International Conference on Cyber Security and Computer Science (ICONCS'18),
Oct 18-20, 2018 Safranbolu, Turkey

A WildCAT Based Observable Bayesian Student Model

Soufiane Boulehouache[1], Selma Ouareth[1], and Ramdane Maamri[2]
[1] 20 Août 1955 University of Skikda, Algeria, sboulehouache@yahoo.fr.
[2] Mentouri University of Constantine, Algeria, rmaamri@yahoo.fr

***Abstract* - The Student Model is dedicated to personalize and to adapt the learning. With pedagogical strategy self-switching, the monitoring of the student model is the cornerstone of pedagogical strategy adapting. To efficiently achieve the monitoring operation, we propose a fine grained WildCAT based Observable Bayesian Student Model. On one side, it represents how the user relates to the concepts of the knowledge structure using the pedagogical component. On the other side, it integrates concept level sensors that results in an Observable Networks' Sensors. This permits to ensure the collect of the instant student knowledge level. In addition, it uses a publish/subscribe communication model to notify the Student Cognitive changes to the monitoring component. On this side, the Monitoring Component subscribe as a receiver of appropriate cognitive changes. To experiment the likelihood and the usefulness of this model, a framework is constructed using WildCAT on a Student Cognitive Level.**

***Keywords* - Bayesian Student Model, Self-*, MAPE-K, Software Monitoring, WildCAT, Pedagogical Strategy Self-Switching.**

I. INTRODUCTION

The Student Model is dedicated to personalize and to adapt the learning. The Student Model stores the dynamic features of the student during learning sessions. In order to construct a student model, it has to be considered what information and data about a student should be gathered, how it will update, and how it will be used in order to provide adaptation [1, 2]. Generally, the overlay model which represents the student's knowledge level [3], is used. The overlay model can represent the user knowledge for each concept independently and this is the reason for its extensive use [3]. Knowledge refers to the prior knowledge of a student on the knowledge domain as well as her/his current knowledge level. This is usually measured through questionnaires and tests that the student has to complete during the learning process [3].most QoS values offered by service providers are not static and can change over time [7], [8], [9].

To deal with the uncertainty of the student evaluation, Bayesian Networks are used. The attractiveness of Bayesian models comes from their high representative power and the fact that they lend themselves to an intuitive graphical representation, as well as the fact that they offer a well defined formalism that lends itself to sound probability computations of unobserved no des from evidence of observed no des [4].

Self-* 1 has emerged to deal with highly dynamic context of use. It seeks improve computing systems with a similar aim of decreasing human involvement [5]. They are closed-loop systems with feedback from the *Self* and the *Context* [6]. They are designed as two separated interacting sub-systems. They use a distinct external Manager SubSystem (MrSS) that implements the adaptation logic to control the Managed Sub-System (MdSS) that implements the functional logic. According to Garlan, Cheng, Huang, Schmerl, and Steenkiste [7], recent works use external models and mechanisms in a closed-loop control fashion to achieve various goals by monitoring and adapting system behavior at run-time. Also, the use of a distinct MrSS component permits to provide a high level of flexibility to evolve. It permits the easy replacement of the MrSS by a more sophisticated one. The MrSS is related to the MdSS using sensors and effectors. We can say that through the (MrSS), the Self-Adaptive System changes its behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible [8].

In fact, with a Self-* Pedagogical Agent, the monitoring of the student model is the cornerstone of pedagogical adapting. To efficiently achieve the monitoring operation, we propose a fine grained WildCAT based Observable Bayesian Student Model. This last represents how the user relates to the concepts of the knowledge structure using the pedagogical component.

We discuss the proposition in the remainder of this paper as following. In section two, we present the architecture of Self-* Pedagogical Systems. Here, we focus on the MrSS's and the MdSS's sub-components internal structure. In section three, we present the WildCAT based Observable Bayesian Student Model. The section four presents a road map of the construction of an Observable Student Model using the proposed model. The next section describes a use case of the Observable Bayesian Student Model in a Multistrategic Pedagogical System. Here we focus mainly on the updating and the monitoring of the student model. The last section presents conclusions and perspectives.

