



A software architecture perspective about Moodle flexibility for supporting empirical research of teaching theories

Marcelo Campo, et al. [full author details at the end of the article]

Received: 23 April 2020 / Accepted: 23 July 2020/Published online: 31 July 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Moodle represents a great contribution to the educational world since it provides an evolving platform for Virtual Learning Management Systems (VLMS) that became a standard de facto for most of the educational institutions around the world. Through the pedagogical functions provided, it collects in the many globally spread out databases a huge amount of information regarding the activities that teachers and students perform during the learning process. This reality makes Moodle a natural choice for conducting experimental research by Artificial Intelligence researchers interested in theories for improving learning and teaching; particularly those related with the controversial *learning styles* concept. Roughly defined, a learning style intends to be a model of the way and media an apprentice acquires knowledge and hence the way a teacher should present that knowledge to the apprentice matching his/her learning style. Independently of the many controversies (be these scientific, psychological or even ethical) about the soundness and real outcomes that such ideas can bring to improve learning, it's a worthy intriguing research area for many researchers pursuing the ideal automated teacher: *the teachbot dream*. Behind this goal we have developed Middle, a Moodle plug-in able to infer the learning style of each student taking a course using an advanced version of a Bayesian network model that we previously tested. Middle intends support *personalized* teaching based on the Felder-Silverman's ILS model and has been evaluated through controlled experiments and pilot test in high schools and university courses. Such experiments showed promising results that shed some light on learning styles modeling and its potential outcomes. During the experience we found strong limitations in the Moodle design regarding its supposed flexibility to incorporate new functionalities. From a strict software architecture point of view, we found that such flexibility is far from being enough to easier the implementation of the dynamic computational behavior required to support a *teachbot*. This made our effort much harder than expected, perhaps because of the illusion induced by the widespread use of Moodle. In this article we present our results and experiences extending Moddle with intelligent behavior from a software architecture point of view, focusing on the lessons learnt in such extension. Our experience indicates that this simplicity is far from being so and hence it is worth to share the limitations and how we overcome them.

Keywords Learning styles · Bayesian networks · Moodle · Virtual learning environments

1 Introduction

Moodle represents a great contribution to the educational world since it provides an evolving platform for Virtual Learning Management Systems (VLMS) that became a standard de facto for most of the educational institution around the world. One reason relies on it satisfies adequately the requirements of a VLMS not only giving a visual space for learning activities, but also provides active resources to maintain grade books, generate competency-based learning plans, track learning, manage communication spaces, as well as to manage the evaluation process through different pedagogical tools.

Given that it is developed under the open-source philosophy it allows developers to add new functionalities through the well known plug-in mechanism commonly used by web-browsers to extend the core functionality. So, it presented a natural choice for experimental research on theories for improving learning and teaching because, through the pedagogical functions, it collects in its database a huge amount of information regarding the activities that teachers and students perform during the learning process. This information can be then analyzed by “intelligent” data analysis tools.

In certain way it could provide a vehicle to enable the exploration of machine learning techniques that go beyond the old expert systems metaphor pursuing an ideal automated learning approach that we refer to as a “teachbot”. A *teachbot* would be an intelligent software agent (Wooldridge and Jennings 1995) that ideally implements a bi-directional learning loop among teachers and students in which the teacher based on the knowledge of both subject area and student performance in previous courses. With this information a teachbot would design a course personalized for each student, acting as a teacher, a coach or even a friend obtaining hence a better learning experience inside the ideal cycle proposed by Kolb among others (Kolb 2014), and (Felder and Silverman 1988), (Fleming 1995).

Simultaneously, the *teachbot* would adjust its inferences going through a continuous improvement process as usually we say in the software development world.

Independently of the many controversies (be these scientific, psychological or even ethical) about the soundness and real outcomes that such ideas can bring to improve learning (Lilienfeld et al. 2010), (Coffield et al. 2004), (Kirschner 2017), (Knoll et al. 2017), (Rogowsky et al. 2020), (Cuevas 2020) it’s a worthy intriguing research area for many researchers in Artificial Intelligence. The main premise is that the pursued research goals could enlighten the way towards confirming some learning theories. However, from a strict software architecture point of view, we found that Moodle flexibility is not enough to easier the implementation of the dynamic computational behavior needed to support a seamless software extension to these ends.

Most of the reported works in this line have to resort to extracting information from the raw database and apply different machine learning techniques for creating datasets subsequently analyzed with external tools like WEKA (Witten et al. 2016). The automatically inferred results are used to contrast them with the ones obtained usually by pencil-and-paper tests, but in some approaches are used as a feedback for the teacher deciding modify or correct, generally, the whole course design. That is, they got interesting insights regarding the suitability and accuracy of a given machine learning approach but they are very far yet from the aim of truly *personalizing* the learning process.

Architecturally speaking this limitation is due to the lack of an externally accessible built-in mechanism for capturing the user interactions (student or teacher) with Moodle. This would make it easier data gathering about their behavior in real-time. Also it plays against the idea of emulating a coach acting like the recommender subsystems (Bobadilla et al. 2013) quite common in many today's information systems, for example. In the current Moodle design is almost impossible without a very important effort.

