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)

Quality parameters	Goals
Syntactic	• Syntactic correctness
Semantic	• Feasible validity • Feasible completeness
Pragmatic	• Feasible comprehension

- 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)

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

Undesirable categories

Desirable categories

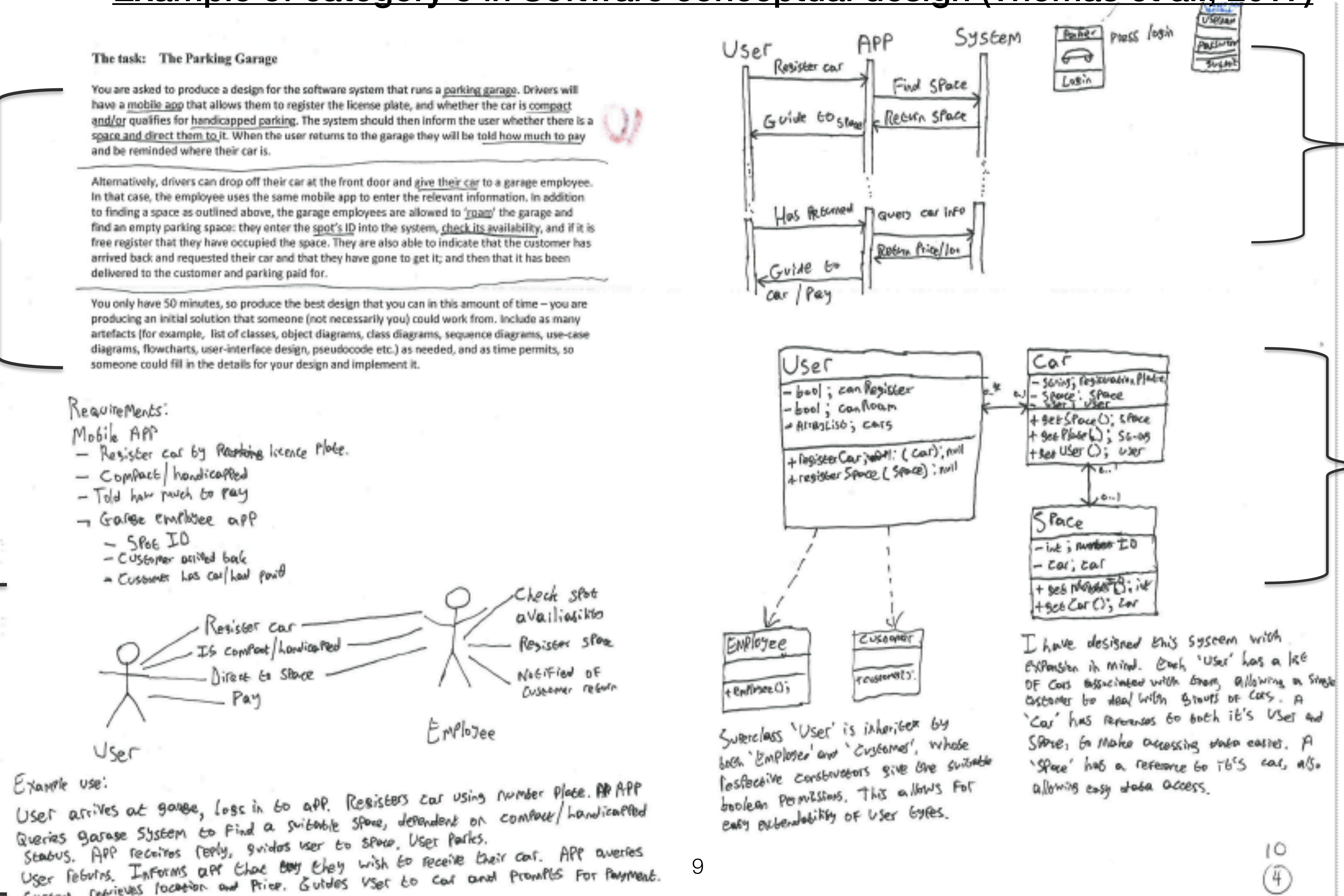
Example of category 5 in Software conceptual design (Thomas et al., 2017)

Problem

Sequence diagram

Class diagram

Use case


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 plate, and whether the car is compact and/or qualifies for handicapped parking. The system should then inform the user whether there is a space and direct them to it. When the user 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 employee uses the same mobile app to enter the relevant information. In addition to finding a space as outlined above, the garage employees are allowed to 'roam' the garage and find an empty parking space: they enter the spot's 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 parking paid for.

You only have 50 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 artefacts (for example, list of classes, object diagrams, class diagrams, sequence diagrams, use-case diagrams, flowcharts, user-interface design, pseudocode etc.) as needed, and as time permits, so someone could fill in the details for your design and implement it.

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

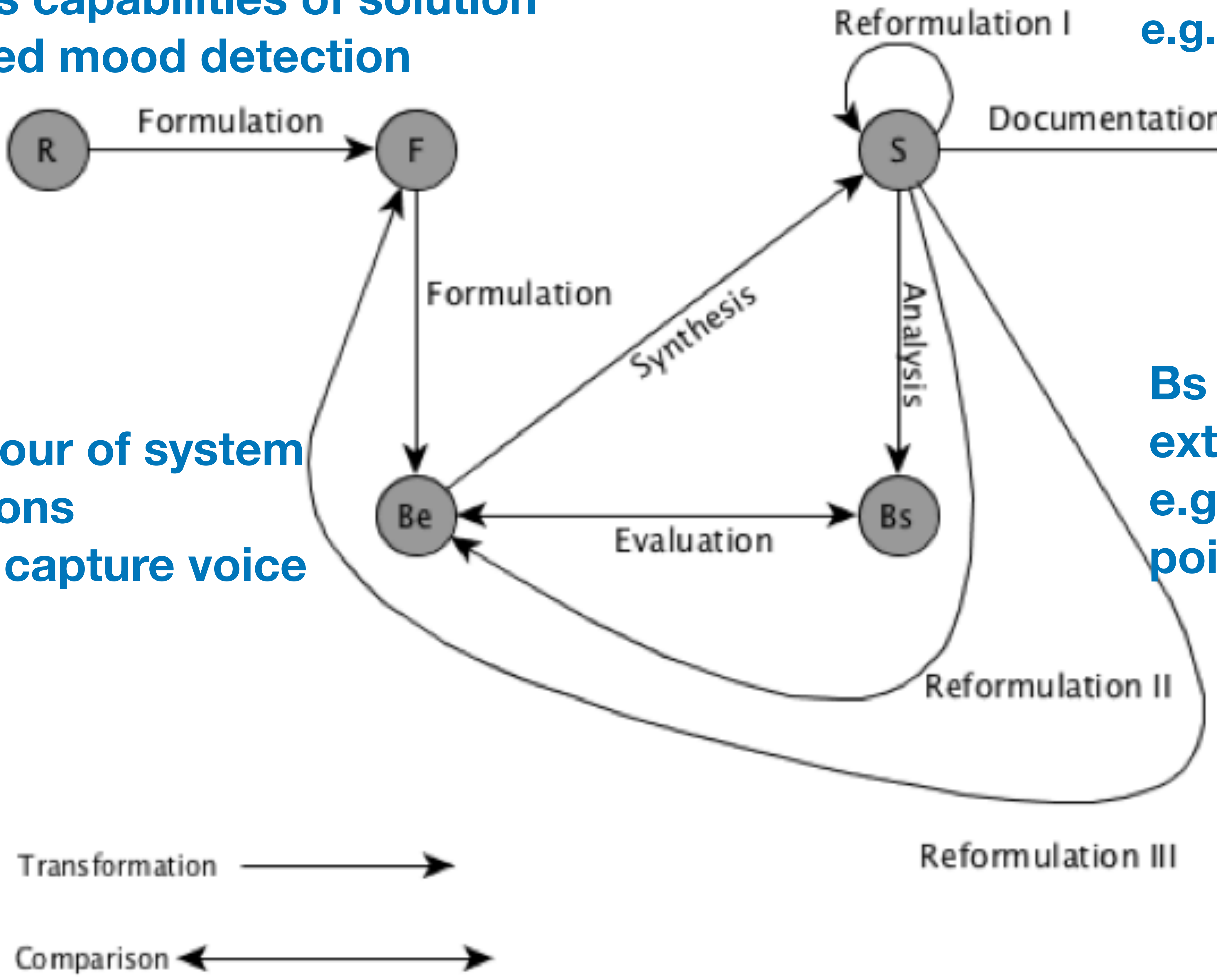
Function-Behaviour-Structure (FBS) Design Framework

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

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

Be - Expected behaviour of system extracted from functions
e.g. System needs to capture voice

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 d… nt (Christophe et al., 2010), Th…
- Supports Integ…
  - Unified Mode… of view
  - Need for unif… dentified (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