II. ARCHITECTURE OF A SELF-* PEDAGOGICAL AGENT

Within this section we present the architecture of the Self-* Pedagogical Agent. It aims to provide on the fly reconfiguration of the Pedagogical Agent's structure. The self-* is achieved by the Manager Sub-System (MrSS) that self-(re)configures the Managed Sub-System (MdSS) to implement the appropriate Pedagogical Agent regarding the student state. The internal structure of the Managed and the Manger Sub-Systems is designed using the Fractal Component Model as it is represented by the *Figure 1*.

A. *Manager Sub-System*

The Autonomic Manager Sub-System (MrSS) is an Autonomic entity. It is a distinct Component that is responsible of the adapting of the Pedagogical Agent's internal structure. The triggering of the self-adapting is achieved through the four stages of the MAPE-K mo del. First, there is the monitoring of the Student Cognitive Level by listening to notification received from the WildCAT Observable Bayesian Student Mo del. Next, It analyzes the student outcomes to decide if an adaptation is needed and what it concerns, Planning the appropriate changes to make and Executing the adapting to the appropriate components. It triggers (re-)assembling to implement the appropriate Pedagogical Strategy.

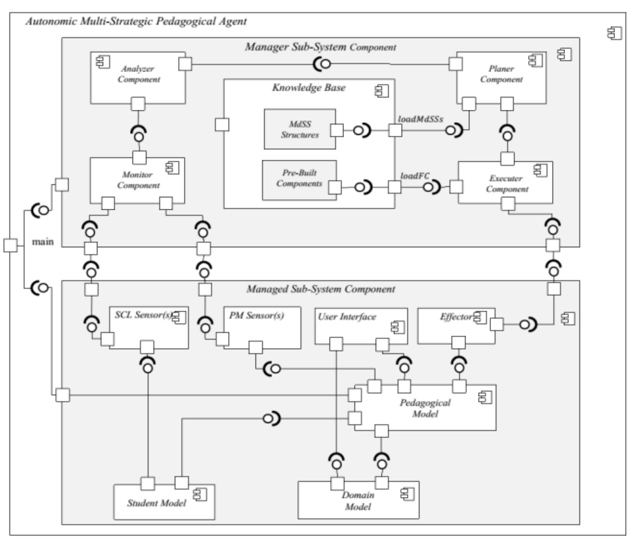

Figure 1. Architecture of the Self-* Pedagogical Agent.

B. *Managed Sub-System*

The Managed Sub-System (*MdSS*) aims to provide a fine-tuned, cost-effective and flexible Pedagogical Agent building and adapting. It is a composite component structured of sub-components and bindings between the required interfaces and the provided interfaces to each other. The sub-components concerns the well known pedagogical system that are the Domain Model, the Student Model, the Pedagogical Model and the User Interface components. It is built by the assembling of pre-built approved Components. Also, maintaining a *MdSS* is simplified to the replacement of one or more of its sub-components by a pre-built version of the corresponding sub-component.

III. OBSERVABLE BAYESIAN STUDENT MODEL

The proposed Observable Bayesian Student Model is presented by justifying the student feature chosen to achieve the adaptation, the used formalism to represent this feature and the implementation of the resulted model using WildCAT.

A. *Student Cognitive Levels*

The Student Cognitive Level (SCL) is the feature mostly used in Pedagogical Systems. It is evaluated at the concept level and it reflects the understanding of the learner. Abou-Jaoude and Frasson [9] have defined a Student Model with four different knowledge levels presented in Table 1: 1) Novice, 2) Beginner, 3) Intermediate and 4) the Expert. It ranges from no prior understanding of the concept at all to extensive understanding. At the Expert Level, the student have acquired extensive knowledge that affects what they notice and how they organize, represent, and interpret information in their environment. This, in turn, affects their abilities to remember, reason, and solve problems [10].

Table 1. Student Cognitive Levels [9].

