabarti & Bligh, 2001; Pahl & Beitz, 2013)

Learner Difficulties

- Graduating students cannot design software (Thomas et al., 2017)
- Difficulties such as fixation, strategies, generating ideas/solution concepts (Stempfle, 2011; Gero, 2018; Tang et al., 2010)

Software Engineering (SE) Teaching- Learning approaches

- Directed towards SE methodologies and processes, tools for requirement analysis/project management (Naveda et al., 2008; Teel et al., 2012; Fonseca et al., 2017)
Research Gap
What is software conceptual design (SCD)?

Definitions

- Definition of conceptual design - design literature

- The functional requirements are elicited and schematic descriptions of solution are generated (Chakrabarti & Bligh, 2001)

- Software Conceptual Design (Jackson, 2013)
  - description which is implementation independent
  - support analysis
  - support exploration of design spaces
What is software conceptual design (SCD)?
Outcome quality (Lindland et al., 1994)

<table>
<thead>
<tr>
<th>Quality parameters</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic</td>
<td>• Syntactic correctness</td>
</tr>
<tr>
<td>Semantic</td>
<td>• Feasible validity</td>
</tr>
<tr>
<td></td>
<td>• Feasible completeness</td>
</tr>
<tr>
<td>Pragmatic</td>
<td>• Feasible comprehension</td>
</tr>
</tbody>
</table>

- This framework is operationalised for evaluation in our pedagogy
- It incorporates all the three perspectives of - Syntax, Semantic, Pragmatic
What is the expected output of SCD? (Eckerdal et al., 2006)

<table>
<thead>
<tr>
<th>Category</th>
<th>Content (Indicators)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restatement</td>
<td>No design content other than stated in the description</td>
</tr>
<tr>
<td>Skumtomte</td>
<td>Unimportant implementation details</td>
</tr>
<tr>
<td>First step</td>
<td>Some significant work beyond restatement</td>
</tr>
<tr>
<td>Partial design</td>
<td>Understandable description of parts and overview</td>
</tr>
<tr>
<td></td>
<td>Description of parts maybe incomplete or superficial</td>
</tr>
<tr>
<td></td>
<td>Communication between parts may not be completely described</td>
</tr>
<tr>
<td>Complete Design</td>
<td>*Well developed solution</td>
</tr>
<tr>
<td></td>
<td>*Understandable overview</td>
</tr>
<tr>
<td></td>
<td>*Solution parts description includes explicit communication between them</td>
</tr>
<tr>
<td></td>
<td>*Formal representations as well as text</td>
</tr>
</tbody>
</table>

Undesirable categories

Desirable categories
Example of category 5 in Software conceptual design (Thomas et al., 2017)

The task: The Parking Garage

You are asked to produce a design for the software system that runs a parking garage. Drivers will have a mobile app that allows them to register the license, and whether the car is compatible with the garage system. The system should therefore include the user whether there is a space and where to find it. When the driver returns to the garage they will be told how much to pay and be reminded where their car is.

Alternatively, drivers can drop off their car at the front door and give their car to a garage employee. In that case, the employees use the same mobile app to enter the relevant information. In addition to finding a space as outlined above, the garage employees are allowed to check the garage and find an empty parking space. They enter the space ID into the system, check its availability, and if it is free register that they have occupied the space. They are also able to indicate that the customer has arrived back and requested their car and that they have gone to get it; and then that it has been delivered to the customer and the car is paid for.

You only have 36 minutes, so produce the best design that you can in this amount of time: you are producing an initial solution that someone (not necessarily you) could work from, include as many artifacts (for example, use cases, context diagrams, sequence diagrams, use case diagrams, flowcharts, user interface design, pseudocode etc.) as needed and at time permits, so someone could fill in the details for your design and implement it.

Requirements:
Mobile App
- Register car and parking license plate
- Complete/handicapped
- Tie the fact to pay
- Google Earth app
- Use ID
- Customer and bank
- Customer can call on phone

Use case

Example Use:
User enters at garage, logs in the app, registers car using license plate, app opens screen to find a space, shows distances on a color-coded map, clicks on space, enters parking lot, uses phone to park, enters app for payment, parking lot is confirmed and paid. Again visit to car and moves for payment.

Class diagram

Sequence diagram
What are the expert practices in creation of SCD?

• Experts make
  
  • implicit connections between the various representations (Hungerford et al., 2004)
  
  • build an integrated model of the design (Petre, 2009)
  
• Professional software design experts during creation of SCD utilise
  
  • design strategies - mixed breadth strategies (Ball et al, 2010), co-evolve problem & solution (Tang et al, 2010)
  
  • cognitive processes - mental simulation, abstraction, association (Ball et al, 2010)
  
  • formal representations - integrated UML modeling (Chren et al, 2019)
How do novices create designs?