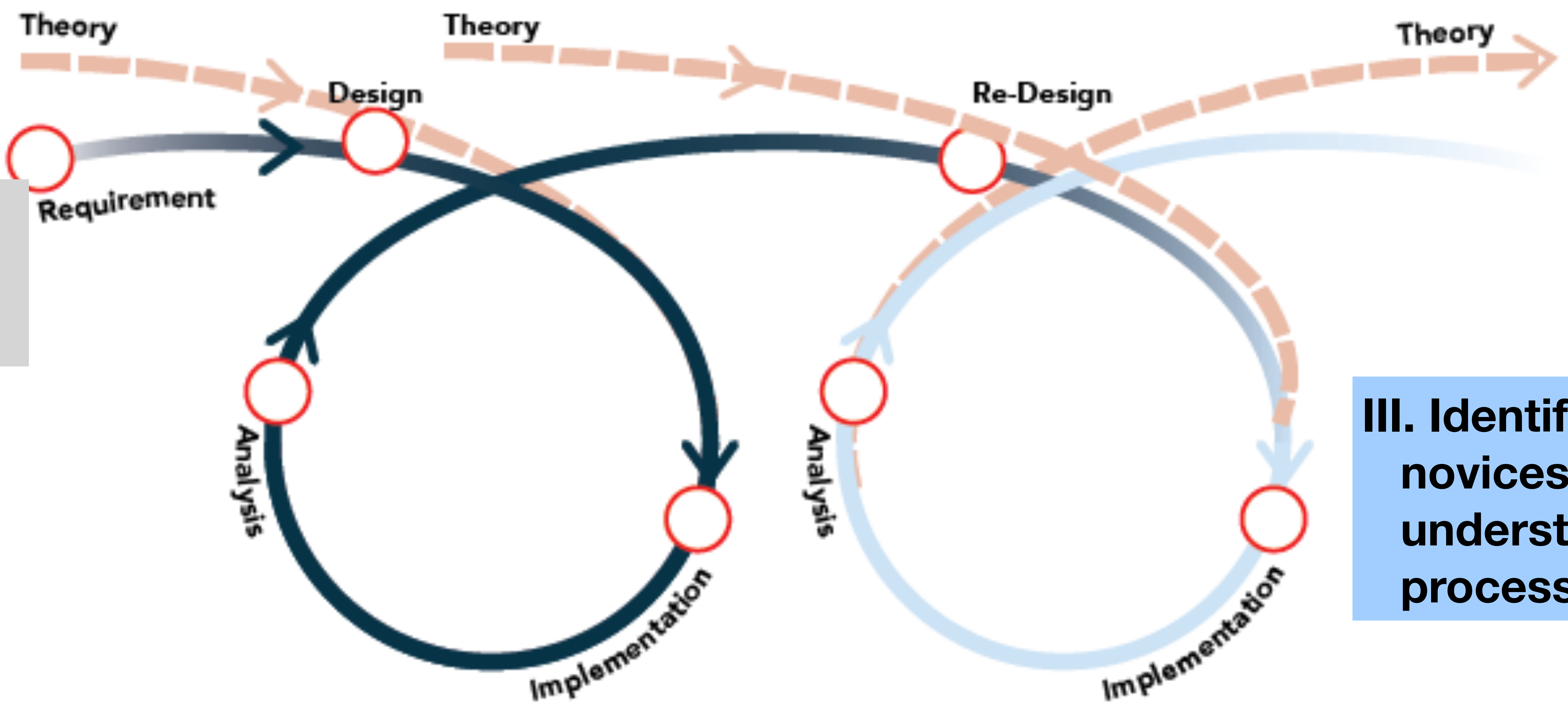
Theoretical Framework

Methodological Framework

Design-Based Research

- Iterative
- Integrates variety of research methods
- Pragmatic

I. Unpacking novices' design strategies & cognitive processes



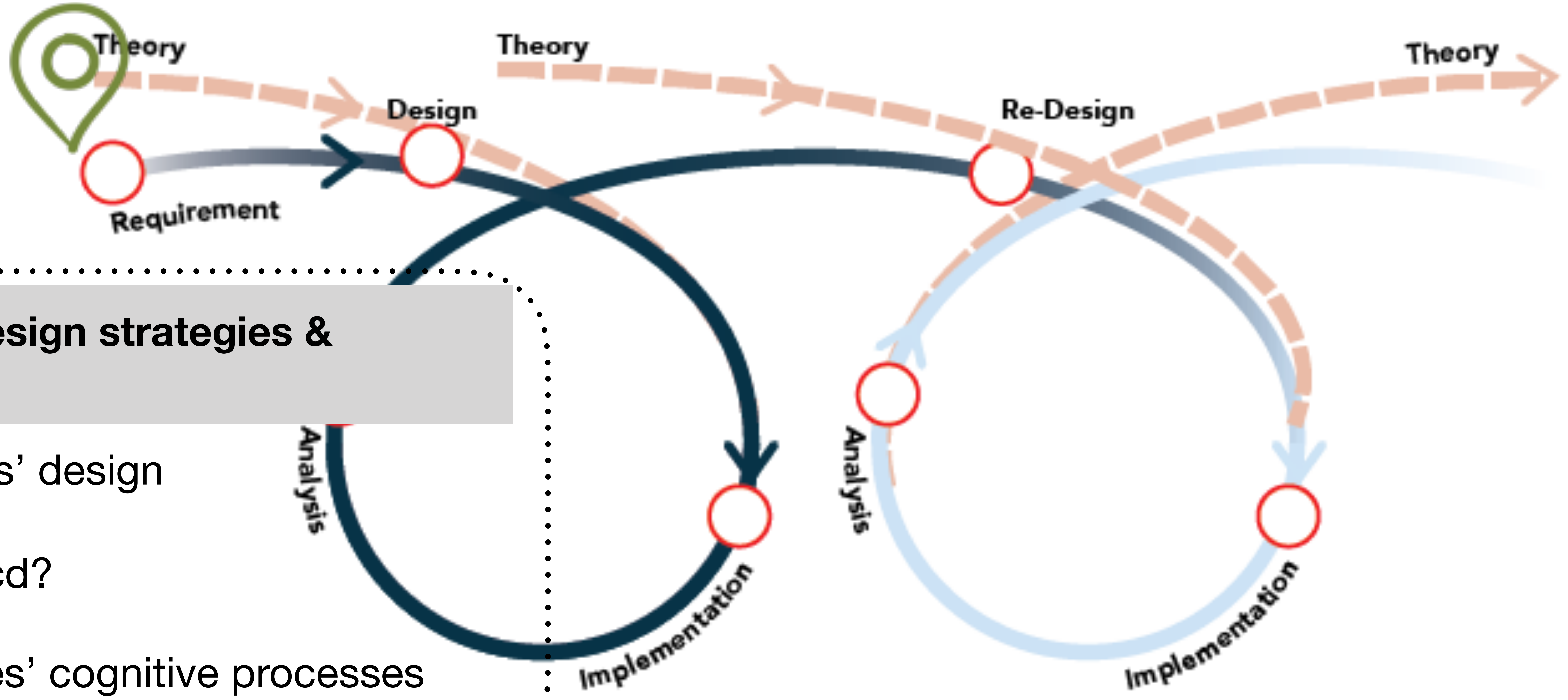
III. Identifying changes in novices' SCD understanding & process

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

Reference - Barab, S. (2014). Design-based research: A methodological toolkit for engineering change. In The Cambridge Handbook of the Learning Sciences, Second Edition. Cambridge University Press.

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

Design-Based Research



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

Study 1

Research Question	Methodology	Data Collection	Analysis
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)

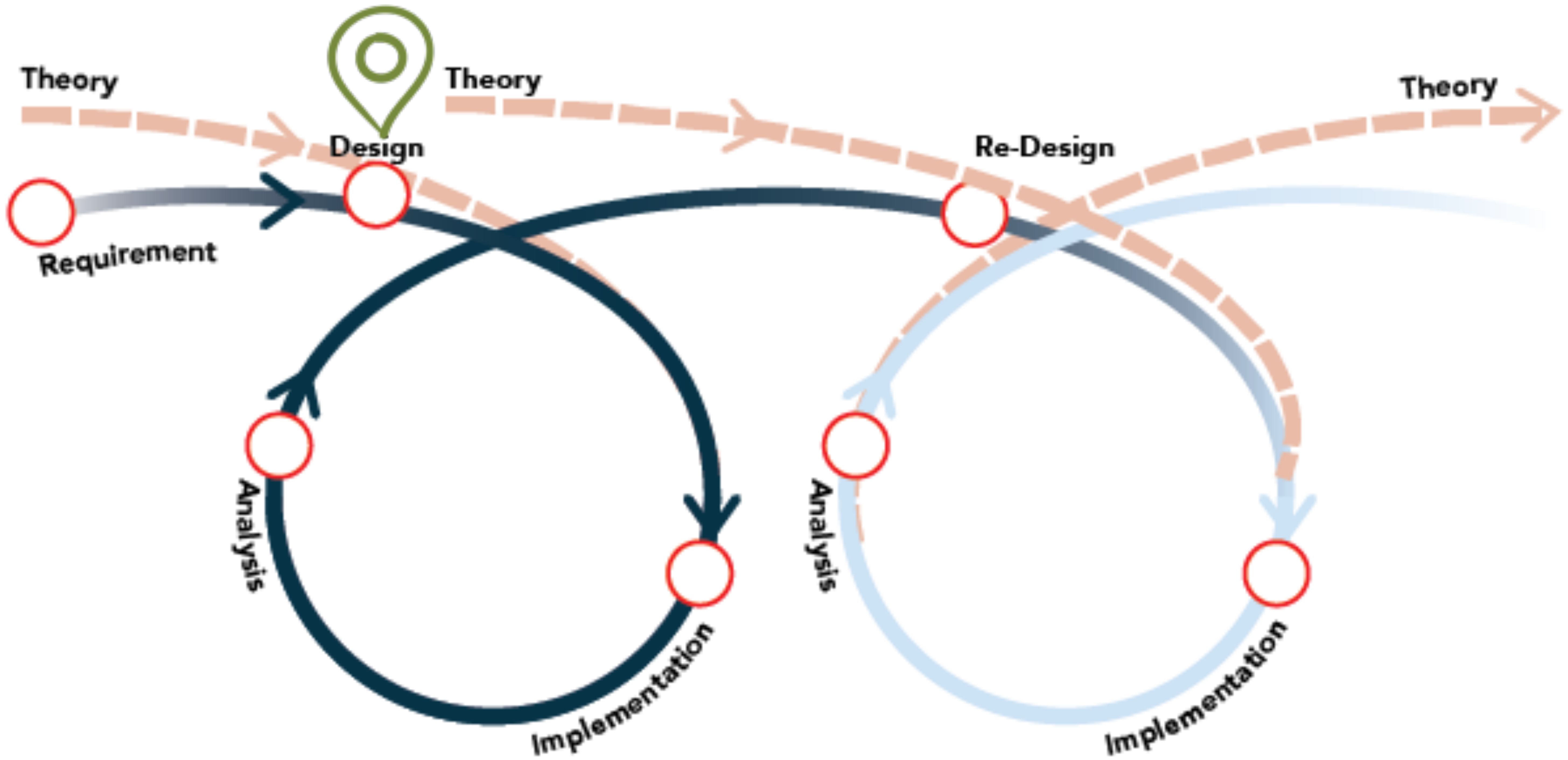
Summary of Results of RQ1

Study 1

Novice group	Results RQ 1.a. …	Results RQ 1.b. What are novices' cognitive processes while creating SCD?
Unsuccessful novices (category 1-3)	…on …s/	• information seeking
Successful novices (category 4 & 5)	…s	• mental simulation, association, analogical reasoning and synthesis

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

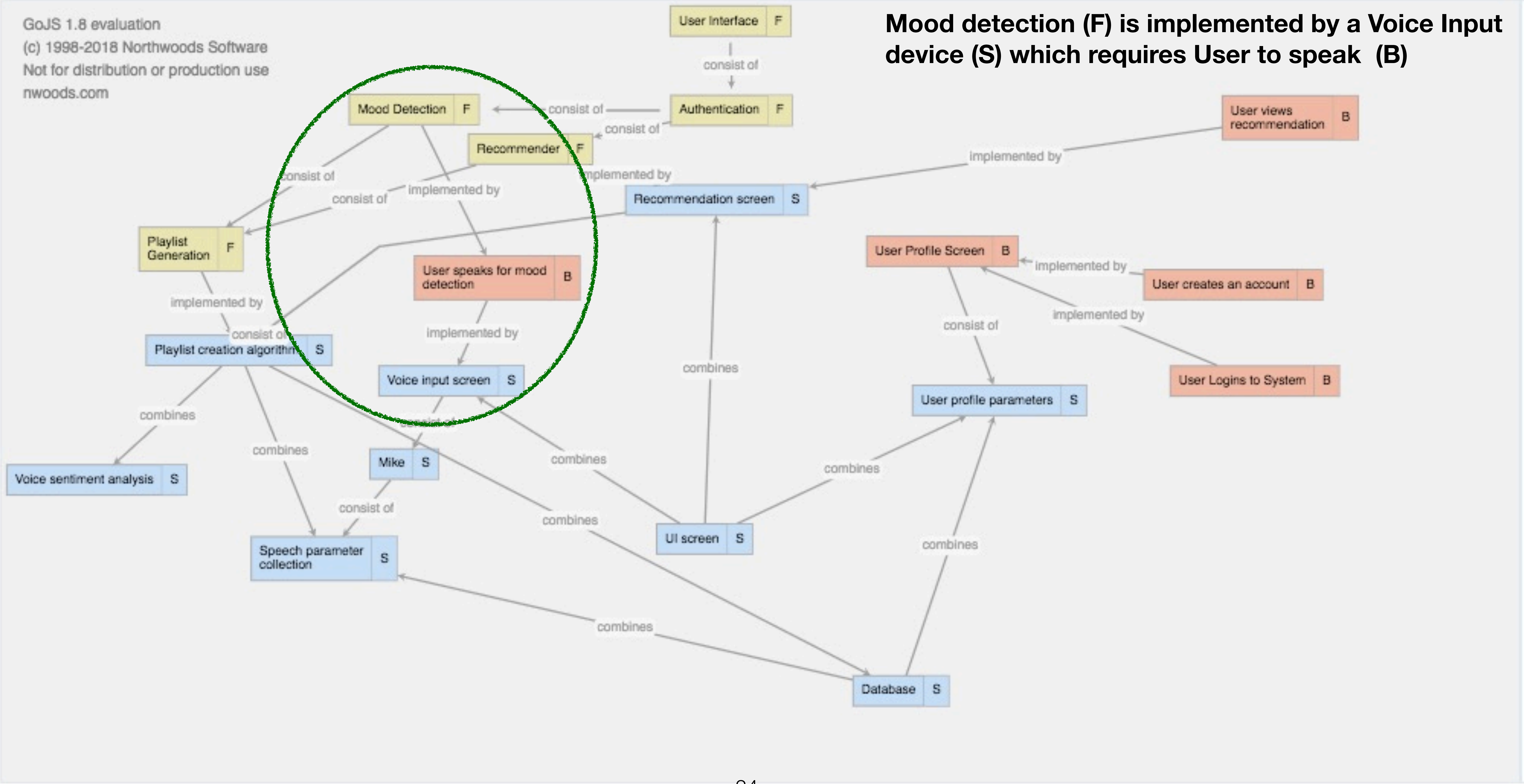
Design-Based Research



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)