Knowledge level (% of concept understanding)	**Explanation**
Novice (0 %)	No prior knowledge of the course at all
Beginner (10–30%)	The student is being familiar with the course structure and main beginner concepts of the course but lacks in practice and he/she is expected to answer basic questions correctly
Intermed. (40–60%)	The student was learning the intermediate concepts of the course and he/she is expected to answer and solve, correctly most of the intermediate questions
Expert (80 –100 %)	The student knows the course very well; he/she is expected to answer most of the expert questions and problems correctly

B. *Bayesian Network Modeling for Prerequisite Relationship*

The Bayesian Student Model $SM = (G, \theta).G = (C, R)$ is a Directed Acyclic Graph (DAG) where C represents the knowledge level of the student in the concepts set C and R represents the "prerequisite-of" and the "explained-by" interdependency relationships. $\theta = \{P(Ci|Pa(Ci))\}$. A set of probability for each vertex Ci conditionally to the state of its $Pa(Ci))$ in G. A set of variable $C = \{Ci, \ldots, Cn\}$ associated to the graph as:

$$P(C_i, \ldots, C_n) = \sum_{c \in C} P(Ci|Pa(Ci)) \qquad (1)$$

$Pa(C_i)$ a set of parents of C_i.

The Bayesian Network of the *Figure 2* is constructed using the example, the probabilities and the evidence e_1: $P(find\ div = known) = 0)$ introduced in [11, 12].

C. *WildCAT representation*

The Student Cognitive Level (SCL) is represented using WildCAT. The choice of WildCAT monitoring framework is motivated by the simplicity and the dynamic of its data model that is suitable to represent the SCL frequent changes and the Learning Path of the pedagogical systems execution context. Also, WildCAT's data acquisition sub-framework hides the data acquisition [13] that does not increase the complexity of such complex system. the tree structure of WildCAT is appropriate to represent the prerequisite based knowledge structure. Also, WildCAT OW2 [14] use Resources and Attributes node to represent the contextual domains. This is appropriate to capture the knowledge level evaluated at the concept level. Another advantage of WildCAT is its definition of synthetic attributes as they compute higher-level or aggregated information from low-level data acquired from data sources. The Synthetic attributes are dynamically and automatically up dated when their dependencies change [13]. We use these synthetic attributes to compute the Instantaneous learning outcomes of the student in a concept given dynamic changes in its prerequisite sub-components using equation (1). Furthermore, WildCAT proposes an event-driven structure with publishers/subscribers where context changes are represented as events. We exploit this ability in monitoring the Student Mo del without decreasing the performance of the Self-* Pedagogical Agent.

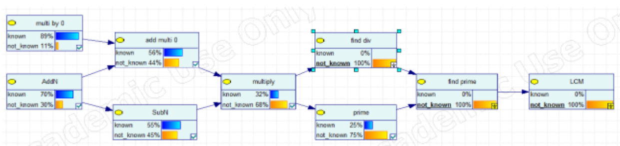

Figure 2. Bayesian Network for a Prerequisite Relationship using GeNIe (BayesFusion, LLC).

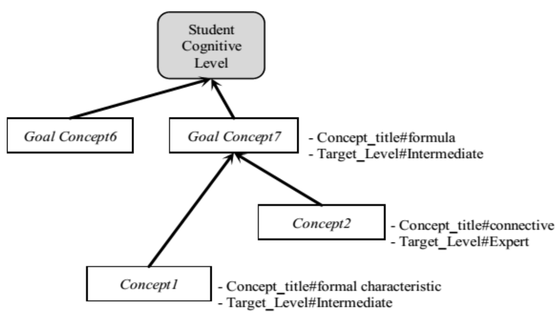

Figure 3. Resource hierarchy for the Student Cognitive Level Contextual Domain.

The Student Cognitive Level reflects the learner's understanding of domain's concepts using a given selected pedagogical strategy. Here, the learning outcomes of the student are captured and structured using the WildCAT's data model.

