TELKOMNIKA, Vol. 13, No. 4, December 2015, pp. 1422~1436
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v13i4.xxxx

Computing Game and Learning State in Serious Game for Learning

Ririn Dwi Agustin, Ayu Purwarianti, Kridanto Surendro, Iping S Suwardi
STEI ITB, Jalan Ganesa No. 10 Bandung, West Java, Indonesia
*Corresponding author, email: ririn_dwia@unpas.ac.id, endro@informatika.org, ayu@informatika.org, iping@informatika.org

Abstract

In order to support the adaptive SGfL, teaching materials must be representedin game component that becomes the target of adaptivity. If adaptive architecture of the game only use 'game state' (GS) to recognize player's state, SGfL require another indicator –'learning state' (LS)– to identify the learning progress. It is a necessary to formulate computational framework for both states in SGfL. The computational framework was divided into two moduls, macro-strategy and micro-strategy. Macro-strategy control the learning path based on learning map in AND-OR Graph data stucture. This paper focus on the Macro-strategy modul, that using online, direct, and centralized adaptivity method. The adaptivity in game has five components as its target. Based on those targets, eight development models of SGfL concept was enumerated. With similarity and difference analysis toward possibility of united LS and GS in computational framework to implement the nine SGfL concept into design and application, there are three groups of the development models i.e. (1) better united GS and LS, (2) must manage LS and GS as different entity, and (3) can choose whether to be united or not. In the model which is united LS with GS, computing model at the macro-strategy modul use and-or graph and forward chaining. However, in the opposite case, macro-strategy requires two intelligent computing solutions, those are and-or graph with forward chaining to manage LS collaborated with Finite State Automata to manage GS. The proposed computational framework of SGfL was resulted from the similarity and difference analysis toward all possible representations of teaching materials into the adaptive components of the game. It was not dependent of type of learning domain and also of the game genre.

***Keywords**: Serious Gamefor Learming, Learning State, Game State, And-Or Graph, FSA*

1. Introduction

Entertainment game requires adaptivity for the game more fun and unpredictable [1]. While in the serious game for learning (SGfL), adaptivity is a necessity because of SGfL should be able to adjust to the progress of player skill proficiency and achievement of learning targets. This paper outlines a study of adaptivity in SGfL by utilizing theories, constructs, methods, techniques, tools, or other artifacts of adaptivity in the game, instructional design framework, adaptivity in a serious game itself. The study results manifested in the form of a flexible computing model to the variety of adaptive game component which represented the teaching material, so versatile also for the learning domain and game genres. On this study, ITS (intelligent tutoring system) will be used to evaluate the completeness of computational models SGfL features an intelligent learning system. This paper is a subset of the research on the development of concept models and design models of SGfL with this approach: the transformation of non-game instructional design into the game.

There are three research are closely associated with this paper. The main one is the result of a survey by Lopez about the progress and the movement of research at adaptivity in the area of game [1]. Second is the basic theory from Reygeluth about instructional design framework, which has a micro-strategy and macro-strategy terminology in organizational strategy, which certainly have an impact on the delivery strategy and management strategy[3]. Third, the paper MDKickmeier-Rust, which proposes about adaptivity in a serious game, namely a non-invasive method of micro-adaptivity, within the meaning adaptivity towards learning does not interfere with the flow of game. The Rust's method applied in the case study on narative game based learning. The paper also touched on the need for macro-adaptivity that one of its

Received June 26, 2015; Revised August 13, 2015; Accepted September 2, 2015

functions is to manage the sequence of the curriculum. Macro-adaptivity is expected to be non-invasive as well [3]. Paper from MD KNickmeier-Rust will be used as the primary means of comparison with SGfL models proposed in this paper.

Lopez classify diversity of research results about adaptivity in the game based on the purpose, method and the targeted game component was adapted. This article discuss about SGfL flexible computing models for the 5 different game components as adaptivity target. Studied adaptation method in this paper was limited to the online method with direct adaptivity and centralized mechanism, and variable input only from player skill proficiency aspect. Online adaptivity means that adaptation carried out during running game, controlled by the data about learning progress of player. Direct adaptivity means that the rules for decision-making and choice of actions that can be selected in the decision, has been prepared by the game designer before running the game. Centralized mechanism means that all decision and action for adaptation are done and controlled by one module, not distributed to some independent agents.

The scope of adaptation the learning task in macro-strategy is control the learning path based on map of competencies. Among competency have a prerequisite relationship. Macro-strategy ensure that a competency can only be studied if its prerequisite has been mastered. Variety adaptivity that must be provided in macro-strategy of SGfL are (1) to intervene when the player in the stuck, where no game state can be explored (3) lower/raise the minimum the threshold criteria for mastery of competencies based on the trends of the player's ability to learn (4) encourage the player to repeat the game for achieve a higher level of mastery of the competencies.

If the adaptivity model that be proposed by MDKickmeier-Rust is non-invasive [3] [9], this research contrary want to examine how to integrate learning with components, flow and logic of the game. The reason is "because SGfL must be adaptive based on the learning progress of the player, and the primary object to be adapted are teaching materials and delivery technique, so the teaching material should be represented in a game component that becomes the target of adaptivity". Research question of the studies reviewed in this paper is how the invasive patterns of the representation of the learning material into 5 different components of the game, and then to found flexible computing model for adaptivity in serious game for learning (SGfL).

Reuse component for varian of implementation context is key component of flexible computing model. Domain analysis is a method to find the reuse component, using similarity and difference analysis in the domain problem. Research in this paper used FODA (Feature Oriented Domain Analysis) consist of context analysis: In order to establish scope, domain modeling: in order to define the problem space, architectural modeling, in order to characterize the solution space [4] [5]. Detail of these stage was described in research method. The solution space will be manifested in functional model and architecural model.

2. Research Method

Picture 1 described detail step of research in this paper. Contex analysis give source material to further research. Based on literatur review, in this chapter will be explained about the material. The space of problem in the form of an enumeration of the variants development model of SGfL concept. The variants was developed based on representations of teaching materials on a variety of game component. The next step is analysis of similarity and difference of the functional features that required for computing the game engine. That is not on aspects of multimedia interaction. Including experiments to gain a firmer clarity about how to manage learning state and Game State the game will be described also in the research method. Characteristics of the solution is found in the form of features and computational models be written in result and analysis.