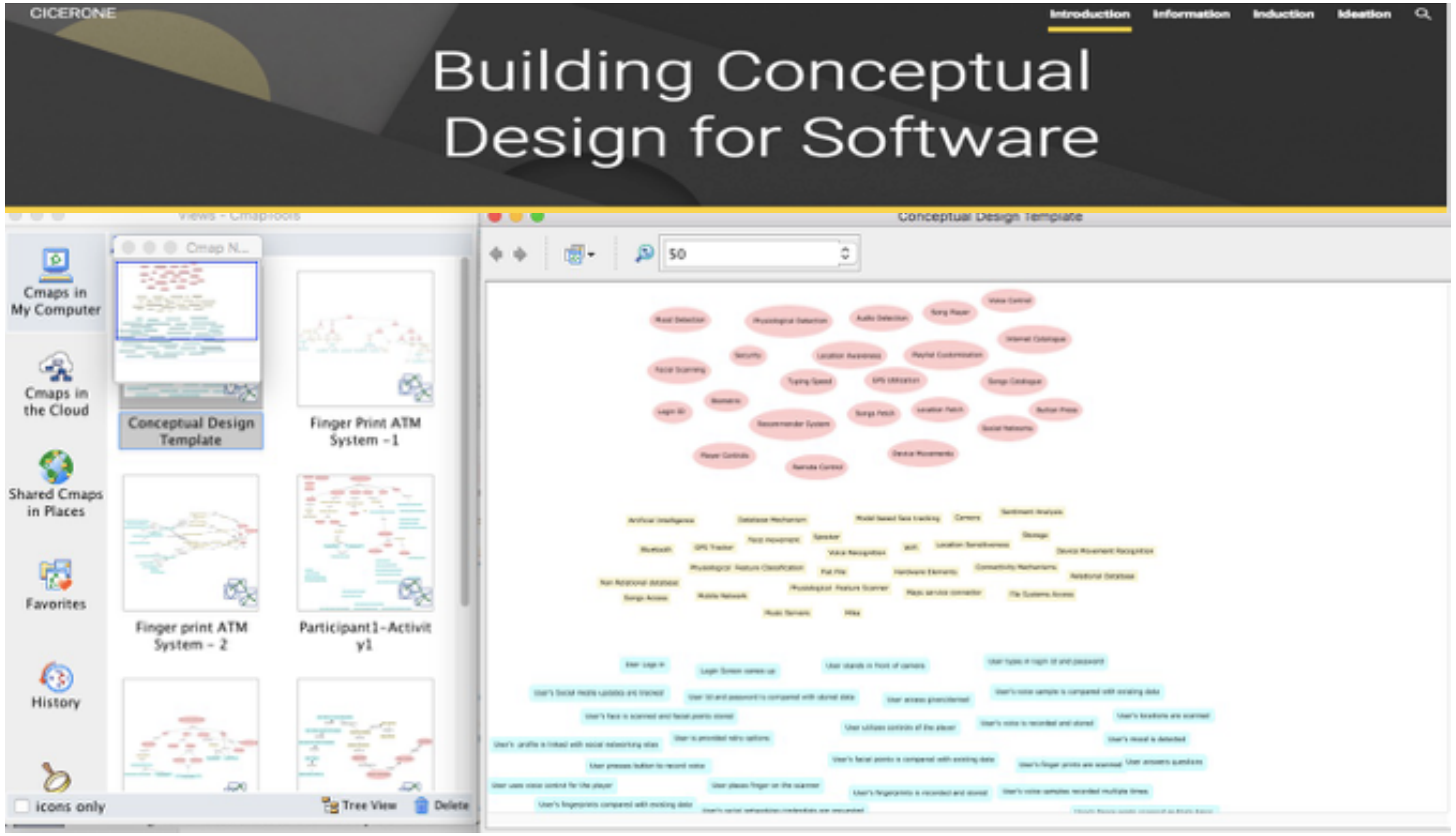


Learning Objectives for FBS graph based pedagogy

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

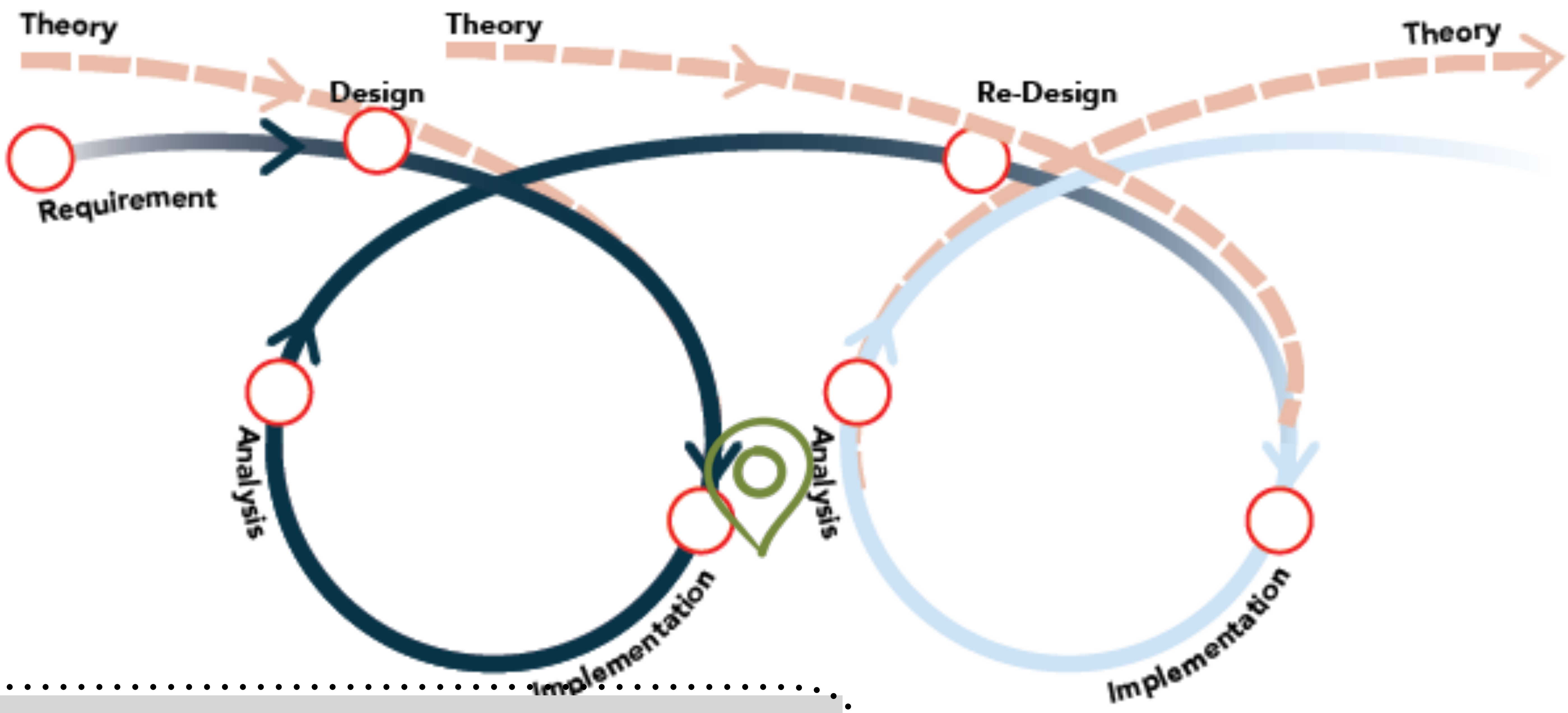
FBS graph based intervention- I & II



combination of a webpage and IHMC CMAP tool

- 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

Design-Based Research



Study 2 & 3

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

Study 2 & 3

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

Findings from Study 2 & 3 leading to features required in TELE

Difficulty from Study 2	Difficulty from Study 3	Features required in TELE
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

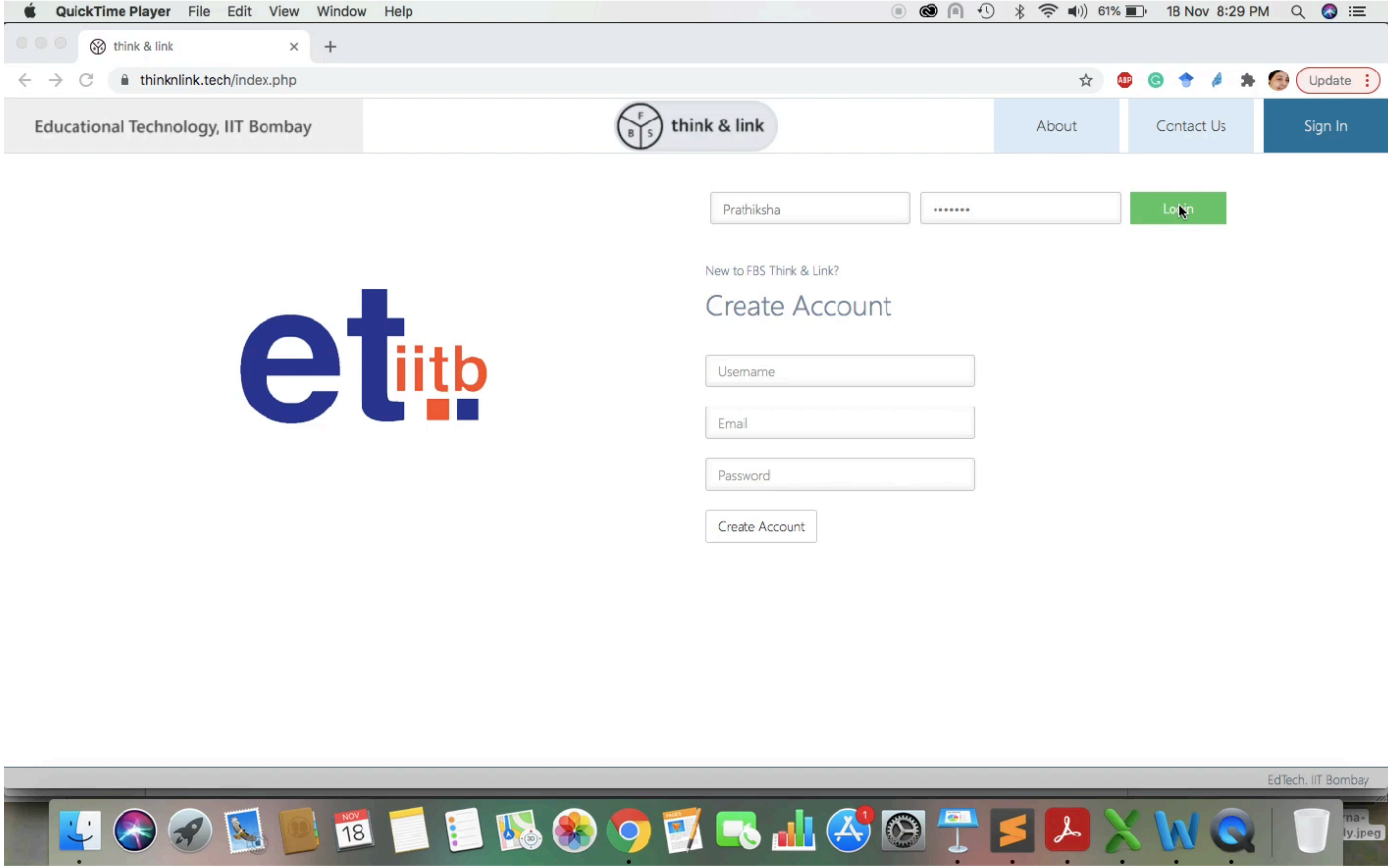
Sample learner login

Username : Prathiksha
Passwd : seokjin

Teacher/instructor login

Username : etiitb
Passwd : thinknlink2019

‘think & link’ - Demo



FBS graph based pedagogy in ‘think & link’