• From engineering design literature novices utilise
  • depth-first strategy (Ahmed et al., 2003; Hokanson, 2001)
  • random search strategy (Chrysikou & Weisberg, 2005)
  • design fixation (Vishwanathan & Linsey, 2013)
  • generating ideas/solution concepts (Pan et al., 2010)
• Software Conceptual Design
  • novices unable to create design that had overview of parts and relationship between parts (Eckerdal et al., 2006)
• Characterisation of novice difficulties is missing
  • processes & strategies in SCD are not unpacked
  • difficulties mapping to processes & strategies
Research goals of this thesis

1. Developing an understanding of *novice processes* in *software conceptual design (SCD)*

2. Designing and evaluating a *technology enhanced learning environment* to support creation of *software conceptual design (SCD)*
Scope and Context

• Domain: Software Engineering

• Specific Topic: Software Conceptual Design (SCD)

• Problem: Teaching-Learning of SCD

• Learner Characteristic: UG second - final year computer engineering/information technology

• Learning Context: Software Engineering design lab/ Final year project lab
Approach to Solution

- Theoretical Framework
- Methodological Framework
Function-Behaviour-Structure (FBS) Design Framework

**F- Function** captures capabilities of solution  
e.g. Voice based mood detection

**Be - Expected behaviour of system**  
e.x. System needs to capture voice

**S-Structure** indicates solution concepts & components  
e.g. Emospark camera, facial recognition algorithm

**Bs - Behaviour of structure** extracted from structure  
e.g. Facial features/points extracted

Design processes in FBS design ontology (Gero & Kannengiesser, 2014)
Why is FBS an appropriate framework for SCD?

• Universal Design Framework

• Applicable to any engineering discipline (Krutchen, 2005)

• Engineering design (Umeda, 1996), Mechanical design (Shimomura et al., 1998), Product development (Christophe et al., 2010), Theoretically proposed for programming (Guzdial, 2012)

• Supports Integrated View

• Unified Modeling Language (UML) most commonly used, however has notations from different points of view

• Need for unified and integrated view to support the consistency and completeness of the design was identified (Niepostyn & Bluemke, 2012)

• Supports Abstraction

• Software engineers grapple with abstraction at conceptual design phase (Pressman, 2005)

• FBS design framework is categorised as a abstract micro model that can be represent design as elementary abstract processes (Wynn & Clarkson, 2018)

“as a theoretical vehicle for understanding design, and as a conceptual basis for computerized tools intended to support practicing designers” (Galle, 2009)
Approach to Solution

Theoretical Framework

Methodological Framework
I. Unpacking novices’ design strategies & cognitive processes

II. Unpacking novices’ difficulties while learning using FBS based interventions

III. Identifying changes in novices’ SCD understanding & process

• Iterative
• Integrates variety of research methods
• Pragmatic

DBR Cycle 1
Unpacking Novice SCD processes and FBS design framework based interventions

05 July 2021, Ph.D. defence
Study 1

Unpacking novices’ design strategies & cognitive processes

RQ 1.a What are novices’ design strategies while creating scd?

RQ 1.b What are novices’ cognitive processes while creating scd?
## Study 1 - Method

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methodology</th>
<th>Data Collection</th>
<th>Analysis</th>
</tr>
</thead>
</table>
| RQ 1.a - What are the design strategies that novices’ follow while creating a SCD? | Exploratory Qualitative Study (Mack, 2005) n=5 | • Video recording  
• Screen capture  
• Participant generated artifact (notes, drawings, electronic documents generated) | • Categories of SCD (Eckerdal et al., 2006)  
• FBS based Linkograph analysis (Kan & Gero, 2009) |
| RQ 1.b - What are the cognitive processes that novices’ use while creating a SCD? |                                                                                       | • Participant generated artifact (notes, drawings, electronic documents generated)  
• Video recording  
• Interview transcripts | • Categories of SCD (Eckerdal et al., 2006)  
• Deductive thematic analysis (Aronson, 1994) based on Conceptual design cognition (Hay et al, 2017) |
# Summary of Results of RQ1

## Novice group

<table>
<thead>
<tr>
<th>Novice group</th>
<th>Category</th>
<th>Content (Indicators)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsuccessful novices (category 1-3)</td>
<td>Restatement</td>
<td>No design content other than stated in the description</td>
</tr>
<tr>
<td></td>
<td>Skumtomte</td>
<td>Unimportant implementation details</td>
</tr>
<tr>
<td></td>
<td>First step</td>
<td>Some significant work beyond restatement</td>
</tr>
<tr>
<td>Successful novices (category 4 &amp; 5 )</td>
<td>Partial design</td>
<td>Understandable description of parts and overview&lt;br&gt;Description of parts maybe incomplete or superficial&lt;br&gt;Communication between parts may not be completely described</td>
</tr>
<tr>
<td></td>
<td>Complete Design</td>
<td>* Well developed solution&lt;br&gt;* Understandable overview&lt;br&gt;* Solution parts description includes explicit communication between them&lt;br&gt;* Formal representations as well as text</td>
</tr>
</tbody>
</table>

## Results RQ 1.b. What are novices’ cognitive processes while creating SCD?

- Information seeking
- Mental simulation, association, analogical reasoning and synthesis
II. Unpacking novices’ difficulties while learning using FBS based interventions
Sample FBS graph for the mood based music player design problem

Mood detection (F) is implemented by a Voice Input device (S) which requires User to speak (B)
Learning Objectives for FBS graph based pedagogy

Learners need to

• build a syntactic & semantic conceptual model of FBS

• link the FBS design elements and to create FBS graphs

• apply the FBS conceptual model and strategies to create a FBS graph in a new problem context
FBS graph based intervention- I & II

- Two phases in both intervention
- Phase I - worked example, FBS graph for a finger print ATM system was provided
- Phase II - different problem, learners need to connect the appropriate F/B/S nodes
- Post-test - Set their own problem and create FBS graph for the same

combination of a webpage and IHMC CMAP tool
Unpacking novices’ difficulties while learning using FBS based interventions

RQ 2.a After interacting with the FBS based interventions what are categories of SCD that learners’ create?