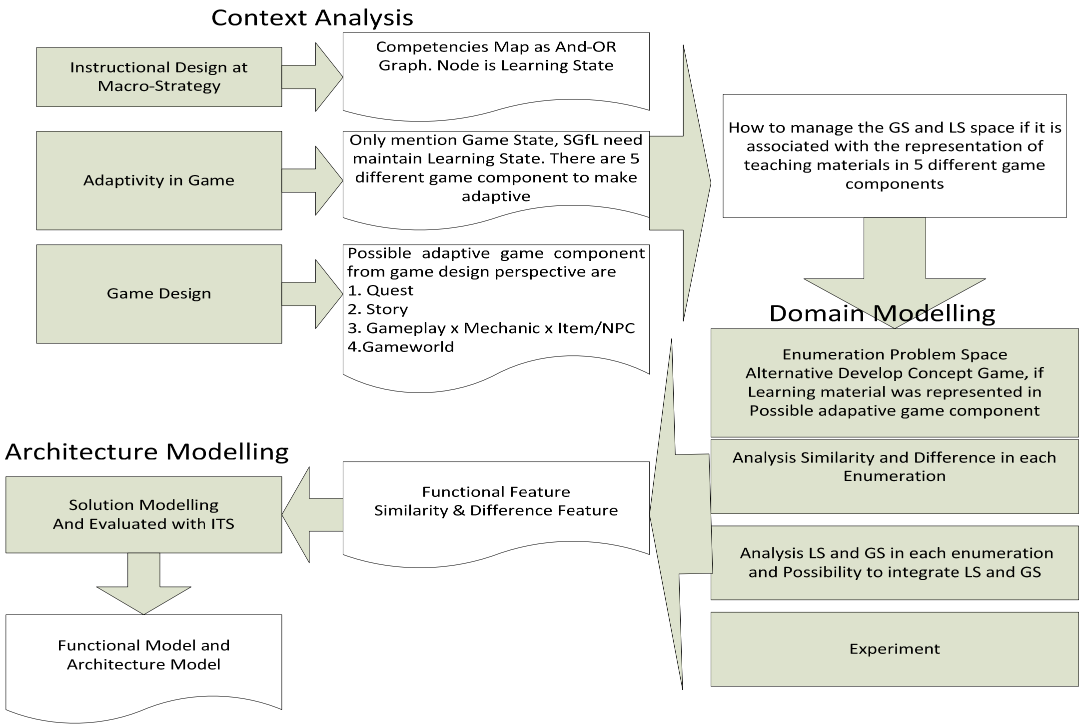


Figure 1. Research Methodology

2.1. Context Analysis

2.1.1 Adaptivity in Game and the Adoption into SGfL

Adaptivity architecture in game could be seen in Figure 2. Adaptation mechanism is listed below.

1. Monitoring player action
2. Interpret player action into variables in player modelling
3. Assign values into player mode
4. Predict Next State Experience using game state and information from player model
5. Construct game elements based on the Next State Experience

Generally, adaptivity in serious game could be done by online or offline. Offline mechanism was done by survey approach to user when user login and before the game loaded, so the engine was called as “content generation”. Meanwhile online mechanism is done along the game based on data obtained from model player (driven approach data) so the engine fit with the name “content adaptation”.

In online mechanism, rules and technique for decision making and types of decisions could be taken may use two approaches, which are direct adaptation (all things are prepared by designer) or indirect adaptation (using machine’s learning to find customized combination of action). To use indirect adaptation, need long enough learning process towards the system and a lot of data for automatic learning.

Refer to the figure 2, there are 5 different components of the game that could be the target of adaptivity. If adaptivity is limited only by the player competence proficiency, not motivation or other mental conditions, then the 5 kinds of components it is an opportunity to represent the teaching materials. That rule was induced from fact that teaching materials and delivery mechanisms that will be the target of the adaptivity in SGfL. Map of learning embodied in the organization of competence in AND-OR GRAPH. Control over the students' learning pathways are the same as controlling the position of students in the map of learning

Set of different status in the game where the player must have been in one of the declared status is the game states space. When this architecture is adopted into SGfL, position of the players can be viewed from two angles, the play map (game state), and the angle of learning (learning state). It can be presumed that in SGfL should be managed game state (GS) space and learning state (LS) space. How space management for the GS and LS, if it is associated with the representation of teaching materials in five option components of the game that be target of the adaptivity is a research question that becomes the main subject of this paper.

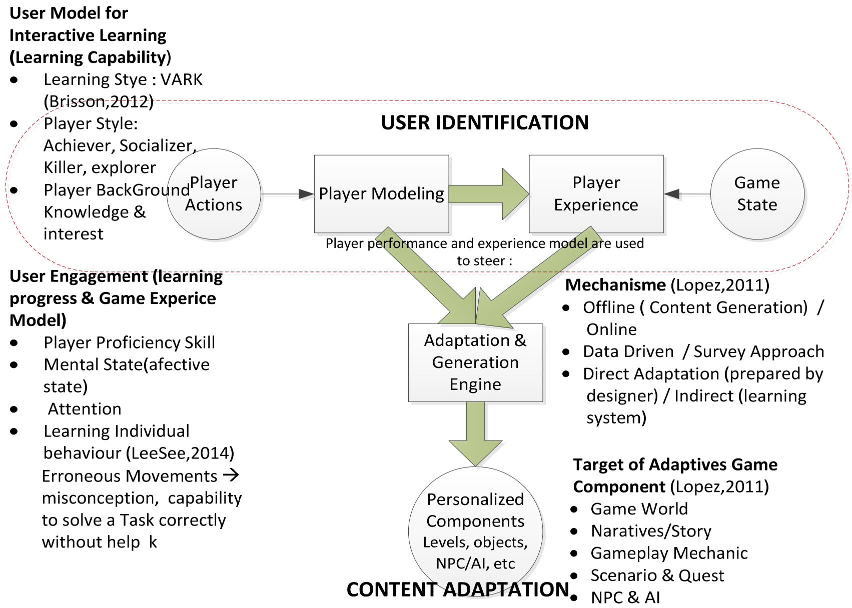


Figure 2. Architecture of Adaptivity in Game – summarized from [1]-[6-10]

2.1.2 Scope of Macrostrategy in Instructional Design

Reigeluth defines framework of instructional design consisted of three elements, which are Condition, Method, and Outcome [2]. The purpose of design is to set the right method that suite the condition existed and the outcome expected. There are four aspect of Condition, i.e.: learning content, learner context, learning context, and specific requirement. Method consists of organizational strategy, delivery strategy, and management strategy. Organizational strategy divided into two levels, which are macro strategy and micro strategy. Macro strategy manages "what do I want the Student to learn" and "what did I know about the Student". Practically, it is organizing learning contents, what the students have to learn to achieve the learning outcome, how to sort, conclude, or synthesize them. Delivery strategy in macro level introduces learning activity and controls Micro strategy module. Management Strategy make decisions towards which contents of learning delivered in what context at every T (time) in learning process. Management strategy needs knowledge about 1) condition and progress of student's learning, 2) map of material's organization, context, and interactions, and 3) strategy to match student's condition with material.

Subjects learned are managed as a set of learning state (LS). LS is a cross of competency levels (Bloom/Anderson taxonomy) in one domain and knowledge object (see table 1). LS, one with another, have a relation between, which is usually made in prerequisite form. One LS could contain one or more knowledge object. To measure student's ability towards an LS, will be needed a standard definition of player's mastery over an LS. Learning method used for an LS also need to be defined. Details of subjects in an LS will be managed in Micro strategy module.