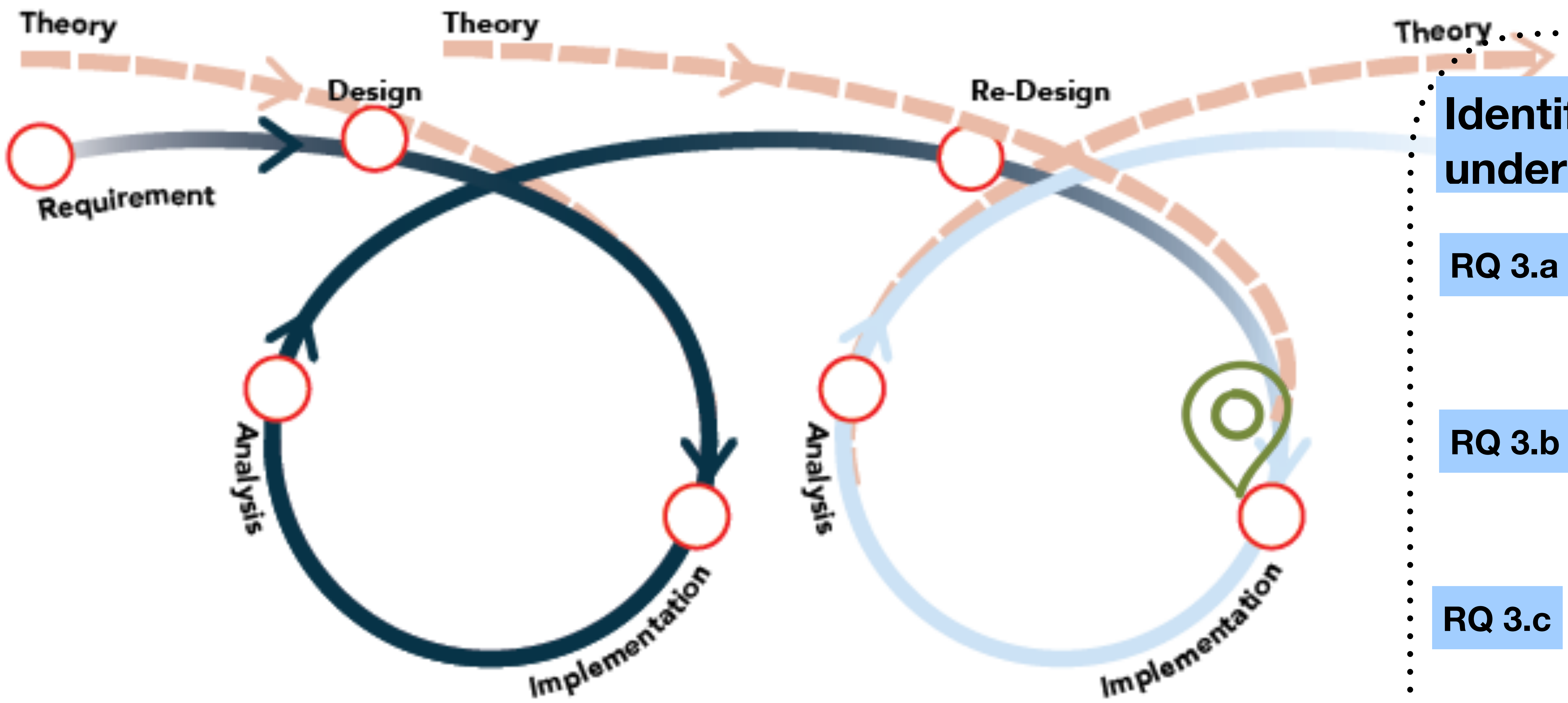
- Learners should be taken through progressive planes of cognition doing, evaluation, synthesis

- 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

Design-Based Research



Study 4 & 5

III. Identifying changes in novices' SCD understanding & process

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?

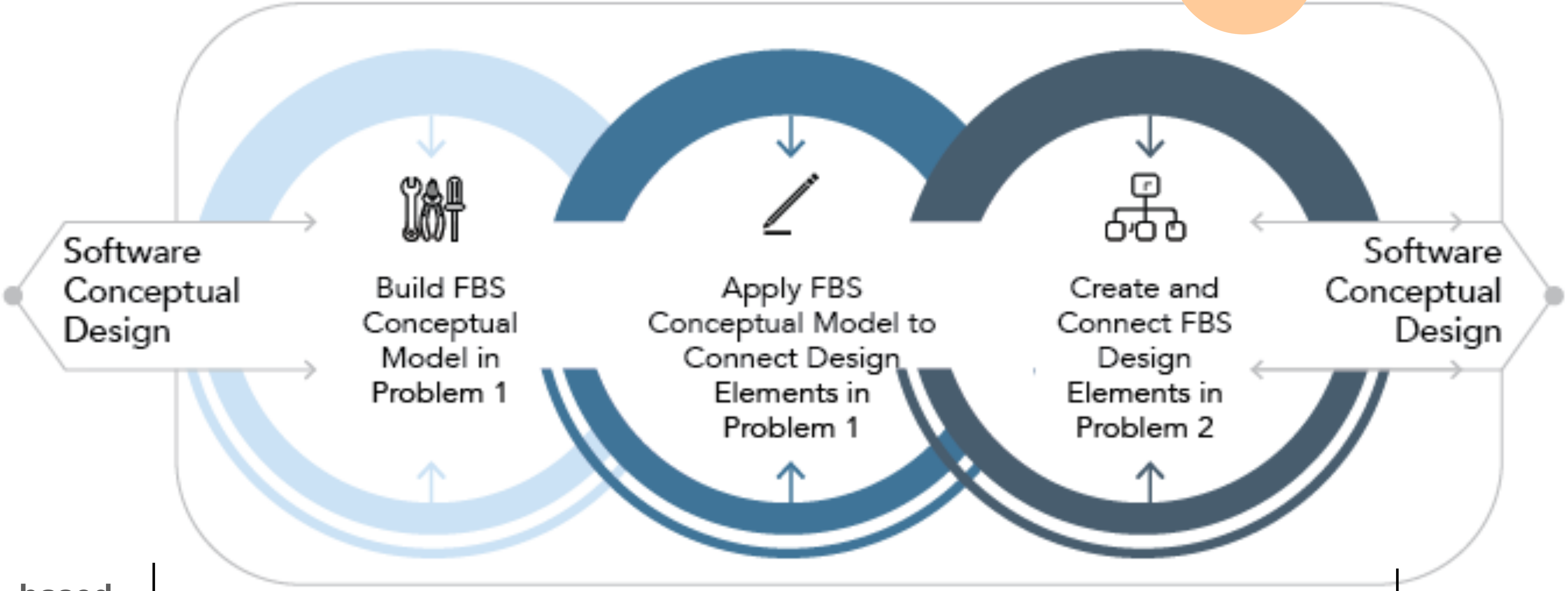
RQ 3.c What changes in process of creating SCD do the learners' perceive?

RQ 3.d How do the learners' use the features in TELE?

Motivation | Research Gap | Solution Approach | DBR Cycle 1 | **DBR Cycle 2** | Discussion

Study Design 4 & 5

Study 4 & 5



Questionnaire	Mood- based music player **Pre-test ~ 1 hr**	Mood- based music player **1.5 hr**	**1.5 hr**	Self posed problem **1.5 hr**	Finger print based ATM system **Post-test ~ 1 hr**	Questionnaire Focus Group Interviews

Retrospective Interview (study 5)

Retrospective Interview (study 5)

Retrospective Interview (study 5)

RQ 3.a - Method

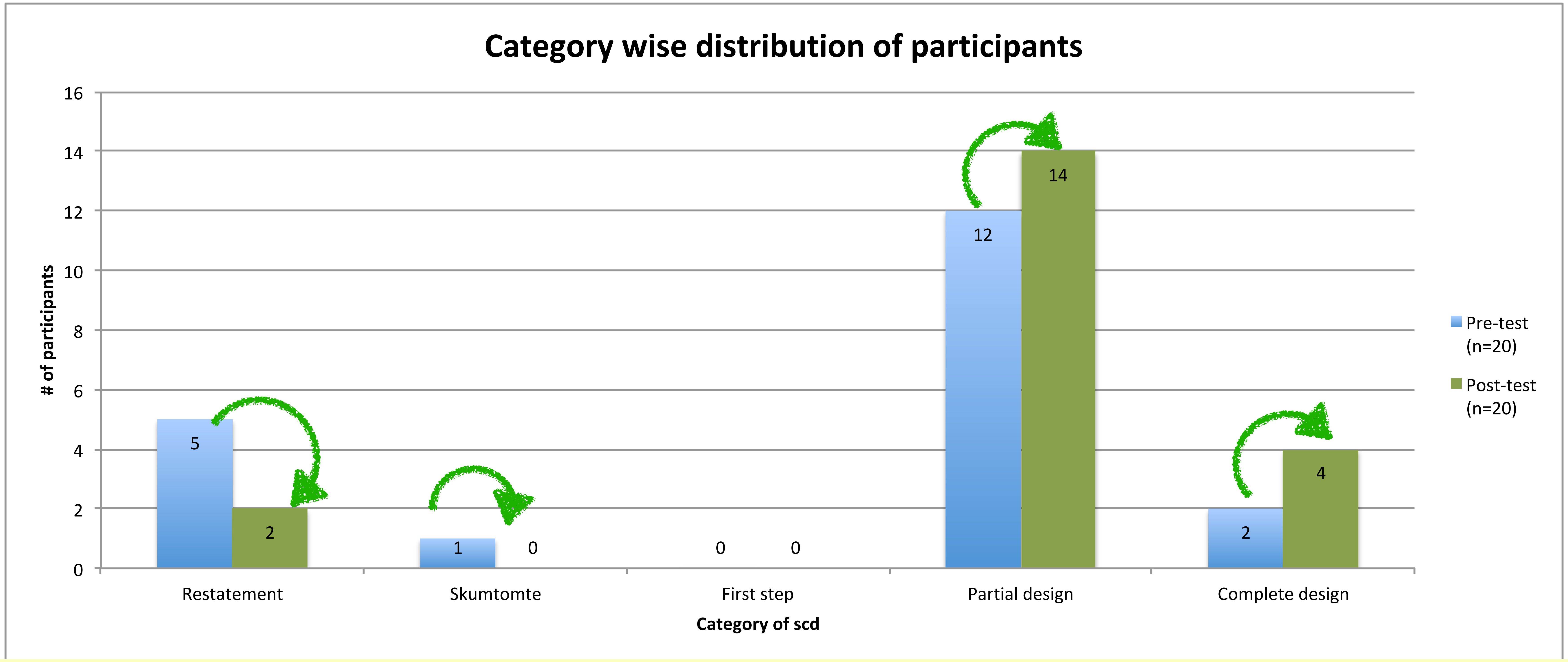
Research Question	Methodology	Analysis
RQ 3.a - What are the categories of SCD that learners' create?	Workshop study • N=20 (study 4) • N=18 (study 5)	• Categories of SCD (Eckerdal et al., 2006)

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

Results - RQ 3.a

Study 4

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



In post-test

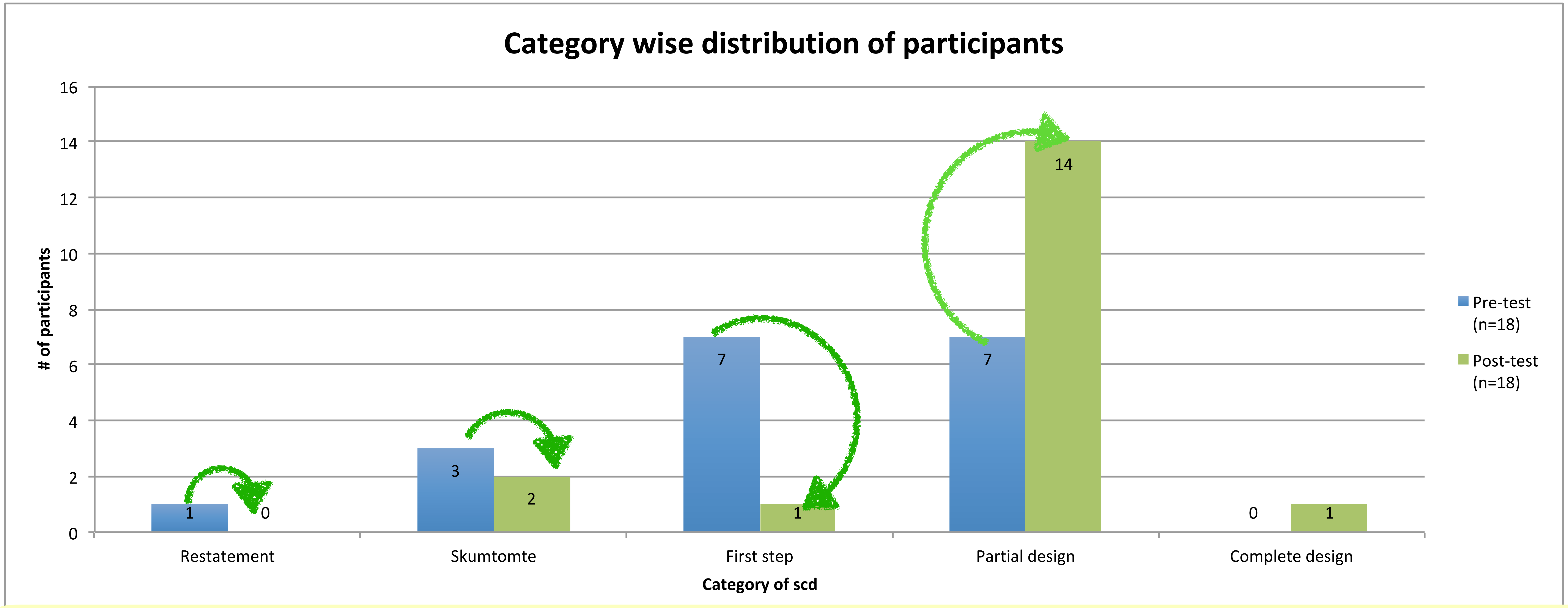
- slight increase in artifacts categorised in partial design & complete design
- slight decrease in artifacts categorised in restatement, skumtomte, first step

Motivation | Research Gap | Solution Approach | DBR Cycle 1 | DBR Cycle 2 | Discussion

Results - RQ 3.a

Study 5

<u>RQ 3.a What are the categories of SCD that learners' create?</u>



Confirms study 4 findings

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

Study 4 & 5

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

Results - RQ 3.b

Study 4 & 5

RQ 3.b What is the difference in learners' understanding of SCD?