The WildCAT data mo del is used to represent the context (Student Mo del). WildCAT can represent the Student Mo del of the Multistrategic Self-Switching Pedagogical Agent in the form of a set of contextual domains. It means that the Context of WildCAT represent the Student Mo del. In our case, we focus on a Context Domain for the Student Cognitive Level to achieve the self-switching. It is identified by the unique name "StudentCognitiveLevel". Because the Student Cognitive State is evaluated at the concept level, the WildCAT's data mo del will b e an augmented version of the domain mo del' concepts. It is defined and organized using resources and attributes associated to the concepts of the domain mo del. The Resources of the WildCAT's data mo del are the Goal Concepts and the concepts constituting the Learning Plans. Also, there are the explanations, the current pedagogical strategy and the level resource tree rooted in the "StudentCognitiveLevel". Concerning the relationships between resources, we have the has-prerequisite, has-explanation, has-evaluation, has-level and at-strategy. The Attributes hold values that concern: the selected goal concepts, the current Pedagogical Strategy, the Student Cognitive Levels associated to a given concept and the evaluations for each Pedagogical Strategy.

```
Path: "self://StudentMo del/StudentCognitiveLevel/GoalConcepts#ConceptA"
Path: ../GoalConce pts/C onceptA/TutorTutee_Explanation/Test2#8.5"
Path: _ ../GoalConcepts/ConceptA/Level#Novice
Path: _ ../GoalConcepts/ConceptA/CurrentStrategy#TutorTutee"
Path: ../GoalConce pts/C onceptA/ConceptB/TutorTutee_Explanation/Test4#5.5"
Path: _ ../GoalConcepts/ConceptA/ConceptB/CurrentStrategy#LearningByDisturbing
Path: _ ../GoalConcepts/ConceptA/ConceptC/LearningCompanion_Explanation/Test1#5.5"
```

The Student Cognitive Level is designed as a shared component between the Pedagogical and the Monitoring Components. They update the knowledge level and intercept the updates respectively. The update of the Student Model is achieved either on a Goal Concept selection, an explanation or an evaluation of/on a given concept. Note that, there is no static data mo del. The tree branch of WildCAT's data model corresponding to the chosen Goal Concept is created dynamically and automatically. The observation of the Student Cognitive Model is done through probes. Each probe is responsible for observing the Student Cognitive Level in a particular concept of the Student Mo del, and notifies the Monitoring Component about that evaluation.

IV. Observable Bayesian Student Mo del at Work

The Observable Bayesian Student Mo del is a shared component between the Pedagogical and the Monitoring Components. The first one up dates the WildCAT's Data Mo del and the second monitors it.

A. Construction

The User Interface Component(the graphical part) presents the knowledge structure in the form of a graphical tree. During a learning session, through this UI, the student can select a given concepts that we call Goal Concept (see *Figure 5*). The Goal Concept and the Knowledge Structure are used to construct the Learning Plan. This latter is constituted of an ordered set of concept needed to achieve the Learning Goal as shown in the *Figure 7*.

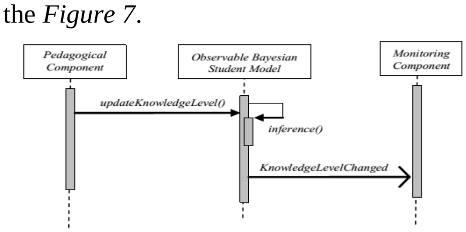

Figure 4. Student Model Management.

B. Update

Here, we focus on the presentation of the Pedagogical Component from the viewpoint of its interface with the Student Cognitive Level. The update of the Bayesian Student Mo del is achieved at concept level after the selection of a Learning Goal or a test session taken by the student concerning the Learning Concept. At this stage, WildCAT will b e up dated to contain the new Learning Path (The set of concepts that are in the of the Goal Concept) or the Student Cognitive Level(s).

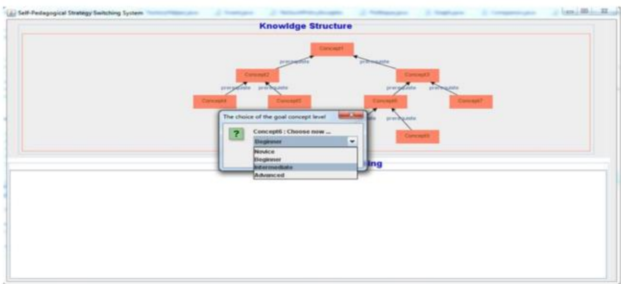

Figure 5. Knowledge Structure.