RQ 2.b What difficulties do learners’ experience while using FBS based learning designs?
## Study 2 & 3 - Method

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methodology</th>
<th>Prior Knowledge</th>
<th>Data Collection</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 2.a - After interacting with the FBS based interventions what are categories of SCD that learners’ create?</td>
<td>Study 2 - Laboratory study (n=2)</td>
<td>Sem V: Structured and Object Oriented Analysis and Design</td>
<td>• Participant generated artifact (notes, drawings, electronic documents generated)</td>
<td>• Rubric for integrated SCD adapted for FBS graph (Lindland et al., 1994)</td>
</tr>
<tr>
<td>RQ 2.b - What difficulties do learners’ experience while using FBS based learning designs?</td>
<td>Study 3 - Laboratory study (n=3)</td>
<td>Sem VI: Software Engineering And design of software</td>
<td>• Researcher observations • Interview transcripts</td>
<td>• Thematic analysis (Clarke &amp; Braun, 2014)</td>
</tr>
</tbody>
</table>
Findings from Study 2 & 3 leading to features required in TELE

<table>
<thead>
<tr>
<th>Difficulty from Study 2</th>
<th>Difficulty from Study 3</th>
<th>Features required in TELE</th>
</tr>
</thead>
</table>
| lack of scaffolds to understand FBS conceptual model | using the worksheet the participants built the conceptual model of FBS | * Scaffolds and prompts for task completion  
* Worksheet containing guided questions for building syntactic and semantic model of FBS |
| lack of scaffolds for strategy to connect FBS | rubric to self-evaluate FBS graph unutilized | * Interactive and improvable FBS graph models as scaffolds  
* Cognitive process triggers as adaptive scaffolds to create and connect FBS design elements  
* Self-evaluation activity to evaluate FBS graph |
DBR Cycle 2 - Design & Development of Technology Enhanced Learning Environment (TELE)
Our Solution - ‘think & link’

Function-Behaviour-Structure (FBS) design framework based learning environment
http://thinknlink.tech

Sample learner login
Username: Prathiksha
Passwd: seokjin

Teacher/instructor login
Username: etiitb
Passwd: thinknlink2019

05 July 2021, Ph.D. defence
'think & link' - Demo

Educational Technology, IIT Bombay

Create Account

Username
Email
Password

Create Account
FBS graph based pedagogy in ‘think & link’

- Learners should be taken through progressive planes of cognition **doing, evaluation, synthesis**
- Learner not only needs to complete the tasks but also **needs to abstract the process of learning** (Litzinger et al 2011; White & Frederiksen, 2005)
- **Reflection tasks interleaved** to evaluate the artefact and adjust the process
- **Planning questions for the learner to set goals before proceeding with tasks**
III. Identifying changes in novices’ SCD understanding & process

RQ 3.a What are the categories of SCD that learners' create?

RQ 3.b What are the changes in learners’ understanding of SCD?

RQ 3.c What changes in process of creating SCD do the learners’ perceive?

RQ 3.d How do the learners’ use the features in TELE?
Study Design 4 & 5

**Software Conceptual Design**
- Build FBS Conceptual Model in Problem 1
- Apply FBS Conceptual Model to Connect Design Elements in Problem 1
- Create and Connect FBS Design Elements in Problem 2

**Mood-based music player**
- Pre-test (~ 1 hr)
- 1.5 hr
- Retrospective Interview (study 5)

**Self posed problem**
- 1.5 hr
- Retrospective Interview (study 5)

**Finger print based ATM system**
- 1.5 hr
- Retrospective Interview (study 5)
- Post-test (~ 1 hr)

**Questionnaire**
- Focus Group Interviews
## RQ 3.a - Method

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methodology</th>
<th>Category</th>
<th>Content (Indicators)</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 3.a - What are the categories of SCD that learners’ create?</td>
<td>Workshop study</td>
<td>Restatement</td>
<td>No design content other than stated in the description</td>
<td>Categories of SCD (Eckerdal et al., 2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skumtome</td>
<td>Unimportant implementation details</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>First step</td>
<td>Some significant work beyond restatement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Partial design</td>
<td>Understandable description of parts and overview Description of parts maybe incomplete or superficial Communication between parts may not be completely described</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Complete Design</td>
<td>&quot;Well developed solution&quot; &quot;Understandable overview&quot; &quot;Solution parts description includes explicit communication between them&quot; &quot;Formal representations as well as text&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Results - RQ 3.a

RQ 3.a What are the categories of SCD that learners’ create?

In post-test
- slight increase in artifacts categorised in partial design & complete design
- slight decrease in artifacts categorised in restatement, skumtomte, first step
Results - RQ 3.a

RQ 3.a What are the categories of SCD that learners’ create?

In post-test
- Increase in participants creating SCD depicting only behaviour & dynamic aspects
- Decrease in participants creating SCD only depicting static aspects
**RQ 3.b - Method**

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methodology</th>
<th>Participants</th>
<th>Data Collection</th>
<th>Analysis</th>
</tr>
</thead>
</table>
| **RQ 3.b - What is the difference in learners' understanding of SCD?** | Workshop study | Study 4 - Final year computer engineering students N=20 (study 4)  
Study 5 - Second year computer and information technology students N=18 (study 5) | Pre-post responses to open-ended questions in survey | Thematic analysis (Clarke and Braun, 2014) |
Results - RQ 3.b

RQ 3.b What is the difference in learners’ understanding of SCD?
### RQ 3.c - Method

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methodology</th>
<th>Participants</th>
<th>Data Collection</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 3.c - What changes in process of creating SCD do the learners’ perceive?</td>
<td>Workshop study • N=20 (study 4) • N=18 (study 5)</td>
<td>Study 4 - Final year computer engineering students Study 5 - Second year computer and information technology students</td>
<td>• Post focus group interviews • Randomly selected participant reflections during TELE usage</td>
<td>Thematic analysis (Clarke and Braun, 2014)</td>
</tr>
</tbody>
</table>
Results - RQ 3.c

RQ 3.c What changes in process of creating SCD do the learners’ perceive?