Referred to [11], in the microstrategy was proposed two kind of activity. Those activities are learning activity and assessment activity. In the learning activity, SGfL will provide varian of support for learning based on player capability to learn the new competencies. Judgment about player's skill proficiency was taken from assessment activity only. So the player is not judged by the length of the learning process.

Table 1. Learning State was build from cognitive level x Knowledge Object

Cognitif Level	Knowledge Object P	Q	R	S
Remembering	P1	Q1	R1	S1
Understanding	P2	Q2	R2	S2
Apply	P3	Q3	R3	S3

Example of learning map in Figure 3, show that learning target consisted of three competences. Competence K1 is a set of competences in node P1, node P2, node S1, node S2, node P3, and node S3. Competence K2 is a set of competences in node P1, node Q1, node S1, node Q2, node T1, node T2, and node T3. Meanwhile competence K3 is a set of competences in node P1, node Q1, node Q2, node R1, node R2, node Q3, and node R3. LS Q1 needs mastery of LS P1 to be played.

Management strategy take responsibility to drive the learning path based on the learning map. Example in figure 3, the path only be started from P1, because there is no prerequisite of P1. Learning state P2, Q1, and S1 can be learned after player have mastered P1. Learning state S2 was opened for player if and only if they have mastered P2 and S1, that is the meaning of the two arrows into the S2 is united with curved lines. Criteria of the mastering of competencies is determined by the assessment result of the player skill proficiency in a learning state.

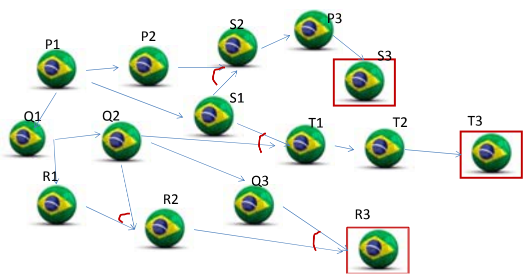


Figure 3. Relationship Prerequisite between LS

Learning Policy Adaptation in Macro Level Strategy

Along learning process, learners' ability is different one to another. For provide adaptivity, the threshold value could be differ for each LS, depends on its difficulty level. If it sets an absolute score for every learner in every LS, then some learner did not experience learning process in a few LS, because they can't fulfill threshold value of prerequisite LS. To solve this, learning designer can apply three types of policy or a combination of them. Below are the three policies.give the learners chance to try again in failed LS, with maximum limit of chances after he get game over state.

(1) lower treshold for learners who have signs of having less ability, so those learners could experience the next LS but with degraded quality of challenges. SGfL can provide 3 kinds grade, low, medium, and high for the treshold.

(2) to learners who have succes through the whole LS required, but not with their optimal result (high treshold), will be given chance to repeat again

By repeating, learners are expected to master the LS better. Order of LS opened can be changed to prevent learners from getting bored. It is better for nonlinierity aspect if SGfL have many alternative material resource.

2.1.3. Rule Analysis For Combine 5 Game Compoment as Target of Adaptivity

Based on Figure 3, there are five types of components that have possibilities to be adaptive. The order and combinations cannot be done freely, however. It has to refer to components' relations in game design aspects.

Combination proposed in Figure 4 is based on details below.

a) Definition of game components according to Richard Rouse [12] and Dave Morris [13]

b) Three types of order in arranging game concept according to **[**12**]**, which are,
   i) *Gameplay → technology → Story;*
   ii) *Technology → story → gameplay;* and
   iii) *Story → gameplay → technology*
c) Framework of gameplay developments and game mechanic from Carlo Fabricatore [14]
   A few proposition obtained are listed below.
   a) Quest /Challenge/Puzzle is core of interaction between game with player
   b) Gameplay, Game Mechanic, and NPC can not define separately → (Gameplay X Game Mechanic X PC/NPC)
   c) Quest is weak entity toward story or toward (Gameplay X Game Mechanic X NPC)
   d) Story can be followed by Gameplay or Gameplay be followed by Story
   e) Gameworld must be relevant with plot. Plot is even in that all of game component collaborate to make story. → changes in the gameworld make impact on the change in the majority of components.
   f) At player story or emergent story, story was created by Gameplay X Game Mechanic X PC/NPC. → (Gameplay X Game Mechanic X PC/NPC) X Story

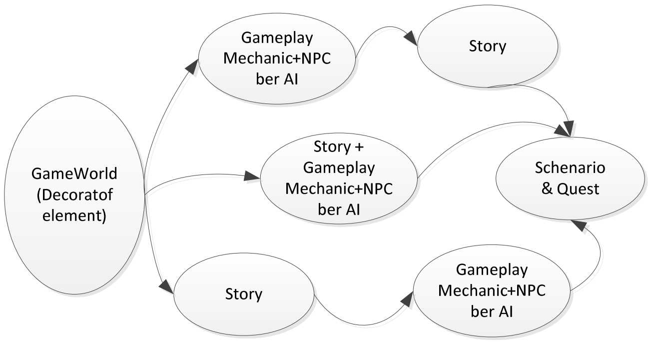


Figure 4. Diagram of Combination Rule for Adaptif Component in Game

2.2. Domain Modelling

2.2.1. Enumeration Space of Problem with Rule

In this section will be displayed eight kinds of development models of the concept of serious games (see figure 5), based on rule at figure 4 and using the terminology of Carlo Fabricatore and game development framework gameplay mechanic that is concerned with learnability. For gameplay, starting from the core gameplay is facilitated by a core mechanic or more Game play can be enriched with core metagameplay without changing the core mechanic. Core mechanic can be enriched with satelite mechanic, in the form of enhancement or power-up, or alternate mechanic. Peripheral gameplay may be used if the story forced to introduce a new gameplay to the player.

Quest is the essence of the interaction of the game to create a challenge. In the quest to be represented SGfL teaching materials. Forms will vary based component quest game where teaching materials are represented. The simplest is a puzzle. The most complicated is if the representation of the material in the form gameplay x mechanic x item/NPC.