RQ 3.c - Method

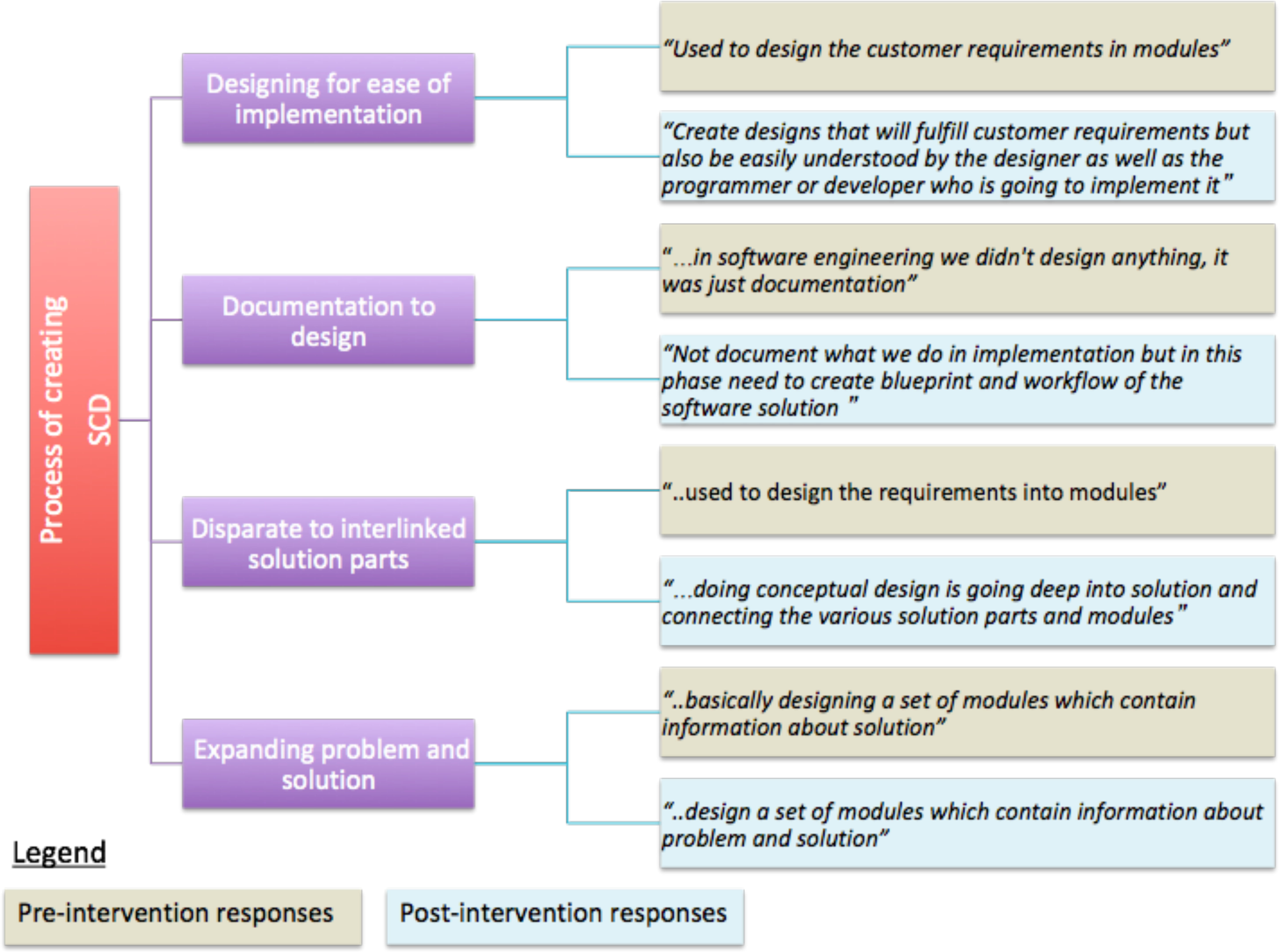
Study 4 & 5

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

Study 4

Results - RQ 3.c

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



Results - RQ 3.c

Study 5

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

- **Process of creating SCD**
  - **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**
    - "..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 you so 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 to serve that purpose, what are the logical components will I needs."*

Legend

Pre-intervention responses | Post-intervention responses

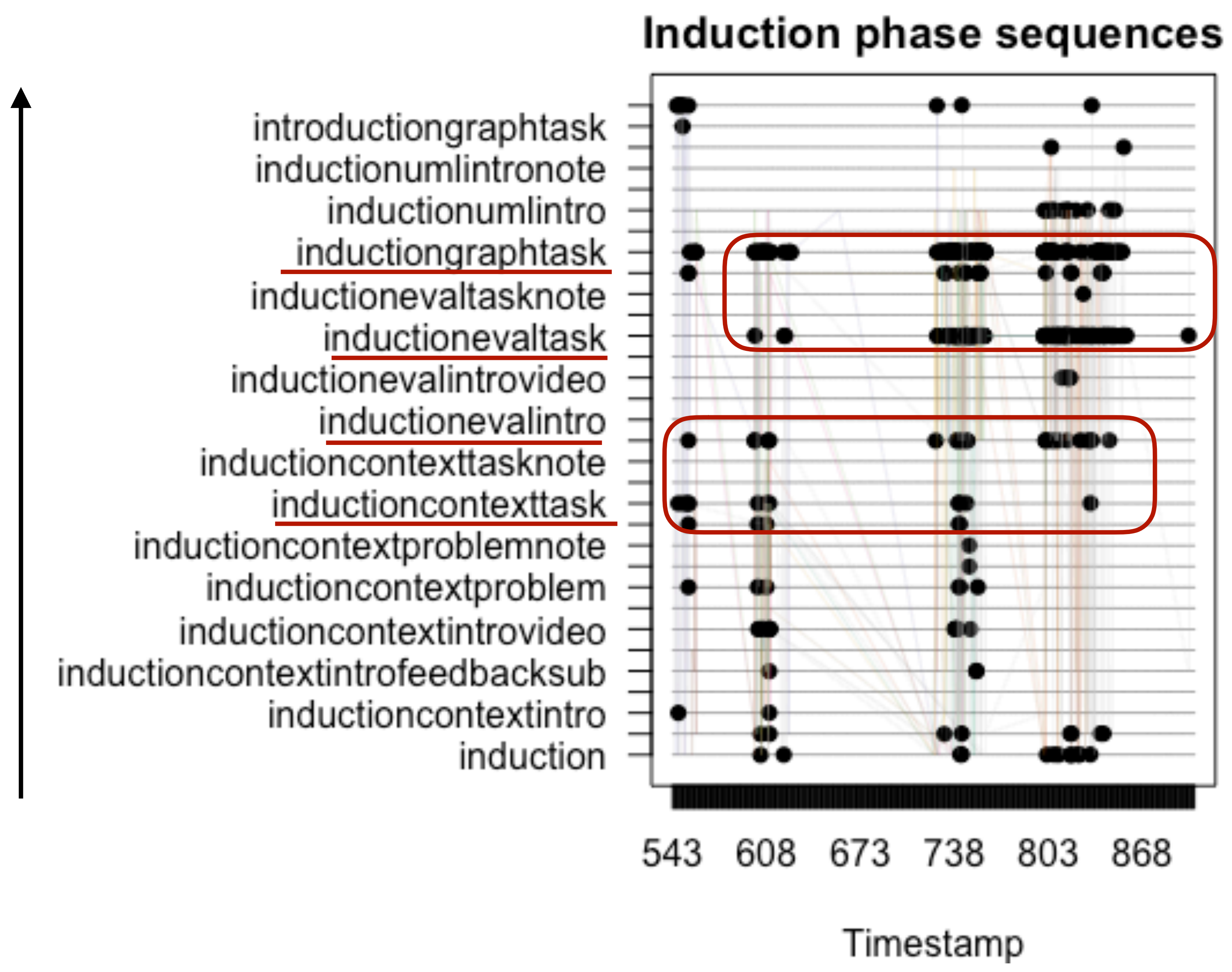
RQ 3.d - Method

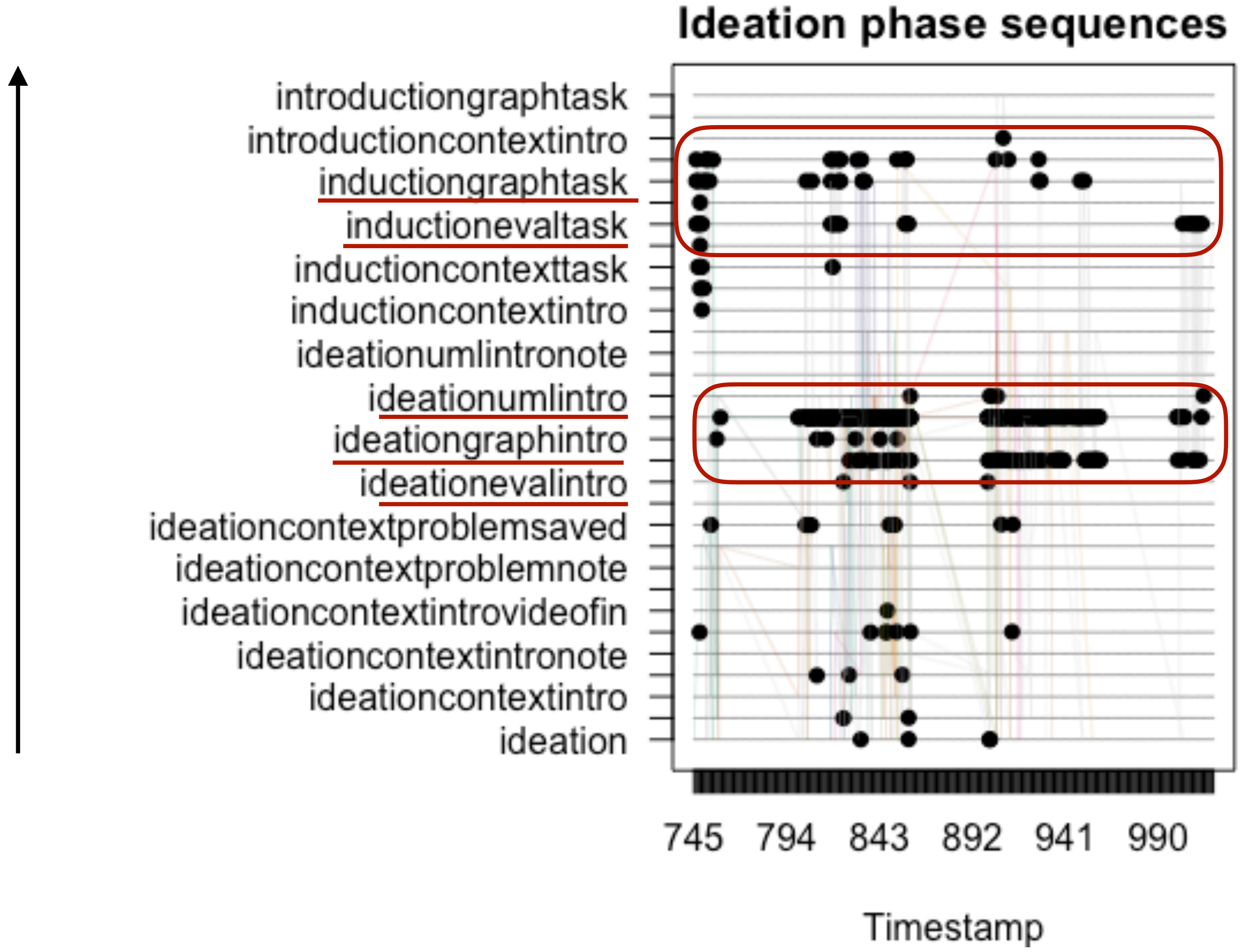
Study 4 & 5

Research Question	Methodology	Participants	Data Collection	Analysis
RQ 3.d -How do the learners' use the features in TELE?	Workshop study • N=20 (study 4) • N=18 (study 5)	Study 4 - Final year computer engineering students Study 5 - Second year computer and information technology students	• Participant actions and events recorded in the system	• Event sequence mining in R (Ritschard et al, 2013)