Legend
- Pre-Intervention responses
- Post-Intervention responses
Results - RQ 3.c

RQ 3.c What changes in process of creating SCD do the learners’ perceive?

- Designing for implementation
  - "only drawing the use case diagram and like, I didn’t know much about how the programmer will be implementing"
  - "how the functions are related to each other and how each and every structure is present inside the function and how the user will be able to use it. how capabilities of the software are implemented."

- Designing for requirements explanation
  - "I used design diagrams, like I have made use-case diagrams, to satisfy requirements."
  - "now creating sketches to make it more clear for the programmer to understand what the requirements are expected"

- Balancing between capabilities and complexity
  - "I used to design all the requirements into features"
  - "like increase features beyond a point and it could clutter up the graph and make the software complex so you like that balancing between that functions"

- Utilizing FBS strategies
  - "I start with a use case diagram, but then I don’t know how my software will work. I used to get stuck."
  - "I started with the structure, what all components will I need; as in physical components and then I went to functions, what all functions they would serve and then serve that purpose, what are the logical components will I needs."
**RQ 3.d - Method**

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methodology</th>
<th>Participants</th>
<th>Data Collection</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 3.d - How do the learners’ use the features in TELE?</td>
<td>Workshop study</td>
<td>Study 4 - Final year computer engineering students</td>
<td>• Participant actions and events recorded in the system</td>
<td>• Event sequence mining in R (Ritschard et al, 2013)</td>
</tr>
<tr>
<td></td>
<td>• N=20 (study 4)</td>
<td>Study 5 - Second year computer and information technology students</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• N=18 (study 5)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Inferences from participant event sequences

- Phase 1 - utilised the FBS graph and completed the worksheet
- Phase II - did not edit the graph and attempted the evaluation task and completed the phase
- Phase III - linear completion of tasks
### Event sequences ↔ Post test

<table>
<thead>
<tr>
<th>Phases in ‘think &amp; link’</th>
<th>Informal design category (1, 2 &amp; 3)</th>
<th>Partial design category (4)</th>
<th>Complete design category (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>do not edit the graph in this phase</td>
<td>edit graph and then evaluate, however while examining their edits reveals only addition of either a function or behaviour</td>
<td>move back &amp; forth between evaluation &amp; graph edit tasks. They also move across the phases I &amp; II</td>
</tr>
<tr>
<td>III</td>
<td>follow linear progression of tasks</td>
<td>refer to evaluation done in previous phase to complete evaluation in this phase</td>
<td>back &amp; forth between problem setting, graph edit &amp; evaluation tasks. They also move across the phases II &amp; III</td>
</tr>
</tbody>
</table>
Discussion
## Claims of this thesis

<table>
<thead>
<tr>
<th>Claims</th>
<th>Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novices fixate when they utilize only F/B/S based design strategies</td>
<td>Study 1</td>
</tr>
<tr>
<td>Following features and scaffolds are required in learning environment that supports the process of creation of SCD</td>
<td>Study 1, 2 &amp; 3</td>
</tr>
<tr>
<td>• sketching feature to create &amp; connect FBS design elements</td>
<td></td>
</tr>
<tr>
<td>• evaluation feature to evaluate connected FBS elements</td>
<td></td>
</tr>
<tr>
<td>• planning &amp; reflection opportunities to abstract SCD process</td>
<td></td>
</tr>
<tr>
<td>• adaptive prompts for integrated design strategies and trigger cognitive processes of mental simulation, abstraction, association</td>
<td></td>
</tr>
<tr>
<td>Novices assimilate SCD disciplinary practices in understanding as well as processes after explicit training in FBS based intervention</td>
<td>Study 4 &amp; 5</td>
</tr>
</tbody>
</table>
Implications

• Guidelines for instructors

  • *Explicitly create and establish relationship* between design elements

  • *Deliberate practice* of SCD

• Scaffolds for *cognitive processes*

• Computing Education researchers

  • *Characterisation* of novice design strategies and difficulties

  • *Function-behaviour-structure design framework* in software engineering
<table>
<thead>
<tr>
<th>Contributions</th>
<th>Implications for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characterisation of novices’ design strategies and cognitive processes while creating software conceptual design</td>
<td>Researchers in computing education research, learning science and design education</td>
</tr>
<tr>
<td>Identified a set of features and scaffolds for novices teaching-learning of FBS based software conceptual design</td>
<td>Instructional designers and software engineering educators</td>
</tr>
<tr>
<td>Pedagogical design of a FBS based learning environment for teaching-learning of software conceptual design</td>
<td>Instructional designers and software engineering educators</td>
</tr>
<tr>
<td>Identified the usage of features in the learning environment by engineering undergraduates</td>
<td>Instructional designers, Researchers in building TELE</td>
</tr>
<tr>
<td>think &amp; link is an instantiation of the FBS based pedagogy. A teacher authoring tool for different FBS graph contexts.</td>
<td>Software engineering students and software engineering educators</td>
</tr>
</tbody>
</table>
Generalizability

• Extension to other design problems

• Instructor authoring tool has been provided

• Similar design problems can be utilised for teaching-learning of SCD

• Extension to other design tasks in CS apart from SCD

• Programming is also a design task. Theoretically programming also has been situated in the FBS design framework space (Guzdial, 2018)

• Application of the FBS graph pedagogy to the comprehension and creation of code
Limitations

- **Learner characteristics were kept constant** - differences in motivation, self efficacy, language were not considered.