1. Quest as Adaptive Component, Story → GamePlay → Quest

Story (embbeded)	Inspired by Story- Support the Quest: gameplay	mechanic (follow gameplay)	item/NPC (follow gameplay)	Learning State	Knowledge Object was presented in quest (inspired by story)	Gameworld (context of Story gameplay, mechanic,item/NPC, quest)
Story -1	gameplay (core+meta)	Core Mechanic [1,2]	(quest as item/NPC)	P1	q1	gameworld A
Story -1+	gameplay (core+meta)	Core Mechanic [1,2]	(quest as item/NPC)	P2	q2	gameworld B
Story -1++	gameplay (core+meta)	Core Mechanic [1,2,3,4]	(quest as item/NPC)	P3	q3	gameworld C

2. Quest as Adaptive Component, GamePlay → Story → Quest

Can be Fixed Component: gameplay	mechanic (follow gameplay)	item/NPC (follow gameplay)	Learning State	Knowledge Object was presented in quest	Story (context of quest)	Gameworld (present the story)
gameplay (core+meta)	Core Mechanic [1,2]	(quest as item/NPC)	P1	q1	Story-1	gameworld A
gameplay (core+meta)	Core Mechanic [1,2]	(quest as item/NPC)	P2	q2	Story-2	gameworld B
gameplay (core+meta)	Core Mechanic [1,2,3,4]	(quest as item/NPC)	P3	q3	Story-3	gameworld C

3. Story as adaptive component, Story →Gameplay → Quest

Learning State	Knowledge Object was presented in Story	Follow the Story: Gameplay	Game Mechanic	Item/NPC	Quest	Gameworld follow story , gameplay, mechanic, item
P1	Story-1	Coregameplay A +meta	Core Mechanic [1,2]	enemy E1	q1	Gameworld A
P2	Story-2	Coregameplay A +meta	Core Mechanic [1,2]	enemy E2, Allied A1	q2	Gameworld B

4. Gameplay x Mechanic x Item/Npc as adaptive component , Gameplay →Story → Quest

Learning State	Knowledge Object was presented in: Gameplay	Game Mechanic	Item/NPC	Story, Inspired by Gameplay,Mechanic, Item/NPC	Quest follow the Story, gameplay, mechanic,item/NPC	Gameworld follow story and others element
P1	Coregameplay [1]	Core Mechanic [1,2]	enemy E1, Allied A1, A2, A3	Story-1	q1	Gameworld A
P2	Coregameplay [1,2]	Core Mechanic [1,2]	enemy E2, Allied A1	Story-2	q2	Gameworld A
P3	Coregameplay [1,2]	Core Mechanic [1,2]+PowerUp	enemy E1, Allied A1, A2, A3	Story-3	q3	Gameworld B

5. Quest as adaptive component, Gameplay → Quest (Player Story)

Can be Fixed Component: gameplay	mechanic (follow gameplay)	item/NPC (follow gameplay)	Learning State	Knowledge Object was presented in quest	Gameworld (context of quest, gameplay, mechanic,item/NPC)
gameplay (core+meta)	Core Mechanic [1,2]	(quest as subset of item/NP(	P1	q1	gameworld A
gameplay (core+meta)	Core Mechanic [1,2]	(quest as subset of item/NP(	P2	q2	gameworld B
gameplay (core+meta)	Core Mechanic [1,2,3,4]	(quest as subset of item/NP(	P3	q3	gameworld C

6. Gameplay x Mechanic x Item/Npc as adaptive component, gameplay →Quest (Player Story)

Learning State	Knowledge Object was presented in (create story): Gameplay	Game Mechanic	Item/NPC	Quest follow gameplay, mechanic,item/N	Gameworld follow the gameplay,
P1	Coregameplay [1,2]	Core Mechanic [1,2]	enemy E1, Allied A1, A2, A	q1	Gameworld A
P2	Coregameplay [1,2]	Core Mechanic [1,2]	enemy E2, Allied A1	q2	Gameworld A
P3	Core meta gameplay [1,2]+	Core Mechanic [1,2,3,4]	enemy E1, Allied A1, A2, A	q3	Gameworld A

7. Gameworld as adaptive component Gameworld as LS, Gameworld → Gameplay → Story → Quest

Learning State	Knowledge Object / Competence	Inspired by Game World: Gameplay	GameMechanic	Item/NPC	Story	Quest Follow Story & gameplay
P1	Gameworld X	CoreGamePlay	Core Mechanic	Enemy X, Allied Y, Item Z	Story-X	q1
P2	Gameworld A	Peripheral GamePlay A	Core Mechanic A	Enemy A, Allied A, Item A	Story-A	q2

8. Gameworld as adaptive component, Gameworld + Gameplay as LS ,
Gameworld → Gameplay → Story → Quest

Learning State	Knowledge Object /	Inspired by Game World: Gameplay	GameMechanic	Item/NPC	Story	Quest Follow Story & gameplay
P1	Gameworld X	CoreGamePlay	Core Mechanic [1,2]	Enemy X, Allied Y, Item Z	Story-X	q1
P2	Gameworld X	CoreGamePlay	Core Mechanic [1,2,3,4]	Enemy X, Allied Y, Item Z	Story-X	q2
P3	Gameworld A	CoreGamePlay +meta	Core Mechanic [1,2,3,4]	Enemy X, Allied Y, Item Z	Story-X	q3
P4	Gameworld A	Peripheral GamePlay A	core Mechanic A [1,2]	Enemy A, Allied A, Item A	Story-A	q4

Figure 5. Enumerasi of Problem Space: 10 kinds of Development Models SGfL Concept

2.2.2. Similarity and Difference Analysis

Of each enumeration can be affirmed, they have the same problem, ie managing learning state as an entity that represents a competency. Characteristics of learning state in macrostrategy level of organizational and management aspects of the strategy outlined in the previous section. The impact of differences in the components used to represent the course material in the game, led to a difference in managing game state (GS) space, whether it can be combined with learning the state or not in the scope of Macrostrategy. Possibility analysis of an integrated managementtoward GS and LS for each enumeration can be seen in tabel 2.

Table 2. Learning State and Game State Analysis in Development Model of the Game Concept

Nu	Id	Model	How to find game concept	learning state in macrostrategy	game state in macrosrategy	Possibility gamestate = learning state
1	1.a	quest based	story-gameplay-quest	class of quest	fchoose (mechanic, quest as item)) or fchoose (class of quest)	possible
2	1.b	quest based	gameplay-story-quest	class of quest	fchoose (mechanic, quest as item)) or fchoose (class of quest)	possible
3	2.b	story based	story-gameplay-quest	story id (tree) or node in branch of tree of story	fchoose(mechanics to choose branch of story tree) or fChooce(story id/tree)	possible, for choice of story id
4	3.b	gameplay based	gameplay-story-quest	C(gameplay,mechanic,item/npc) = usually treated as level of game	the levels of game	more suitable to integrate GS = LS, as choice or decided by system
5	4.a	quest based +story	gameplay-quest create story	class of quest	fchoose(mechanics, quest as item)) for create story	impossible , because mechanic must have functional story
6	4.b	gameplay based+story	gameplay-quest create story	C(gameplay,mechanic,item/npc) = usually treated as level of game	fchoose(mechanics, level)) for create story	impossible , because mechanic must have functional story
7	5.a	gameworld based	non hirarchi	gameworld = usually treated as level of game	the levels of game	more suitable to integrate GS = LS, as choice or decided by system
8	5.b	gameworld based	hirarchical learning state	C(gameworld,gameplay,mechanic, item/npc) =usually treated as level	the levels of game	more suitable to integrate GS = LS, as choice or decided by system

2.2.3. Problem Solving Modelling of Control Learning Path on Learning State Space

The structure of the representation of the learning state space in Figure 3, in artificial intelligence terminology known as AND-OR Graph. Its construction consists of a node, directed arc (in / out), and the relation between the arcs-in on a node. Some arc in which adjoined by a curved line on a node declared the relationship AND. Some arc-in on a node that no curved lines express the relation OR [15]. Nodes that do not have the arc-in, referred to as a fact, which does not have the arc-out called goal, and who have both called subgoal. At the practical level computing, AND-OR Graph represented in sentence calculus propositions in the form of a special clause horn. Examples AND-OR graph representation in figure 3 into the calculus of propositions can be seen in the Figure 6

At LS space, a node represents a competence that should be mastered by player. Arc stated prerequisite relationships. Node with a bow in need of competence from its pair node. Control strategy of the learning path starts from the node that does not have the prerequisite (Fact), discover nodes that all prerequisites met (sub Goal), to the node that indicates a complete learning outcomes studied (goal). The target from study is the overall goal node or a part of it.