Inferences from participant event sequences

Study 4 & 5





- 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

Study 4 & 5

Phases in 'think & link'	Informal design category (1, 2 & 3)	Partial design category (4)	Complete design category (5)
II	do not edit the graph in this phase	edit graph and then evaluate, however while examining their edits reveals only addition of either a function or behaviour	move back & forth between evaluation & graph edit tasks. They also move across the phases I & II
III	follow linear progression of tasks	refer to evaluation done in previous phase to complete evaluation in this phase	back & forth between problem setting, graph edit & evaluation tasks. They also move across the phases II & III

Discussion

Claims of this thesis

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

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

Contributions of this thesis

Contributions	Implications for
Characterisation of novices’ design strategies and cognitive processes while creating software conceptual design	Researchers in computing education research, learning science and design education
Identified a set of features and scaffolds for novices teaching-learning of FBS based software conceptual design	Instructional designers and software engineering educators
Pedagogical design of a FBS based learning environment for teaching-learning of software conceptual design	Instructional designers and software engineering educators
Identified the usage of features in the learning environment by engineering undergraduates	Instructional designers, Researchers in building TELE
think & link is an instantiation of the FBS based pedagogy. A teacher authoring tool for different FBS graph contexts.	Software engineering students and software engineering educators

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

- Lakshmi, T.G. , & Iyer, S. (2020, Jul). Teaching-learning of software conceptual design via function-behaviour-structure framework.13th Workshop on cooperative and human aspects of software engineering:(CHASE 2020). In 2020 42nd International Conference on Software Engineering (ICSE). IEEE.
- Lakshmi, T. G . & Herold, P. C. (2019, December). Heuristic Evaluation and User Experience Redesign of 'Think & Link' Learning Environment–A Case Study. In 2019 IEEE Tenth International Conference on Technology for Education (T4E) (pp. 166-169). IEEE.
- Lakshmi, T. G. (2018, August). Developing Students' Conceptual Design Skills for Software Engineering. In Proceedings of the 2018 ACM Conference on International Computing Education Research (pp. 278-279).
- Lakshmi, T. G. & 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

Publications in pipeline

- Lakshmi, T. G. , & Iyer, S. (2021). Fostering conceptual change in software design in IEEE Transactions on Education. Under review
- Lakshmi, T. G. , & Iyer, S. (2021). Applying the Function-Behaviour-Structure (FBS) design lens to explore novices' approach in software conceptual design. In Computer Science Education.

Thank You

Questions please

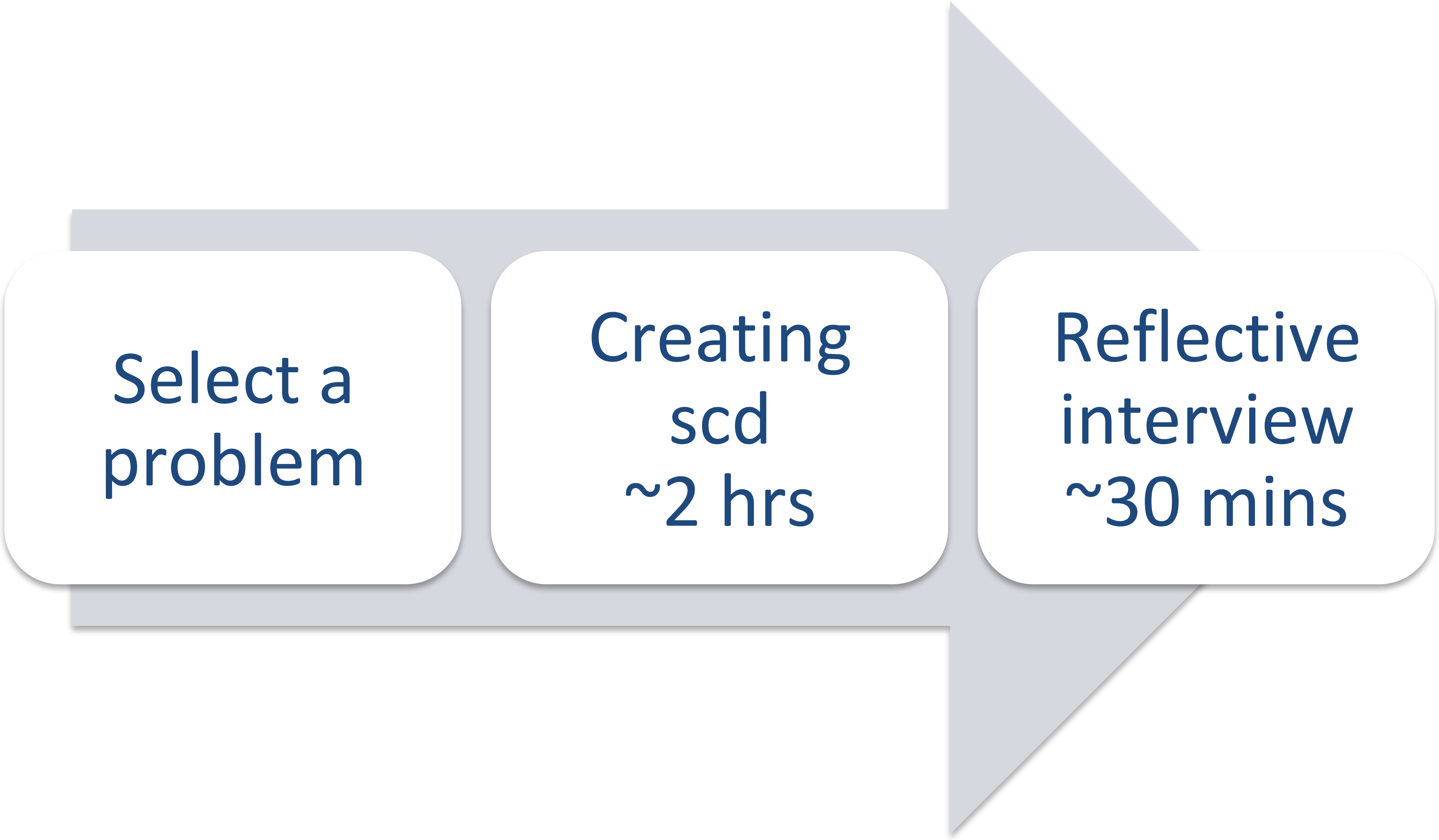
Detail information

Design Problem Characteristics (Brown & Chandrasekaran, 1989)

S.no	Class of Design Problem	Problem Decomposition	Design Plan
1	Class I (Creative)	Not Known	Not Known
2	Class II (Innovative)	Known	Not Known
3	Class III (Routine)	Known	Known

Study 1 - Details

Study 1 - Procedure



- N=5
- Conceptual Design problems -(i) Design a finger print ATM system (ii) Design a mood based automatic player (iii) Design a finger print based payment system (iv) Design a cooking recipe recommender system

Back

Study 1 - Analysis

RQ 1.a What are novices' design strategies while creating SCD?

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

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

Merged timeline, segmentation & generation of linkograph

- Created merged timeline
- Segmented based on FBS codes
- Relationship between segments- linkograph

Linkograph analysis

- Link index, Critical moves
- Analysis of chunks

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

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

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

Merged timeline & interview transcripts

- Code for conceptual design cognition based (Hay et al, 2017) cognitive processes

Back

Artefact Evaluation (Eckerdal et al., 2006)

Category	Content (Indicators)	Representation (indicators)	Group
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	
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	

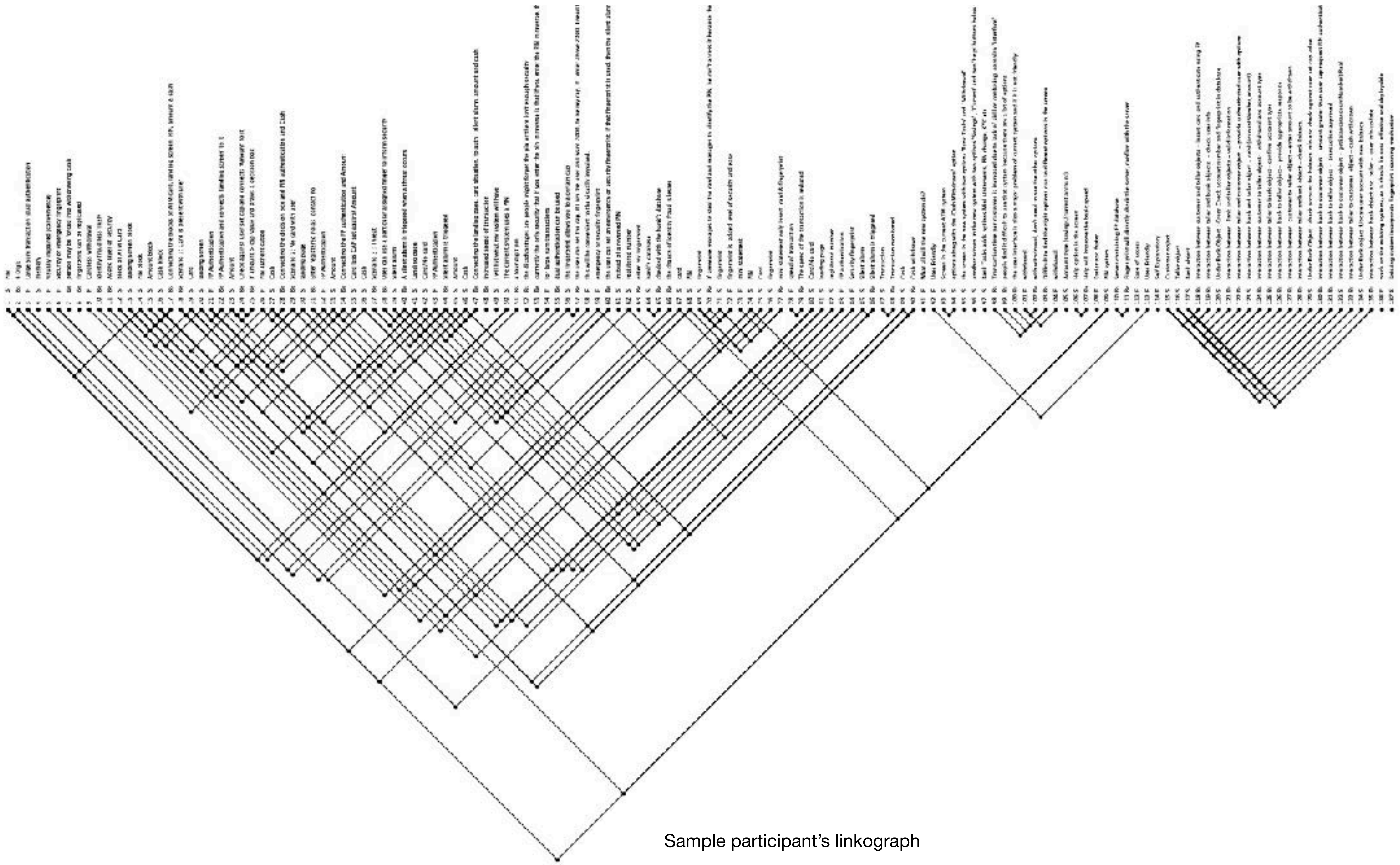
Back

FBS codes for merged timeline

Design Element	Code	Classification Indicator	Example Design Problem (Mood based automatic music player)
Function	F	activity performed by the software system	Mood detection
Expected Behaviour	Be	expected behaviour of the system extracted from the functions	Voice Based Mood Detection (F) - System needs to capture the voice
Structure	S	the solution concepts and components (hardware and software) required to achieve the function	Camera, software to detect mood
Structural Behaviour	Bs	behaviour of the structure, extracted from structures	Camera (S) - Facial features/points are extracted