- **Software Conceptual Design problem characteristics**
  - Problem characteristics - usage familiarity
  - Scaffolds & prompts may vary for different kinds of problems - creative problems

- **Singular perspective - cognitive**
  - Only considered interactions with self as well as the environment
  - Any other theoretical lens would lead to other results
Future Work

• Role of perspective switching in SCD

• ‘think & link’
  • Large scale research studies
  • Adaptive visual dialogue agent
  • Mining for learner actions and FBS graph
  • Instructor and learner dashboard as meta-cognitive scaffolds

• Role of affect in SCD - motivation, interest, self-efficacy

• Role of collaboration in SCD
Publications

**Thesis Publications**

**Publications in pipeline**
- **Lakshmi, T. G. & Iyer, S.** (2021). Fostering conceptual change in software design in IEEE Transactions on Education. Under review
Thank You

Questions please
Detail information
# Design Problem Characteristics (Brown & Chandrasekaran, 1989)

<table>
<thead>
<tr>
<th>S.no</th>
<th>Class of Design Problem</th>
<th>Problem Decomposition</th>
<th>Design Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Class I (Creative)</td>
<td>Not Known</td>
<td>Not Known</td>
</tr>
<tr>
<td>2</td>
<td>Class II (Innovative)</td>
<td>Known</td>
<td>Not Known</td>
</tr>
<tr>
<td>3</td>
<td>Class III (Routine)</td>
<td>Known</td>
<td>Known</td>
</tr>
</tbody>
</table>
Study 1 - Details
Study 1 - Procedure

- N=5
- Conceptual Design problems - (i) Design a fingerprint ATM system (ii) Design a mood-based automatic player (iii) Design a fingerprint-based payment system (iv) Design a cooking recipe recommender system
RQ 1.a What are novices’ design strategies while creating SCD?

- Identified two groups of participants based on artifact evaluation – Successful & Unsuccessful

- Artifact evaluation using scd categories (Eckerdal et al, 2006)

- Merged timeline, segmentation & generation of linkograph
  - Created merged timeline
  - Segmented based on FBS codes
  - Relationship between segments- linkograph

- Link index, Critical moves
  - Analysis of chunks

- Linkograph analysis

RQ 1.b What are novices’ cognitive processes while creating SCD?

- Identified two groups of participants based on artifact evaluation – Successful & Unsuccessful

- Artifact evaluation using scd categories (Eckerdal et al, 2006)

- Merged timeline & interview transcripts
  - Code for conceptual design cognition based (Hay et al, 2017) cognitive processes

- Study 1 - Analysis

- Link index, Critical moves
  - Analysis of chunks

- Linkograph analysis

- Merged timeline & interview transcripts
  - Code for conceptual design cognition based (Hay et al, 2017) cognitive processes
<table>
<thead>
<tr>
<th>Category</th>
<th>Content (Indicators)</th>
<th>Representation (indicators)</th>
<th>Group</th>
</tr>
</thead>
</table>
| Nothing        | Little or unintelligible content                                                                                                                                                                                                                                                                                                                   | Single labelled diagram  
Informal design                                                                                   | Unsuccessful       |
| Restatement    | * Restate requirements from task description  
* No design content other than stated in the description                                                                                                                                                                                                                                  | List or Bulleted items  
Informal design                                                                                   |                    |
| Skumtomte      | * Add a small amount to restating task  
* Unimportant implementation details  
* No overall system view and any work on modules                                                                                                                                                                                                                                | Simple GUI  
Notations such as flow chart                                                                 | Unsuccessful       |
| First step     | * Some significant work beyond restatement                                                                                                                                                                                                                                                                                                          | Formal notation representing structure  
Design of one of the system’s components like GUI or Database                                  |                    |
| Partial design | * Understandable description of parts and overview  
* Description of parts maybe incomplete or superficial  
* Communication between parts may not be completely described                                                                                                                                | Formal notation representing behaviour  
Illustration of relationship between the parts                                                                                     | Successful         |
| Complete Design | * Well developed solution  
* Understandable overview  
* Solution parts description includes explicit communication between them  
* Formal representations as well as text                                                                                                                                                                                                   | Multiple formal notations such as Use case, Class diagram, component diagram                           |                    |

Artefact Evaluation (Eckerdal et al., 2006)
FBS codes for merged timeline

<table>
<thead>
<tr>
<th>Design Element</th>
<th>Code</th>
<th>Classification Indicator</th>
<th>Example Design Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>F</td>
<td>activity performed by the software system</td>
<td>Mood detection</td>
</tr>
<tr>
<td>Expected Behaviour</td>
<td>Be</td>
<td>expected behaviour of the system extracted from the functions</td>
<td>Voice Based Mood Detection (F) - System needs to capture the voice</td>
</tr>
<tr>
<td>Structure</td>
<td>S</td>
<td>the solution concepts and components (hardware and software) required to achieve the function</td>
<td>Camera, software to detect mood</td>
</tr>
<tr>
<td>Structural Behaviour</td>
<td>Bs</td>
<td>behaviour of the structure, extracted from structures</td>
<td>Camera (S) - Facial features/points are extracted</td>
</tr>
</tbody>
</table>
Design Strategies & Cognitive Processes Analysis - Glimpse