There are two kinds of agoritma to build inference engine for AND-OR graph, ie forward chaining and backward chaining. The algorithm in accordance with the management strategy of macro-modul is a forward chaining. Figure 6 describing the forward chaining algorithm applied to the game "Save The KOD Kingdom".

2.2.4. The Experiment: Build Prototype SGfL about Learning SQL

Experiments conducted by building two prototype game with different genres, to support learning on the same topic, ie SQL. Adaptivity target component is a quest. Game version 1 implements enumeration 1 (seet figure 5) which design of the game mechanic at macro-strategy make LS united with GS. The second version implement enumeration number 2, which game mechanic have narrative function, so GS did not unite with LS.

Game version 1, where LS = GS, is “SAVE THE KOD KINGDOM”. It was adventure game, embedded story, with only one game mechanic in macro level, that is player choose area where the next quest is waiting to be done. Goal of this SGfL is to collect points from solving problems in a kingdom. Character of the player here is a princess, Ruruna, who gets a sudden duty from his father to rule the kingdom. The story of this game could be seen at figure 8.

```
Infered: contentedall proposition symbol with related data
TabClausa : contentedd norm relational form if IF-THEN clause
TabCount : contented clausa id and number of premises
Agenda : stack for computation in Forward chaining
Result : contented proposition symbol was known as true
TabTracking : contented proposition have been processed and
 become true
Begin
Input (Tabclauses)
Input (tabInfered)
Create(TabCount)
Push( all Fakta, Agenda)
GameOver=False
While agenda.notempty()= false and gameover==false do
    P = Agenda.pop()
If Infered[P].value = false Then { the proposition symbol still false}
{display property atribut of P}
{ask for user, input assesment result of P}
IFassresult>= P.treshold Then
          Push(TabTracking,P);
          Infered[P].value = True ;
          ClausesMatch= Select * From Tabclauses Where Left=P.symbol,
          For i=1 to end ClausesMatch Do
              Decrement (TabCount[clausesMatch.Norule]. Count)
If TabCount[clausesMatch.Norule].Currentcount == 0 then
If TabCount[clausesMatch.Norule].Symbol is goal Then
{ask for user, input assesment result for the Goal symbol}
                    IF assResult>= Goal.treshold THEN
{save the symbol to Result}
                       If Symbol tersebut GoldenGoal then
                          Gameover=true { winner, finish game withexcellcene}
                       endIF
                    ENDIF
Else
                    Push (TabCount[clausesMatch.Norule].Symbol)
                    Urutkan agenda berdasarkan nilai heuristic
Endif
                 TabCount[clausesMatch.Norule].statusExecute=True
Endif
          endFor
{update the count matrix}
Endif
{display " try another option"}
Endif {proposition symbol have been processed-Skip}
 Endwhile
```

```
1.  If P1 then P2
2.  If P2 then S1
3.  If P2 thenS2
4.  If S2thenP3
5.  If P3 thenT2
6.  If Q1thenQ2
7.  If Q1thenR1
8.  If Q2 AND R1 then R2
9.  If Q2 thenQ3
10. If S1 thenQ3
11. If R2 thenR3
12. If Q3thenT3
13. If R3AND Q3 thenT1
```

Figure 6. Proposition Calculus from Figure 3 and Forward Chaining Algorithm

It could be seen in Figure 7, there were two "!"signs. They were provided for the player to choose them. Engine of game that unit learning state and game state controls computation behind these sign choices that appeared on the game interface. Capture of computation process could be seen in Figure 8. There, P1 is already done. P2 and Q1 appeared on the interface as the choices of quest which the player would do next. The other learning states are not yet to be opened.

Game version 2 is ALTERCITY. Its genre is career simulation; with game mechanic are actions that relevant with carrier management and daily life. Quest was put as an item, which is done by player in player's job and training while player was building his carrier. The goal is to collect wealth, to get the highest position in the most prestigious company in a town called Alter city, and to get a prospective couples. Character of player is an informatics technology graduate named Mada who starts his career in Altercity. There are two kinds of learning state, working state and training state. Training state represented "know" and "understand" competence level. Working state represent "apply" competence.

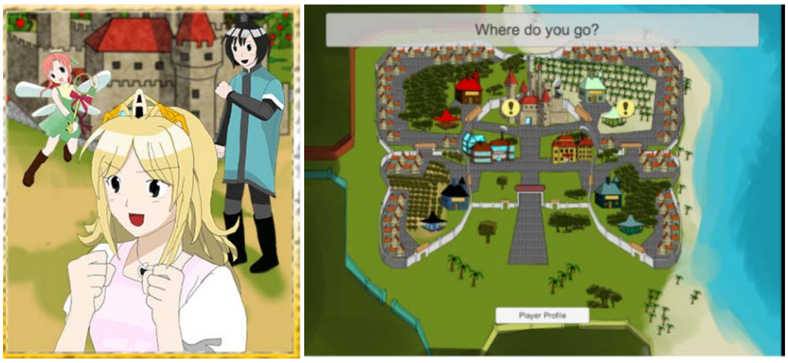
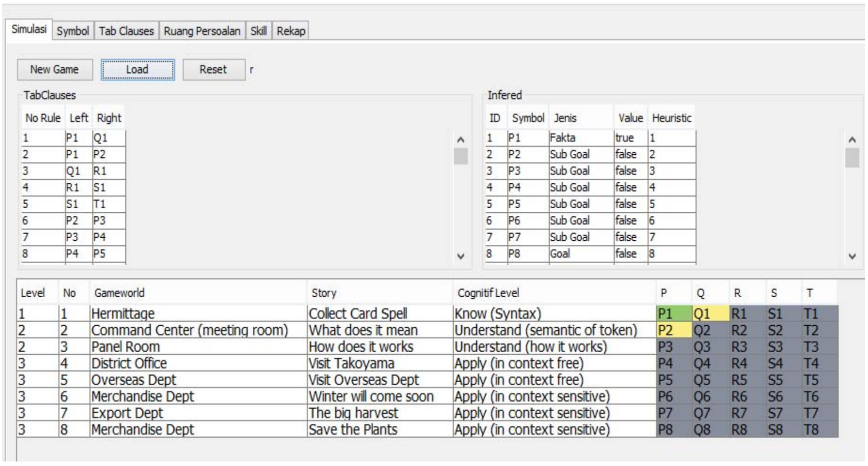


Figure 7. Illustration of Game Interface version 1- player choose sign!