The Pedagogical Component is responsible of the teaching and the evaluation of the student. After a given test, knows the state of student is the Learning Concept variable (called observation variable lc in Bayesian Network terms). Using the equation (1), the Pedagogical Component determines the probabilities of the Learning Concept's descendant variables (called target variables CL_C in Bayesian Network terms) conditional on the observations $P\ (CL_C \mid lc\ )$.

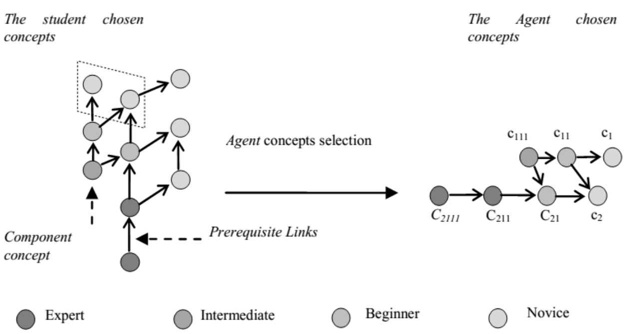

Figure 6. Learning Plan Construction.

WildCAT permits the data model to evolve over time, both at the values of the attributes and at the structure of the model itself [15]. For this reason, WildCAT provides pull mode (synchronous requests). The Pedagogical Component uses this service to dynamically evolve the student model implemented by the WildCAT's data model. From an implementation point of view, it is achieved by the discovery and the interrogation methods of the *Context* class parameterized by the *Path* class. The modification achieved by the pedagogical component can touch the following elements:

- **Modification of the attributes' values**. It concerns the change of the student cognitive level, the change of the current strategy, the evaluation result in a given concept, etc.
- **Add or remove of resources**. It concerns the selection or the canceling of a new a Goal Concept respectively

C. Monitoring

The monitoring of the Student Cognitive Level is achieved by listening to notifications received about the learner outcomes. Next, It analyzes the student outcomes to decide if a PSS is needed and finding the best SCL-PS matching and, Planning the appropriate changes to make and Executing the switching of the appropriate components. It triggers (re-) assembling to implement the appropriate Pedagogical Strategy.

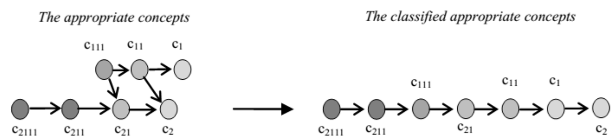

Figure /. Concepts Scheduling.

The Monitoring Component is responsible of the monitoring of the Student Cognitive Level up dates. It achieves the monitoring of the Student Cognitive Level by listening to notifications received about the learner outcomes. Because the runtime context of the Self-* Pedagogical Agent is extremely dynamic, the Monitoring Component uses the push mo de (Asynchronous Notifications) of WildCAT. It uses the second interface of the *Context* class to subscribe as a listener of specific generated events. In addition, it uses simpler $ATTRIBUTE\_CHANGED$ event kind to pass the decision of the self-* to the Analyzer Component. So, this interface permits to the Self-* Pedagogical Agent to perceive its context that is the Student Cognitive Level. The interface *ContextListener* is implemented by the Content class of the Monitoring Component to be notified of changes in the Student Model Component as a consequence of learning. It is responsible of the receiving of the event resulting of the Resources and Attributes changes. It uses Active Attributes that are the sensors related to the Student Cognitive Level. The Monitoring Component is inscribed as a listener only for the changes of Student Cognitive Level's. So, It can perceive the Student levels changes during the usage of the Self-* Pedagogical Agent.

The WildCAT based Observable Bayesian Student Model lets to inspect its contextual content by registering queries on the events generated by the hierarchy. It allows to determine for every event, the concept in the hierarchy which emitted the event. WildCAT has event type emitted by an attribute that indicates the modification of that attribute and hold its new value [14]. For this case, we have a private attribute *'ConceptSensor'* that is a WildCAT sensor. This attribute will notify a WildCATConcept activity in methods *'increaseSCL'* and *'decreaseSCL'*. We consider the new SCL value as the monitoring data to be notified by this sensor. This sensor is attached to a WildCAT context to notify such events.