- FBS framework - Protocol based Linkograph analysis (Goldschmidt, 2013)
- Linkograph - areas of interest
- Zoom into the cognitive processes
Study 1 - Detail results

RQ 1.a Sample design strategies

Successful Group

Unsuccessful Group
## Conceptual design cognition in SCD (based on Hay et al, 2017)

<table>
<thead>
<tr>
<th>Category of Cognitive Process</th>
<th>Role in Design</th>
<th>Cognitive Processes and definition</th>
</tr>
</thead>
</table>
| Long-term memory             | Retrieving experiences or representations | Episodic retrieval – retrieval of previous experience  
Semantic retrieval – retrieval of type of product and function during concept generation |
| Creative output production   | Producing & combining concepts | Analogical Reasoning – process of using information about known semantic concepts to understand newly presented concepts  
Concept generation, i.e. the process of generating ideas for solutions/partial solutions to design problems  
Developing a solution based on the outcomes of actions taken to structure/restructure the problem during co-evolutionary design |
| Executive Functions          | Planning, monitoring & selecting | Problem structuring and analysis – Setting up goals and defining constraints  
Evaluating concepts - process of assessing concepts against design requirements, constraints, and other criteria  
Decision making - process of determining what concept(s) should be taken forward for further development from a range of alternatives  
Reasoning - process of developing a rationale for design decisions |
### Study 1 - Detail results

**RQ 1.b Cognitive Processes**

<table>
<thead>
<tr>
<th>Cognitive Process</th>
<th>Design Strategy</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Long-term memory</strong></td>
<td>Retrieval</td>
<td>generation of structures, functions</td>
</tr>
<tr>
<td><strong>Creative output production</strong></td>
<td>Analogy reasoning</td>
<td>generation of structures, functions, expected behaviours</td>
</tr>
<tr>
<td><strong>Executive Function</strong></td>
<td>Problem Structuring (defining goals)</td>
<td>generation of functions</td>
</tr>
<tr>
<td><strong>Executive Function</strong></td>
<td>Problem Analysis (constraints identification)</td>
<td>generation of functions, structures</td>
</tr>
<tr>
<td><strong>Creative output production</strong></td>
<td>Generating Concepts (mental visualisation)</td>
<td>generation of expected behaviour</td>
</tr>
</tbody>
</table>

- Cognitive processes - Conceptual design cognition (Hay et al., 2017)
- Zoom into the cognitive processes
- Deductive thematic analysis (Aronson, 1994)
Study 2 & 3 - Details
### Rubric for FBS graph evaluation based on Lindland et al. (1994)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Target Performance</th>
<th>Needs Improvement</th>
<th>Inadequate</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity</td>
<td>More than or equal to 12 nodes each for F, B, S.</td>
<td>Only 4 nodes each for F, B, S</td>
<td>Only 2 nodes each for F, B, S</td>
<td>Only a node each for F, B &amp; S</td>
</tr>
<tr>
<td>Levels</td>
<td>Two levels in the function, structure and behaviour subgraph are present in the FBS graph.</td>
<td>Only two levels in function and behaviour subgraph are present in the FBS graph.</td>
<td>Two levels in either function or behaviour subgraph are present in the FBS graph.</td>
<td>There are no levels in all the three-function, behaviour and structure subgraph.</td>
</tr>
<tr>
<td>Connectivity</td>
<td>All the nodes in the FBS graph are connected</td>
<td>Some of the nodes are connected but there exists nodes in the FBS graph that are not connected.</td>
<td>FBS elements are grouped together to form disconnected forests.</td>
<td>There are listing of FBS elements in the graph space</td>
</tr>
<tr>
<td>Semantic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validity</td>
<td>All FBS branches are unique, relevant to the problem and satisfy the problem requirements.</td>
<td>There are unique relevant FBS branches. However the problem requirements are not satisfied.</td>
<td>There are some repetitive FBS branches which do not satisfy the problem requirement.</td>
<td>All the FBS branches are repetitive, irrelevant and do not satisfy the problem requirement.</td>
</tr>
<tr>
<td>Consistency</td>
<td>A combination of FBS elements, subgraphs and branches are not contradictory to one another.</td>
<td>A combination of FBS elements and sub graphs are not contradictory but some FBS branches are contradictory.</td>
<td>The structure nodes are inconsistent.</td>
<td>All FBS elements are contradictory to one another.</td>
</tr>
<tr>
<td>Level Adjacency</td>
<td>All the adjacent pair of nodes in the graphs are at the same level</td>
<td>Only the nodes in F-F &amp; B-B are at the same level</td>
<td>Only the nodes in F-F are at the same level</td>
<td>At any level of F, B, S there exists no adjacent nodes at the same level</td>
</tr>
<tr>
<td>Pragmatism</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formal Realization</td>
<td>All the design elements of the FBS graph along with their relations are mappable to the appropriate formal representation (UML diagrams).</td>
<td>Only some of the design elements of the FBS graph along with their relations are mappable to the appropriate formal representation (UML diagrams).</td>
<td>Only the FBS design elements but not their relationship mappable to the appropriate formal representation</td>
<td>None of the FBS graph elements and their relations are mappable to the appropriate formal representations</td>
</tr>
</tbody>
</table>
Sample response to post-test

"Generation of an Adaptive Scratch programming System for students based on their selection of type of tutorials, activities done during the tutorials, grades, attendance and type of Practice sessions chosen. Based on this a particular type of Scratch programming session will be selected for every student."