No Rule	Left	Right
1	P1	Q1
2	P1	P2
3	Q1	R1
4	R1	S1
5	S1	T1
6	P2	P3
7	P3	P4
8	P4	P5

ID	Symbol	Jenis	Value	Heuristic
1	P1	Fakta	true	1
2	P2	Sub Goal	false	2
3	P3	Sub Goal	false	3
4	P4	Sub Goal	false	4
5	P5	Sub Goal	false	5
6	P6	Sub Goal	false	6
7	P7	Sub Goal	false	7
8	P8	Goal	false	8

Level	No	Gameworld	Story	Cognitif Level	P	Q	R	S	T
1	1	Hermittage	Collect Card Spell	Know (Syntax)	P1	Q1	R1	S1	T1
2	2	Command Center (meeting room)	What does it mean	Understand (semantic of token)	P2	Q2	R2	S2	T2
2	3	Panel Room	How does it works	Understand (how it works)	P3	Q3	R3	S3	T3
3	4	District Office	Visit Takoyama	Apply (in context free)	P4	Q4	R4	S4	T4
3	5	Overseas Dept	Visit Overseas Dept	Apply (in context free)	P5	Q5	R5	S5	T5
3	6	Merchandise Dept	Winter will come soon	Apply (in context sensitive)	P6	Q6	R6	S6	T6
3	7	Export Dept	The big harvest	Apply (in context sensitive)	P7	Q7	R7	S7	T7
3	8	Merchandise Dept	Save the Plants	Apply (in context sensitive)	P8	Q8	R8	S8	T8

Figure 8. Capture of Engine Implementation for Control Learning State

Figure 9 shows the interface of Altercity. From top-left to bottom-right: MADA boarding to altercity, accompanied by allied Pak Eza, which gives clues about what to do in Altercity. The next is global gameworld of Altercity, first residential for MADA, job announcements, Mada goes to Altermart, and Mada met with HRD of ALTERMART to apply for jobs.The contents of jobs drawn from the results of the execution of the FSA toward the And-Or Graph of LS

Figure 9. Game Interface at ALTERCITY

Learning state, game state and game mechanic for Altercity can be seen in the FSA in Figure 10 and tabel 3. Game mechanic with parameter wState and tState represent learning activity.

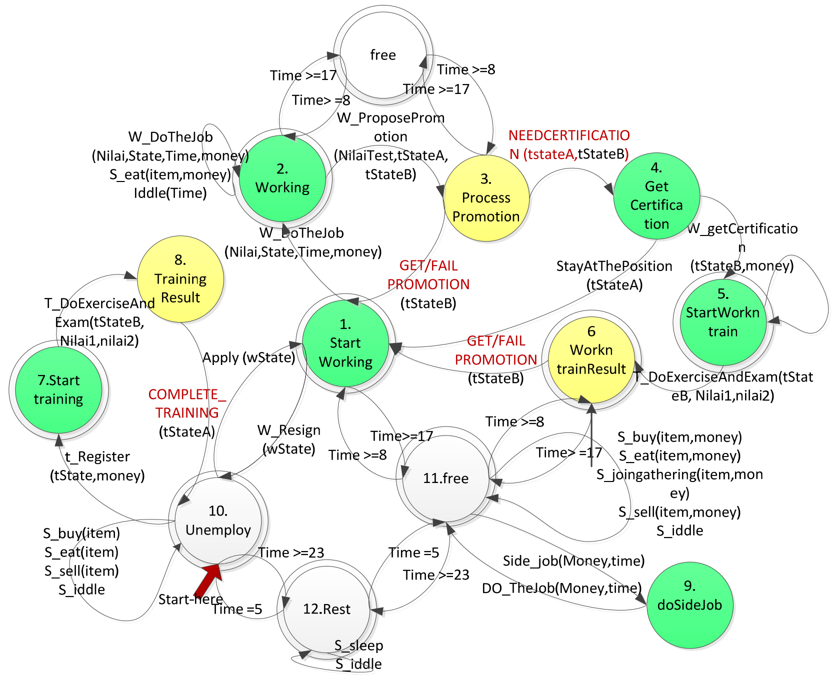


Figure 10. Game State and FSA of Altercity

Table 3. Learning State in Altercity

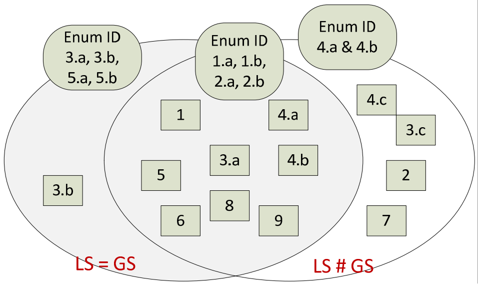
Training Level	Concept Star-1	Concept Star-2	Concept Star-3	Concept Star-4	Cognitif Level	Concept Working	Star-1 Mini market	Star-1 Klinik	Star-1 Sport center	Star-2 Bank	Star-2 District Office	Star-3 Super market	Star-3 Hos pital	Star-4 Bank	Star-4 Town Office	Cognitif Level
Basic	T1-1	T2-1	T3-1	T4-1	Know, understand	Trainee	P11	P21	P31	Q11	Q21	R11	R21	S11	S21	Apply
Junior	T1-2	T2-2	T3-2	T4-2	Know, understand	Staff	P12	P22	P32	Q12	Q22	R12	R22	S12	S22	Apply
Intermed	T1-3	T2-3	T3-3	T4-3	Know, understand	Superviso	P13	P23	P33	Q13	Q23	R13	R23	S13	S23	Apply
Expert	T1-4	T2-4	T3-4	T4-4	Know, understand	Analyst	P14	P24	P34	Q14	Q24	R14	R24	S14	S24	Apply

3. Results and Analysis

3.1. The Commonality of Computing Model

Similarity and difference analysis and the experiment support the identification of functional features required in the computing engine of SGfL, not on multimedia computing to gaming interface. These features are classified based on management pattern GS and LS state, bounded on Macro strategy module only. The result can be seen in the figure 11 and table 4. There are three groups of issues,

(1) The group recommended for unite the GS with LS space, that is, if course material represented in the gameworld or (gameplayxmechanic x item / NPC with designer story).
(2) The group that can choose to use a model of integration or separation between GS and LS
(3) The groups that should management of GS and LS, as different entity, that is, if the representation of course material in (a) quest or in (b) (gameplay x mechanic x item /NPC) as the level, and the player story. Game designers have to design gameplay as story builder