Back

Design Strategies & Cognitive Processes Analysis - Glimpse



Sample participant's linkograph

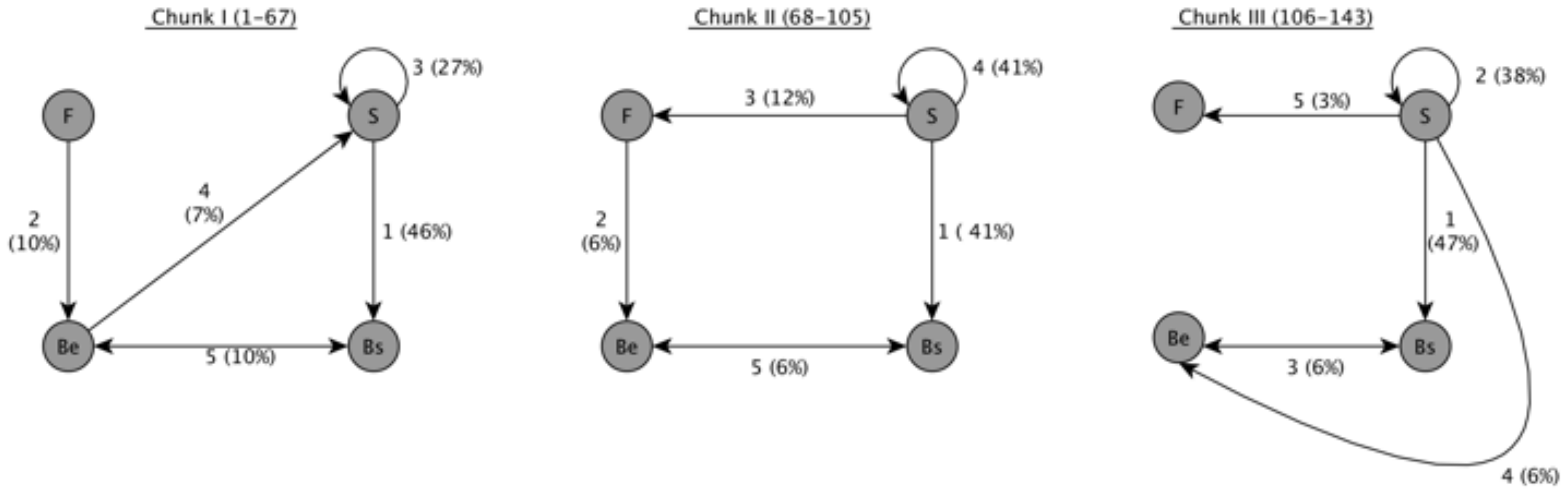
- FBS framework - Protocol based Linkograph analysis (Goldschmidt, 2013)
- Linkograph - areas of interest
- Zoom into the cognitive processes

Back

Study 1 - Detail results
RQ 1.a Sample design strategies

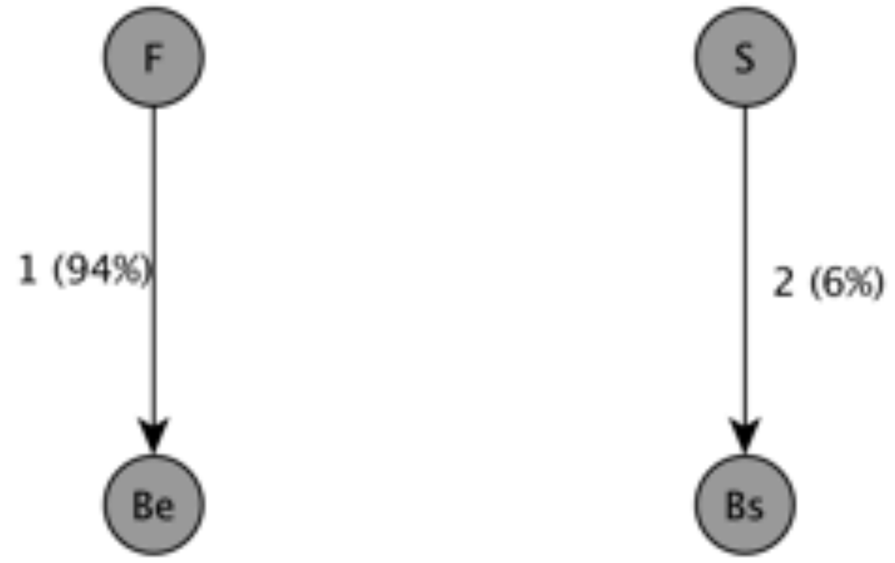
Successful Group

participant4

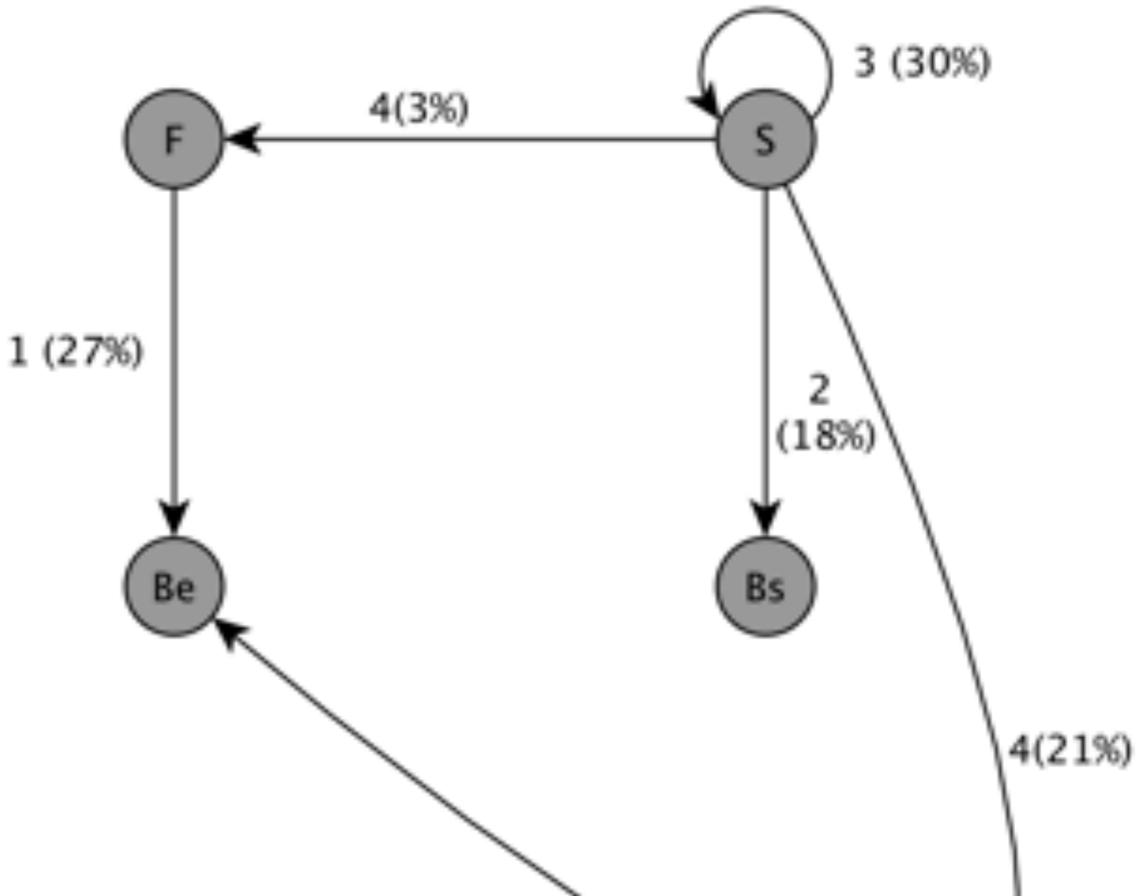


Unsuccessful Group

participant3



participant5



Back

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

Category of Cognitive Process	Role in Design	Cognitive Processes and definition
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

Back

Study 1 - Detail results
RQ 1.b Cognitive Processes

Category of Cognitive Process	Cognitive Process	Design Strategy	Example	
			par2	par4
Long-term memory	Retrieval	generation of structures, functions	exploring current system (Card & PIN based ATM) and its working first, use case of the current system	* Aadhar API *similar systems (FP-ATM , Apple Pay/Google Pay, Startopen)
Creative output production	Analogical reasoning	generation of structures, functions, expected behaviours	exploring PIN and the characteristics of PIN as a 4 digit number	*FP-ATM : payment gateway *Apple Pay/Google Pay: location of finger print authorization/FP storage *Startopen : authentication steps
Executive Function	Problem Structuring (defining goals)	generation of functions	speed of transaction	security
Executive Function	Problem Analysis (constraints identification)	*generation of functions, structures *reformulating expected behaviour and structures	finger print can be replicated, person may be forced or coerced into with drawing cash	failure cases (Aadhar FP collection, FP scanner faulty)
Creative output production	Generating Concepts (via mental	*generation of expected behaviour	card present with the user, no card with the user, threat	connecting mechanism between laptop and FP reader

- Cognitive processes - Conceptual design cognition (Hay et al., 2017)
- Zoom into the cognitive processes
- Deductive thematic analysis (Aronson, 1994)

Back

Study 2 & 3 -Details

Rubric for FBS graph evaluation based on Lindland et al. (1994)

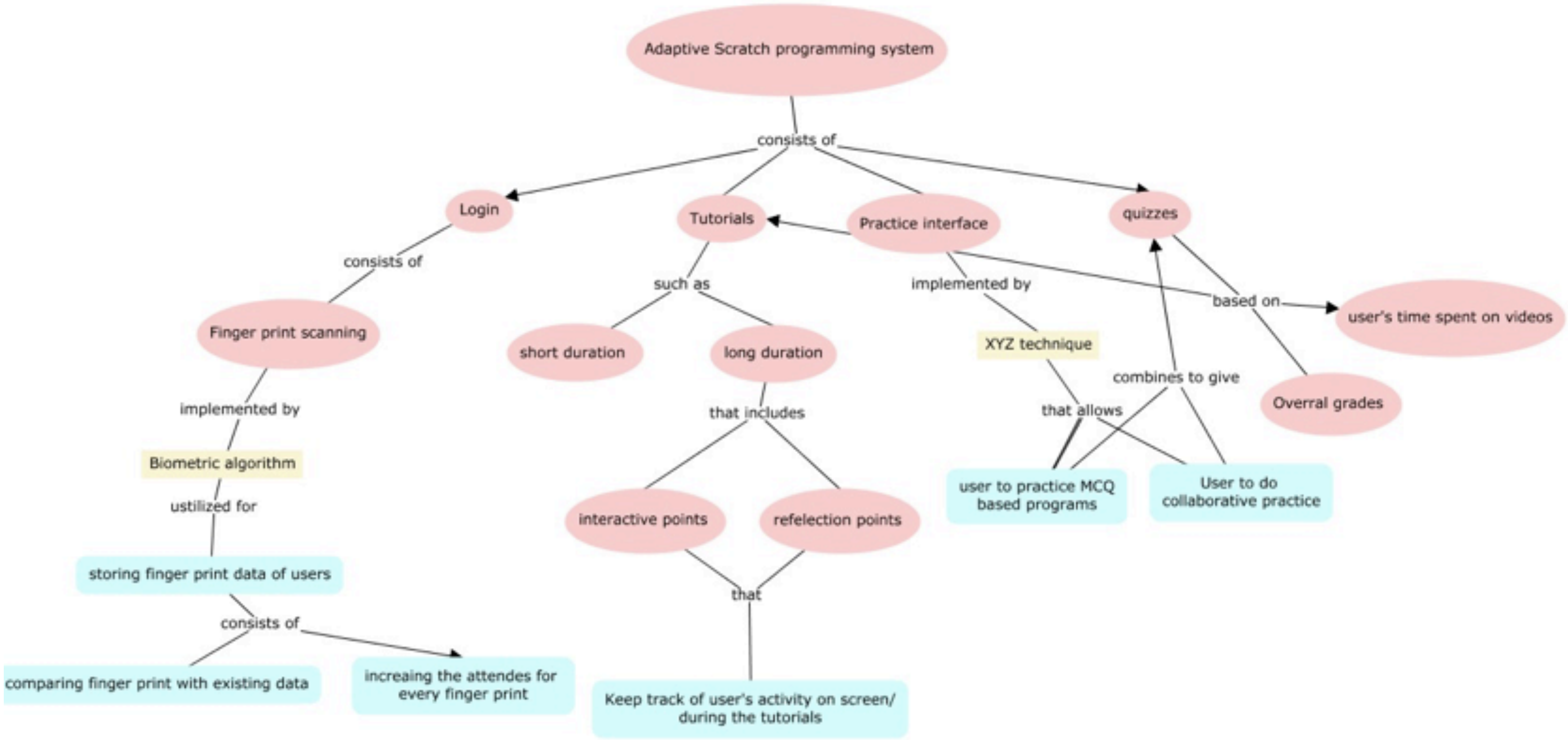
Criteria	Target Performance	Needs Improvement	Inadequate	Missing
Syntax				
Complexity	More than or equal to 12 nodes each for F, B , S.	Only 4 nodes each for F, B, S	Only 2 nodes each for F, B, S	Only a node each for F, B & S
Levels	Two levels in the function, structure and behaviour sub graph are present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

back

Sample response to post-test

Study 2

"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."