05 July 2021, Ph.D. defence
FBS intervention II
Task 2 - FBS graph of a participant

User's profile is linked with social networking sites
User's social networking credentials are requested
User's fingerprints are recorded and stored
User's fingerprints scanned multiple times
User face scanned multiple times
User's access credentials are stored
User answers questions

Login ID

Biometric

Login Screen comes up

User places finger on the scanner
User's fingerprints scanned
User's fingerprints compared with existing data

User is provided retry options

User types in login ID and password
User ID and password is compared with stored data

User access given/denied
Sample response to post-test
Study 4 & 5 - Details
## What is the expected output of SCD? (Eckerdal et al., 2006)

<table>
<thead>
<tr>
<th>Category #</th>
<th>Category</th>
<th>Content (Indicators)</th>
<th>Representation (indicators)</th>
</tr>
</thead>
</table>
| 0          | Nothing          | Little or unintelligible content                                                    | Single labelled diagram
                                                        Informal design                                                                         |
| 1          | Restatement      | * Restate requirements from task description
                                                        * No design content other than stated in the description                              | List or Bulleted items
                                                        Informal design                                                                         |
| 2          | Skumtomte        | * Add a small amount to restating task
                                                        * Unimportant implementation details
                                                        * No overall system view and any work on modules                                         | Simple GUI
                                                        Notations such as flow chart                                                             |
| 3          | First step       | * Some significant work beyond restatement                                           | Formal notation representing structure
                                                        Design of one of the system’s components like GUI or Database                           |
| 4          | Partial design   | * Understandable description of parts and overview
                                                        * Description of parts maybe incomplete or superficial
                                                        * Communication between parts may not be completely described                           | Formal notation representing behaviour
                                                        Illustration of relationship between the parts                                            |
| 5          | Complete Design  | * Well developed solution
                                                        * Understandable overview
                                                        * Solution parts description includes explicit communication between them
                                                        * Formal representations as well as text                                               | Multiple formal notations such as Use case, Class diagram, component diagram             |
Event logging and sequence extraction

• What all gets logged in ‘think & link’?

• **Click on a menu/feature button is an event** and gets logged

• **Internal events** such as - worksheet saved, phase completed also logged

• **A logging row** : log_id, user_id, phase, subphase, subsubphase, event, event_data, event_time, session_id, log_type, temp3

• **Relevant columns** : log_id, user_id, phase, subphase, subsubphase, event, event_time

• **Action abstraction with context summarisation ‘event’** - combining columns : phase, subphase, subsubphase, event

• **Introduction, context, intro, reading problem** - introductioncontextintroreadproblem

• **For each phase we have user_id based entries of** - log_id, user_id, event_time, event

• **TraMineR (Trajectory miner) package in R**
#using the library#
library(TraMineR)

#setting the workspace#
setwd("~/Documents/Lakshmi/Seminar/Learning Analytics/SAKEC/")

#reading the source file#
mvad <- read.csv(file = "tse-sequence-intro.csv", header = TRUE)

#creating a time stamped event sequence#
mvad.seqe <- seqecreate(id=mvad$user_id,timestamp = mvad$event_time, event = mvad$event)

#extracting subsequences found in 50% cases with 4 as number of events in a window#
mvad.subseqee <- seqefsub(mvad.seqe,pmin.support=0.5, max.k = 4)

#writing subsequences into a file#
df <- mvad.subseqee$data
df$subseq <- as.character(mvad.subseqee$subseq)
write.csv(df,'subsequences-intro.csv')

#setting screen size#
par(mar=c(4,15,2,1))

#ordering successive sequences#
seqpcplot(mvad.seqe,
  filter = list(type = "function",
                value = "cumfreq",
                level = 0.8),
  order.align = "last",
  ltype = "non-embeddable",
  cex = 1.5, lwd = .9,
  lcourse = "downwards")
For RQ 3.b, the seqefsub() parameters utilised - time stamped event sequence, pmin.support & max.k

- pmin.support - the minimum occurrence of subsequence in cases
- max.k - maximum number of events allowed in a subsequence (sequence length to be analysed)

Counting method - support is counted per sequence and not per occurrence, i.e. when a sequence contains several occurrences of a same subsequence it is counted only once.

Prefix-tree-based search described in Masseglia (2002)

The algorithm was designed for a small number of event per sequence (<6 typically) and many sequences (Stackoverflow - https://stackoverflow.com/questions/28770833/speeding-up-identification-of-subsequences)

Output - An event sequence is an ordered list of transitions. Represented as a succession of transitions separated by edges or arrows

RQ 3.d - Output of seqcplot()

- The input to this function is the time stamped sequence created from seqecreate() function.

- This function renders the order of the successive elements in sequences that are shared by at least 5% of the observed cases.

- Frequencies of events and embedded sequences with varying width.