1. Manage Learning State (LS) and AND-OR Graph of LS
2. Manage Game State –GS (gameplay x mechanics x item/Npc) in FSA model
3. a Manage Item /NPC, Decoratif, Dialog component to support story and game balance

3.b Manage relation LS and others game component
3.c Manage relation GS and others game component
4.a Manage player model (student model)
4.b Manage player model (gaming variabel)
4.c Manage player model (GS history)
5. Setting and execute Learning Policy as adaptivity in macrolevel
6. Control Learning Path based (and-or Graph of LS wih Forward Chaining)
7. Control Game State with FSA enginer
8. Execute Game balance
9. Inerfacing with game user interface

Figure 11. Result of Similarity and Difference analysis

Table 4. Result of Similarity and Difference analysis

ID	Type	Object	Attribute
Fea-01.0.1	Data	Learning State (LS)	Mandatory
Fea-01.0.2	Organizing	Prerequisite Relation among LS	Mandatory
Fea-02.0.1	(CRUDE)	Game State and rules of its production	Mandatory
Fea-03.a.1		NPC, Item with behaviour, interactivity, value,dialog,and decorative component to support story, gameplay and gamebalance	Mandatory
Fea-03.b.1		Game State'srelation with other game elements	Substitutive Fea-3.c.1
Fea-03.c.1		Learning State'srelation with other game elements	Substitutive Fea-3.b.1
Fea-04.a.1		Manage Player Model (Student Model)	Mandatory
Fea-04.b.1		Manage Player Model (gaming Variabel)	Mandatory
Fea-04.c.1		Manage Player model (game state history)	Mandatory for GS#LS
Fea-08.0.1		Execution Game Balance for Gaming Variabel	Mandatory
Fea-05.0.1	Intellligence Computing	Execute Adaptivity in Learning Policy based on student model	Mandatory
Fea-06.0.1		Control Learning State	Mandatory
Fea-07.b.1		Control Game State- FSA	Mandatory for GS#LS
Fea-09.0.1	Multimedia Programming	Interfacing with UI of game aplication for sensory immersive implementation	mandatory
Fea-10.0.1	Information	View Player Model	Addtional
Fea-10.0.2	Processing & dashboard	View LS Performance in supporting learning efectivity	Additional

The experiment has been developed instance two game concepts, core game design, game interface prototype development, and implementation of the core function of computational models needed to macrostrategy. Especially for prototype version 1, "Save the KOD Kingdom", also has made a prototype for Microstrategy. Lesson learned is obtained

1. Confirming the hypothesis that the SGfL, need to be managed LS space in addition to the GS space. However, in several of the concept of the game, both can be united, so that computing becomes more simple.
2. Proving the model representations And-Or Graph, calculus propositions in the form of horn clauses appropriate to implement organizational strategy at the macro level. Forward chaining algorithm modification according to implement a management strategy at the macro level. The solution can re-use for both kinds of LS and GS space management
3. Exemplifying the mechanism of how to interact with Gameplay Mechanic LS using Finite State Automata as a computational model

In the experiments conducted there are shortcomings, ie the components supporting the game balance game, story, items, and smart NPC, as well as the gameworld is still modest. For prototype version 1, gameworld represented in the form of comic and implemented as a series of images that display before or after quests of an LS. At Altercity, 3D elements of the room, item, and the allied NPC with their dialogue with the player character has not been put into computing. These need further research and experiment. Using multiple smart NPC Allied / Enemy very interesting for further study, from computing aspect and from learning method aspect.

3.2. Functional Model

The functional model will describe the implementation from functional features into the process and data as well as the interaction between processes. Figure 12 describe the functional model for two kinds of computational models as a solution of the two kinds of cases, ie for the case (a) LS integrated with GS and (b) a separate LS and interact with GS. Model B requires the addition of a process number 6 and number 7 in the model b compared models a, additional data store for the player models that save game state history. There is a change in behavior on the number 1, in the model (a) related to the game componet. In the model (a), the main control in the second game, while on the model (b) is in the process 7

3.4. Architectural Model

Based on architecture of adapativity in game that proposed by Lopez [1], this paper proposed architecure adaptivity in SGfL at figure 13. In the Architecture have been put maping functional feature of ITS (italic font) and aspect in framework instructional desain (bold font). All of the feature and the aspect from ITS and Insructional Design framework was included in the map. Because the ITS feature is used as one of the evaluators about the completeness of features an intelligent system to support learning, then from the SGfL architecture can be concluded that SGfL feature has been intelligent enough to adapt to the needs of the player.

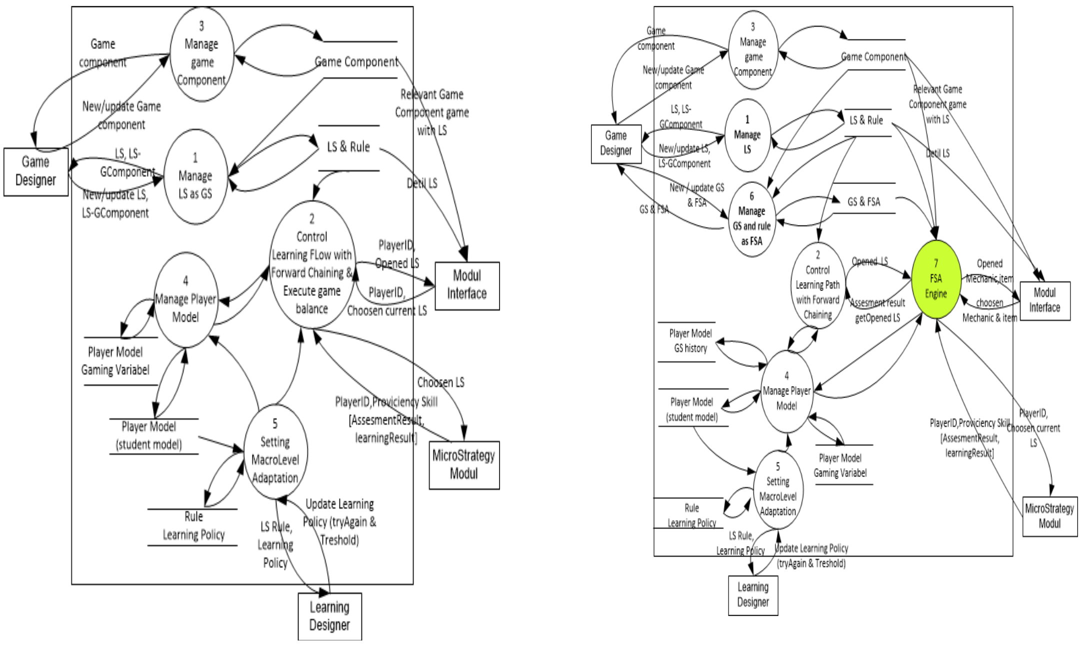


Figure 12. Functional Model of SGfL's computation where LS=GS (left) and LS # GS (right)