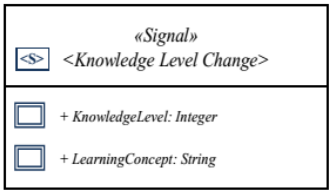

Figure 8. Knowledge Level Notification.

V. USE CASE: MULTISTRATEGIC-PEDAGOGICAL SYSTEMS

Multistrategic Pedagogical Systems (MPSs) achieve effective learning by reproducing the flexibility of human teachers switching of the teaching methods. This vision fills the gap of the one size fit all philosophy of mono-strategic pedagogical systems by integrating multiple SCS related pedagogical strategies. Since, each strategy has specific advantages and it appears useful to use adequately the strategy that will strengthen the acquisition process for a given learner [16]. Also, PSs are appropriate regarding the Student Cognitive Level (SCL), the learning style and the personality. Furthermore, according to Aïmeur, Dufort, Leibu, and Frasson [17], it is necessary to have different tutoring strategies since: 1) Different domains require different approaches (a single teaching method would not work in a multi-domain teaching environment [18]); 2) The variation of teaching strategies serves as a means to maintain the interest and motivation of the learner; 3) Different tutoring strategies fulfill different goals and develop different abilities in the learner. For example, different knowledge levels need different teaching strategies. So, using the learning by disturbing strategy is not however suitable for all the various kinds of students [16]. So, it is necessary to associate the appropriate Pedagogical Strategy (PS) for each learner pattern to increase the learning outcomes. According to Abou-Jaoude and Frasson [9], the flexibility of the *Intelligent Tutoring System (ITS)* can be enhanced using multiple learning strategies that can be successively triggered depending on the progression of learning. Consequently, selecting the appropriate PS in a MPSS regarding the SCS strengthen the understanding and makes the learner doing well. Regarding the unpredictable nature of the SCS changes and to imitate the on the fly human teachers switching, MPSSs dynamically switching their PS to match with these changes. They accomplish the switching by monitoring the SCS and triggering the PSS when it is required. This adapting is achieved by an integrated PSS logic according to the SCS updated and stored in a Student Model (SD). As a conclusion, a good learning strategy should be selected according to the two facets of the learner model: information about the knowledge level of the learner (which is a cognitive part) and information about his affective characteristics (the affective part) [19]. That's why, MPSSs are important to increase the cognitive and the meta-cognitive abilities of the students.

The *Figure 9* shows a simulation of the learning process of the concepts constituting the learning plan. It represent the current concept, the resulted evaluation and the WildCAT resources affected by the learning process.

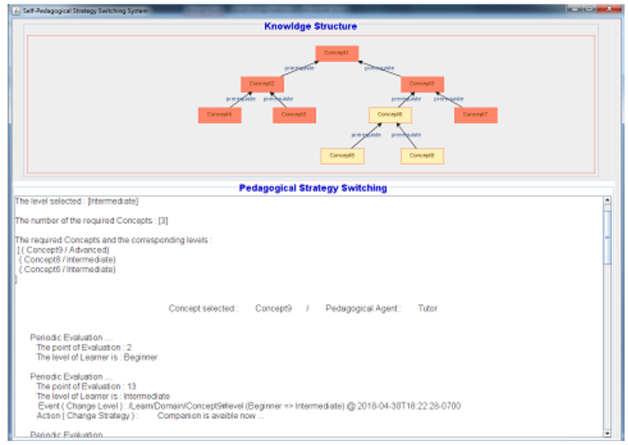

Figure 9. Strategy Switching of the MSSPA.

VI. CONCLUSIONS AND PERSPECTIVES

Within this paper we have presented an Observable Bayesian Student Model using WildCAT. Against the conventional Student Model and in addition to the efficiency and the flexibility, this model is observable. It means that it can be used with Self-* Pedagogical Systems. As a perspective, we work on the design of an Observable Bayesian Student Model for Massive Multistrategic Self-Switching Pedagogical Agent.