### Introduction - most frequent event sequence path

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Count*</th>
<th>Support**</th>
</tr>
</thead>
<tbody>
<tr>
<td>(introduction,introductioncontext)-(introductioncontextintrotvideo)-</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(introductioncontextintroformsub)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(introductiongraphtask,introductionworksheettask)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(introduction)-(introductioncontextintroformsub)-(introductiongraphtask,</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>introductionworksheettask)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(introductionworksheettask)-(introductiongraphtask)-(</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>introductionworksheettask)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* indicates the number of cases in which the event sequence is found  
** indicates the strength of the sequence across cases

**All participants utilise the conjectured features for abstracting the FBS conceptual model**
Comparison of semantic interpretation of FBS design elements

<table>
<thead>
<tr>
<th>Post-test category</th>
<th>Abstraction of relationship of FBS (representative)</th>
</tr>
</thead>
</table>
| Restatement (n=2)  | • *Function Implements Structure*, structure is utilized to achieve the Behaviour, Structure demonstrates the Behaviour which is implemented using function  
• *Function consists Function, Structure implemented by Behaviour*, Function combines Structure, Function represented Structure |
| Partial design (n=14) | • *mood detection implemented by user speaks for mood detection implemented by voice input screen consist of mike used by end user*  
• *Function is achieved by Structure utilized by Behavior*  
• *Function is implemented by Structure which gets utilized during user Behaviors* |
| Complete design (n=4) | • *Structure consist of function & implemented by behaviours*  
• *Functions are implemented by structures which utilize behavior, Behavior combines with structure to implement functions.* |
## Induction - most frequent event sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Count*</th>
<th>Support**</th>
</tr>
</thead>
<tbody>
<tr>
<td>(induction, inductioncontext)-(inductioncontextintrofeedbacksub)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(induction)-(inductioncontextintrofeedbacksub)-(inductiongraphtask)-(inductionevaltask)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(inductiongraphtask)-(inductionevaltask)-(inductioneval)-(inductionphasefin)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(inductiongraphtask)-(inductionevaltask)-(inductionumlintro)</td>
<td>20</td>
<td>1</td>
</tr>
</tbody>
</table>

* indicates the number of cases in which the event sequence is found
** indicates the strength of the sequence across cases

All participants utilise the conjectured features for evaluation of FBS graph
### Comparison of event subsequences

<table>
<thead>
<tr>
<th>Post-test category</th>
<th>Event subsequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restatement</td>
<td>(induction)-(inductioncontexttask)-(inductionevaltask)-(inductionphasefin)</td>
</tr>
<tr>
<td>Partial design</td>
<td>(inductiongraphintro)-(inductiongraphtask)-(inductionumlintro)-(inductionphasefin)</td>
</tr>
<tr>
<td></td>
<td>(inductiongraphtask)-(inductioneval)-(inductionevaltask)-(inductionphasefin)</td>
</tr>
<tr>
<td>Complete design</td>
<td>(inductiongraphtask)-(inductionevaltask)-(inductiongraphtask)</td>
</tr>
<tr>
<td></td>
<td>(inductiongraphtask)-(introductiongraphtask)-(inductiongraphtask)</td>
</tr>
<tr>
<td></td>
<td>(introductionworksheettask)-(inductioncontext)-(inductioneval)-(inductiongraphtask)</td>
</tr>
</tbody>
</table>

**Participants in post-test creating**
- informal designs do not edit the graph in induction phase
- only behaviour based representations edit graph and then evaluate, however while examining their edits it is only addition of either a function or behaviour
- multiple integrated representations move back & forth between evaluation & graph edit tasks. They also move across the phases introduction & induction
### Ideation - most frequent event sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Count</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ideation)-(ideationcontextintrofeedbacksub)-(ideationgraphtask)-(ideationevaltask)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(ideationgraphtask)-(ideationevaltask)-(ideationumlintro)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(ideationcontextproblemread)-(ideationcontextproblemsaved)-(ideationgraphtask)-(ideationevaltask)</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>(ideationcontextproblemsaved)-(ideationgraphtask)-(ideationevaltask)</td>
<td>20</td>
<td>1</td>
</tr>
</tbody>
</table>

* indicates the number of cases in which the event sequence is found
** indicates the strength of the sequence across cases

All participants utilise the conjectured features for editing problem, graph and completing evaluation of FBS graph
Comparison of event subsequences

<table>
<thead>
<tr>
<th>Post-test category</th>
<th>Event subsequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restatement (n=2)</td>
<td>(ideation)-(ideationgraphtask)-(ideationevaltask)-(ideationphasefin)</td>
</tr>
<tr>
<td>Partial design (n=14)</td>
<td>(inductionevaltask)-(ideationevaltask)</td>
</tr>
<tr>
<td>Complete design (n=4)</td>
<td>(inductiongraphtask)-(ideationgraphtask)-(ideationevaltask) (ideation,ideationcontext)-(ideationcontextproblemsaved)-(inductiongraphtask)</td>
</tr>
</tbody>
</table>

Participants in post-test creating
- informal designs follow linear progression of tasks
- only behaviour based representations refer to evaluation done in previous phase to complete evaluation in this phase
- multiple integrated representations move back & forth between problem setting, graph edit & evaluation tasks. They also move across the phases induction & ideation
Guidelines for teacher to teach with Think & Link

• With ‘think & link’

• A second/third year laboratory class, after learners have been exposed to UML representations

• A final year project class in lab for learners to create conceptual design of final year project

• Without ‘think & link’

• Concept - Ideas to UML representations, UML representations are linked, generate them together rather than in isolation
What are the statistical tests for Evaluation of Pre-Post learning gain?

- Single Group pre-post test
- The Wilcoxon test, which refers to either the Rank Sum test or the Signed Rank test, is a nonparametric statistical test that compares two paired groups
- As the nonparametric equivalent of the paired student's t-test, the Signed Rank can be used as an alternative to the t-test when the population data does not follow a normal distribution
- The model assumes that the data comes from two matched, or dependent, populations, following the same person or stock through time or place