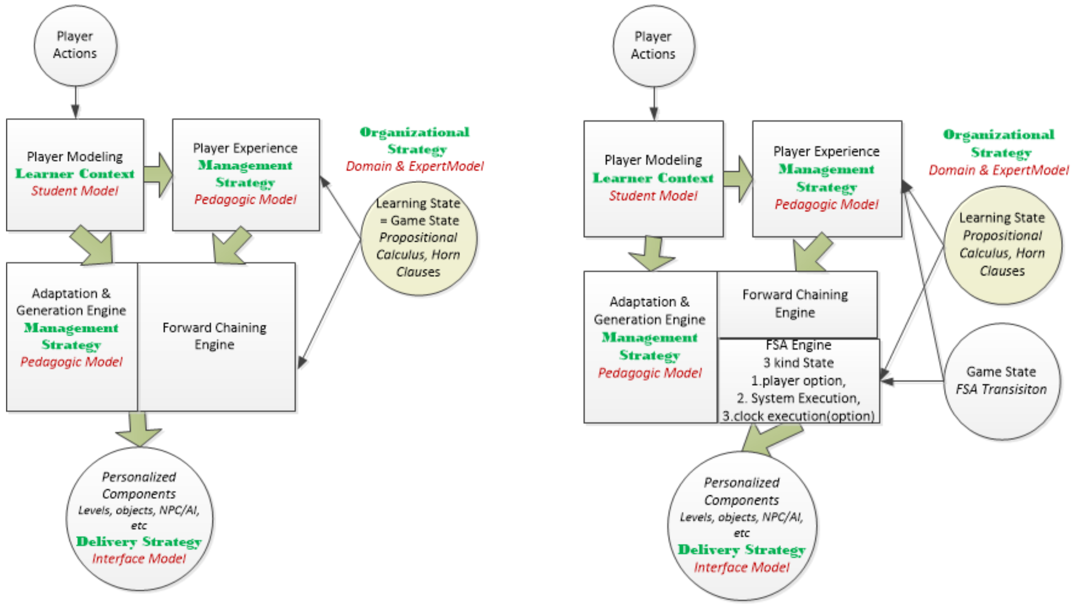


Figure 13. Architecturall Analysis a) Left: GS = LS b) right: GS separated with LS

The proposed architecture model of game as in figure 12, add one entity, that was learning state where the SGfL design, manage the game state as different entity with learning state. Learning state was used to represent learning content at macro level strategy meanwhile game state was a current state of a player as subset of eligible states provided in game.

Generally game state is managed by Finite State automata. And-Or Graph was suitable data structure for represent learning state. The structure can be solved with reasoning approach using rule based system, which knowledge base and inference engine are are mandatory component. Propositional calculus in clause horn is simple and powerful for implemented the and-or graph in knowledge base. Forward chaining was suitable for implement management strategy at macrostrategy.

4. Conclusion

Some of the conclusions that want to be affirmed are as follows:

1. SGfL must manage learning state (LS) space beside game state (GS) space. The issue would be simpler if the GS can be integrated with the LS space, thus managing only one state in the game. But apparently not in all cases both the state space can be integrated.
2. How to manage LS and GS in SGfL on macro-strategy module**,** depend on game component where game designer represent the learning material and learning task beside depend on gameplay design.
3. To support good adaptivity in SGfL, LS must be represented in adaptive game component. Based on 5 different game component as target of adaptivity, have been developed 10 kinds of development model of SGfL Concept.
4. Based on possibility analysis to integrate Gs and LS, from 10 kinds development model of SGfL Concept, there were 3 groups. Those are recommended (4), optional (4), and impossible (2). If learning material was presented in quest and use player story approach then LS must separated from GS. the same is true if the learning material presented in gameplay mechanic x item /NPC and uses the player story concept.
5. Experiment in this research affirmed that
   a. where GSwas integrated with LS, control on game can be handled by rule based system, clausa horn as knowledge representation format and forward chaining
   b. Where GS was separated with LS, control feature in macrostrategy modul need collaboration of Finite State Automata and rule based system with forward chaining.

6. Research on this SGfL need to be continued on the domain analysis of the micro-strategy. Does solution model in macro-strategycan be implemented in micro-strategy? What customization was need for the problem space in micro-strategy.

References

[1] Lopez Ricardo, Bidarra Rafael. Adaptivity Challenges in Games and Simulation : A Survey. *IEEE Transaction on Computational Intelligence and AI in Games*. 2011; 3(2).

[2] Reigeluth C. Instructional Design: Theory and Model Second Edition. Lawrence Erlbaum Associates, 1983

[3] Kickmeier-Rust MD, Albert. Micro-adaptivity: protecting immersion in didactically adaptive digital educational games. Journal of Computer Assisted Learning. 2010; 26: 95–105.

[4] Jatain Aman. Comparison of Domain Analysis Methods and their supporting Quality Attributes. Thesis. Patiala, Thapar University. 2009.

[5] Hirave Kanifnath S, Hanchate Dinesh Bhagwan. Survey on Ex Feature: A Feature Modelling and Recommending Technique For Domain Oriented Product Listing. *International Journal of Engineering Research and General Science*. 2014; 2(6).

[6] Westra Josst et.al. Guiding User Adaptation in Serious Games. Agent for Games and Simulation*II, LNAI 6525*. 2011; pp. 117-131, @ Springer-Verlag Berlin Heidelberg.

[7] Westra Joss et.al. Keep the trainee on track. *IEEE conference Computing Intelligent Game*, *Denmark.* 2010.

[8] Lee Seong Jae, Liu Yun End, Popovic Zoran. *Learning Individual Behaviour in Educational Game: a Data Driven Approach*. The 7th International Conference on Educational Data Mining (EDM). 2014.

[9] Koidl Kevin, Mehm Kevin et.al, *Dynamically adjusting Digital Educational Games Toward Learning Objectives*, The 3rd European Conference on Game Based Learning, 2010; 177-184.

[10] B Magerko, C Heeter, et.al, *Intelligent Adaptation of Digital gamae-based Learning*, ACM Conference Future Play, Canada, 2008.

[11] Minovic Miroslav, et.al. Visualisation of Student Learning Model in Serious Games. Computers in Human behaviour. 2014.

[12] Richard Rouse. Game Design: Theory & Practice Second Edition. Wordware Publishing, Inc, 2005.

[13] Rollings A, Morris D. Game Architecture and Desisgn: A New Edition. Indianapolis, New Riders Publishing. 2004.

[14] Fabricatore Carlo, *Gameplay and Game Mechanics Design: A Key to Quality in Videogames.* OECD-CERI Expert Meeting on Videogames and Education. 2007.

[15] Russel Stuart, Norvig, Peter. Artificial Intelligence a Modern Approach. Prentice Hall, 2003.