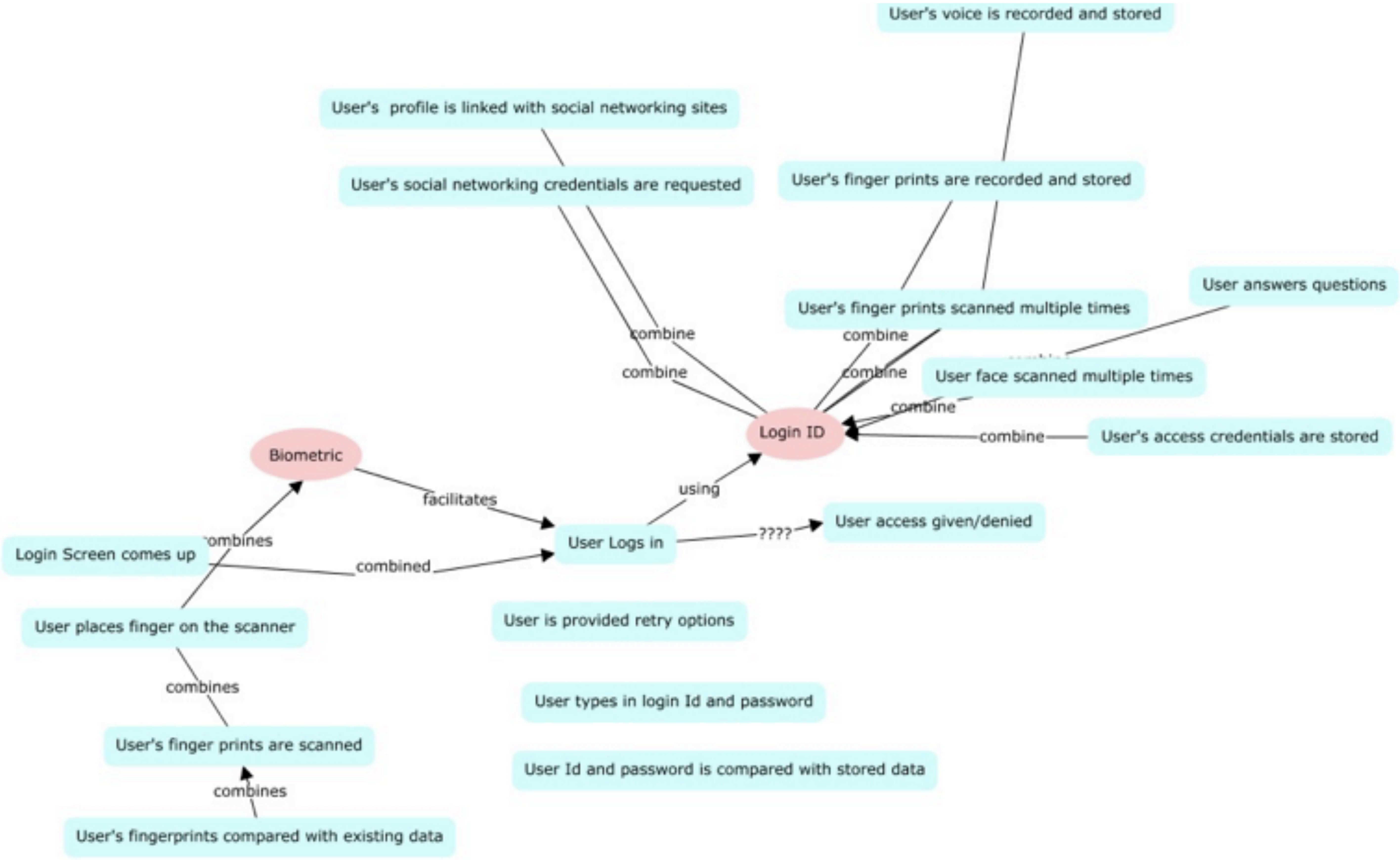


05 July 2021, Ph.D. defence

Back

Study 3

FBS intervention II
Task 2 - FBS graph of a participant

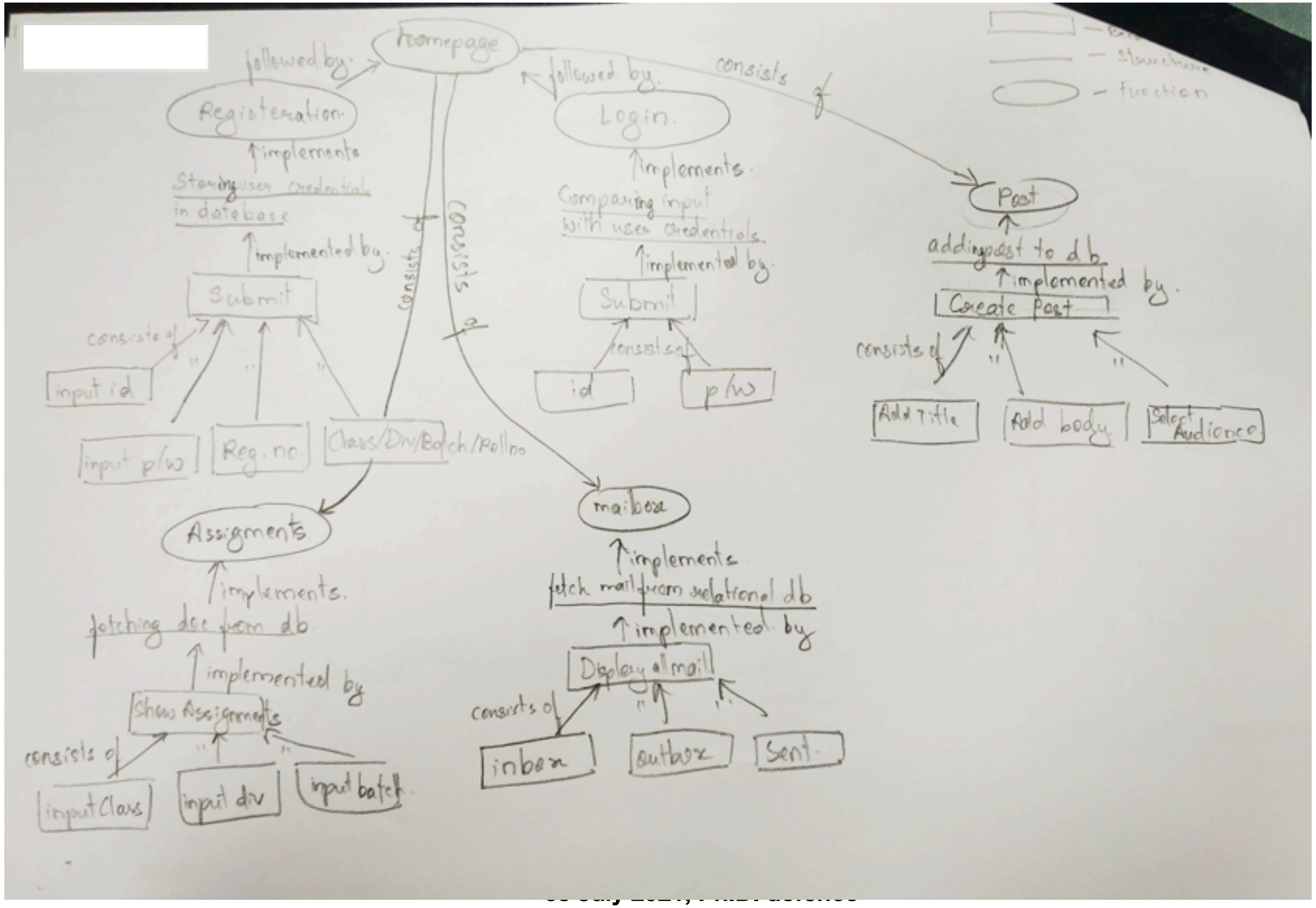


05 July 2021, Ph.D. defence

Back

Sample response to post-test

Study 3



Back

Study 4 & 5 - Details

What is the expected output of SCD? (Eckerdal et al., 2006)

Category #	Category	Content (Indicators)	Representation (indicators)
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

back

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

back

RQ 3.d - R script using TraMineR library

```
#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")
```


back

RQ 3.d - Output of seqefsub()

- 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
- More details - http://traminer.unige.ch/doc/seqefsub.html

back

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
- More details - http://traminer.unige.ch/doc/seqpcplot.html

Introduction - most frequent event sequence path

Study 4

Sequence	Count*	Support**
(introduction,introductioncontext)-(introductioncontextintrovideo)-(introductioncontextintroformsub)	20	1
(introductiongraphtask,introductionworksheettask)	20	1
(introduction)-(introductioncontextintroformsub)-(introductiongraphtask,introductionworksheettask)	20	1
(introductionworksheettask)-(introductiongraphtask)-(introductionworksheettask)	20	1

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

<u>All participants utilise the conjectured features for abstracting the FBS conceptual model</u>

Back

Comparison of semantic interpretation of FBS design elements

Study 4

Post-test category	Abstraction of relationship of FBS (representative)
Restatement (n=2)	● *Function Implements Structure, structure is utilized to achive 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.*

Back

Induction - most frequent event sequences

Study 4

Sequence	Count*	Support**
(induction,inductioncontext)-(inductioncontextintrofeedbacksub)	20	1
(induction)-(inductioncontextintrofeedbacksub)-(inductiongraphtask)-(inductionevaltask)	20	1
(inductiongraphtask)-(inductionevaltask)-(inductioneval)-(inductionphasefin)	20	1
(inductiongraphtask)-(inductionevaltask)-(inductionumlintro)	20	1

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

<u>All participants utilise the conjectured features for evaluation of FBS graph</u>

Back

Comparison of event subsequences

Study 4

Post-test category	Event subsequences
Restatement	(induction)-(inductioncontexttask)-(inductionevaltask)-(inductionphasefin)
Partial design	(inductiongraphintro)-(inductiongraphtask)-(inductionumlintro)-(inductionphasefin) (inductiongraphtask)-(inductioneval)-(inductionevaltask)-(inductionphasefin)
Complete design	(inductiongraphtask)-(inductionevaltask)-(inductiongraphtask) (inductiongraphtask)-(introductiongraphtask)-(inductiongraphtask) (introductionworksheettask)-(inductioncontext)-(inductioneval)-(inductiongraphtask)

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

Back

Ideation - most frequent event sequences

Study 4

Sequence	Count*	Support**
(ideation)-(ideationcontextintrofeedbacksub)-(ideationgraphtask)-(ideationevaltask)	20	1
(ideationgraphtask)-(ideationevaltask)-(ideationumlintro)	20	1
(ideationcontextproblemread)-(ideationcontextproblemsaved)-(ideationgraphtask)-(ideationevaltask)	20	1
(ideationcontextproblemsaved)-(ideationgraphtask)-(ideationevaltask)	20	1

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

<u>All participants utilise the conjectured features for editing problem, graph and completing evaluation of FBS graph</u>

Back

Comparison of event subsequences

Study 4

Post-test category	Event subsequences
Restatement (n=2)	(ideation)-(ideationgraphtask)-(ideationevaltask)-(ideationphasefin)
Partial design (n=14)	(inductionevaltask)-(ideationevaltask)
Complete design (n=4)	(inductiongraphtask)-(ideationgraphtask)-(ideationevaltask) (ideation,ideationcontext)-(ideationcontextproblemsaved)-(inductiongraphtask)

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

Back

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