REFERENCES

[1] E. Millán, T. Loboda, and J. L. Pérez-de-la-Cruz, *Bayesian networks for student model engineering*. Computers & Education, vol. 55, no. 4, pp. 1663–1683, 2010.

[2] L. Nguyen and P. Do, *Combination of Bayesian network and overlay model in user modeling*. In International Conference on Computational Science, Springer, 2009, pp. 5–14.

[3] K. Chrysafiadi and M. Virvou, *Student modeling approaches: A literature review for the last decade*. Expert Systems with Applications, vol. 40, no. 11, pp. 4715–4729, 2013.

[4] M. C. Desmarais and R. S. Baker, *A review of recent advances in learner and skill modeling in intelligent learning environments*. User Modeling and User-Adapted Interaction, vol. 22, no. 1-2, pp. 9–38, 2012.

[5] M. C. Huebscher and J. A. McCann, *A survey of Autonomic Computing —Degrees, Models, and Applications*. ACM Computing Surveys (CSUR), vol. 40, no. 3, p. 7, 2008.

[6] M. Salehie and L. Tahvildari, "Self-Adaptive Software: Landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 4, no. 2, p. 14, 2009.

[7] D. Garlan, S.-W. Cheng, A.-C. Huang, B. Schmerl, and P. Steenkiste, *Rainbow: Architecture-based self-adaptation with reusable infrastructure*. Computer, vol. 37, no. 10, pp. 46–54, 2004.

[8] R. Laddaga and P. Robertson, *Self adaptive software: A position paper*. In SELF-STAR: International Workshop on Self-* Properties in Complex Information Systems, Citeseer, vol. 31, 2004, p. 19.

[9] S. C. Abou-Jaoude and C. Frasson, *An agent for selecting learning strategy*. in *Proceedings of the World Conference on Nouvelles*

Technologies de la Communication et de la Formation (NTICF), Rouen, 1998, pp. 353–358.

[10] J. D. Bransford, A. L. Brown, and R. R. Cocking, *How people learn: Brain, mind, experience, and school.* National Academy Press, 1999.

[11] C. Carmona, E. Millán, J.-L. Pérez-de-la-Cruz, M. Trella, and R. Conejo, *Introducing prerequisite relations in a multi-layered Bayesian student model*. In International Conference on User Modeling, Springer, 2005, pp. 347–356.

[12] C. E. Dowling, C. Hockemeyer, and A. H. Ludwig, *Adaptive assessment and training using the neighbourhood of knowledge states*. In International Conference on Intelligent Tutoring Systems, Springer, 1996, pp. 578–586.

[13] P.-C. David and T. Ledoux, *WildCAT: A generic framework for contextaware applications*. In Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, ACM Press New York, NY, USA, 2005, pp. 1–7.

[14] OW2. (2018). OW2 WildCAT User Guide, version 2.3.0.n, [Online]. Available: http://wildcat.ow2.org/userguide.html (visited on 04/30/2018).

[15] P.-C. David, *Développement de composants fractal adaptatifs: Un langage dédié à l'aspect d'adaptation*. PhD thesis, Université de Nantes, 2005.

[16] C. Frasson, T. Mengelle, and E. Aïmeur, *Using Pedagogical Agents in a Multi-Strategic Intelligent Tutoring System*. In Workshop on Pedagogical agents in AI-ED, vol. 97, 1997, pp. 40–47.

[17] E. Aïmeur, H. Dufort, D. Leibu, and C. Frasson, *Some justifications for the learning by disturbing strategy*. In Proceedings of the Eighth World Conference on Artificial Intelligence in Education, Citeseer, 1997, pp. 119–126.

[18] R. Freedman, S. S. Ali, and S. McRoy, *Links: What is an Intelligent Tutoring System?.* Intelligence, vol. 11, no. 3, pp. 15–16, 2000.

[19] C. Frasson and E. Aïmeur, *A comparison of three Learning Strategies in Intelligent Tutoring Systems*. *Journal of Educational Computing Research*, vol. 14, no. 4, pp. 371–383, 1996.