We arrived to these conclusions when we started the Middle project, as we named it, sponsored by the education bureau of Argentina, which mean thousands of students spread out across the most populated state of the country at the primary and secondary school levels, and Moodle is the common platform for public schools.

So, we started the development of a plug-in for extending Moodle to analyze individual student's interactions for constructing a personalized Bayesian Network (Jensen 1996) through which the student's ILS learning style (Felder and Silverman 1988) is inferred. This Bayesian approach reported in García et al. 2005, García et al. 2007) enabled to gradually organize the presentation of subsequent courses in an autonomous way, as well as to draw conclusions and predictions of their future behavior to teachers.

This extension that, in principle, was thought as naturally simple however was far from being so. Even for software specialists get a detailed knowledge about the Moodle design and its particular implementation is cumbersome. This aspect showed us the limitations of the intended Moodle flexibility to cope with changes and extensions.¹

In this article we present our experiences from a software architecture point of view, focusing on the lessons learnt extending Moodle under a narrow timeframe. It not intends criticize Moodle qualities even less the quality of its developers, just we believe that it is worthy to constructively share the limitations we found and how we overcome them in order to they could be useful for future releases.

The article is organized as follows. Section two provides the context in under which we based our research and describe the Middle goals. Section 3 is devoted to analyze the different architectural concerns regarding Moodle flexibility and the way Moodle was extended emphasizing the problems and solutions found from a software architecture perspective. The second part describes the mapping of ILS concepts to a working architecture and the Bayesian implementation. After this, experimental results are presented and finally some conclusions outlined.

2 The middle goal

The Middle project aims for the ultimate autonomous teacher, a *teachbot*. Certainly, an unattainable goal but the fuel that feed the search to what extent Artificial Intelligence techniques based on learning and teaching theories, as for example the controversial *learning styles*, can contribute to improve learning performance of students. In other

¹ This became more evident, at least in Argentina, with the unexpected misfortune brought in by the COVID-19 which imposed the need of suddenly adapt all the educational system from conventional face-to-face courses to remote and virtual ones. This complex situation ended overcoming Moodle adaptability mechanisms leading to undesired solutions plenty-of- seams.

words, Middle aims at providing an intelligent tool for empirically allow education researchers exploring to what extent such a theory holds, not as an absolute universal law, but as an additional vehicle that could help to improve student and teachers' performance.

2.1 Rationale

Roughly defined, a *learning style* intends to be a model of the way and media an apprentice acquires knowledge and hence the way a teacher should present that knowledge to the apprentice matching his/her learning style. Since its diffusion in the seventies until today, as many theories, this concept has gained many supporters as well as critics about its scientific basis and real outcomes which varies in intensity from true believers, passing through agnostics, skeptical up to strong detractors (Kolb 2014), (Felder and Spurlin 2005), (Scott et al. 2014), (Pashler et al. 2008), (Cuevas 2020). The last critic opinions are based on well sound arguments (from psychological to neuroscience foundations), but in general suffer in certain way the same lack of enough empirical data that lead undoubtedly to an absolute scientific refutation.

Ever since we face the learning style theories we adopted an *agnostic proactive* position for exploring the extent of the usefulness of the Felder-Silverman's Index of Learning Styles in the complex area of Software Engineering Education. We refer to it as complex not only because of the rapid technological changes in which it is immersed, but due also to it involves rapid evolving complex areas as Management, Business and, particularly, Group Psychology that varies from one generation to another (i.e. gen x to z).

During several years we explored the meshing hypothesis mixing conventional face-to-face classes and VLE-based courses carried out as follows:

- Face-to-face: Considering the impossibility of giving a personalized attention to each student and take measures against conventional course, we divide the students into two clusters considering an average value of the ILS Intuitive-Sensitive and Global-Sequential dimensions (we disregarded the other two ones because the number of Reflective and Verbal was statistically almost irrelevant). The first cluster was instructed following a designed course considering the characteristics of the average ILS. The second cluster was instructed following the conventional style we used to dictate the course.
- VLE experiments: They were oriented towards exploring the automatic detection of individual learning styles and its application in an automated course designed for similar ILSs in a personalized way for Scrum Method for software development. In this case we use Virtual Scrum, a VLE environment described in (Rodriguez et al. 2013) enhanced with a somehow *naïve teachbot* (an action recommender indeed). This assistant helps under request apprentices with suggestions about the most convenient step to follow while developing a mandatory task defined by the method.

These results were not conclusive for demonstrating the certainty of the hypothesis sustained by the ILS concept. However, they shed some light on a potential

contribution to enhance learning results *when it is well-understood and applied by trainers* as Felder and Spurlin (2005) affirm.

In the face-to-face case students of the ILS cluster showed a relatively significant better performance in the overall evaluations than the other cluster. But in the second case we discover an intriguing clue: students that got significant better results were those that requested the *teachbot* assistance while developing Scrum-related tasks in the project development having high/medium Sensitive-Sequential ILS, but notably slower than the Intuitive-Global ones who got just acceptable results.

The analytical results of these experiences can be found in (García et al. 2007) (Rodríguez et al. 2013) (Feldman et al. 2014) (Scott et al. 2014) (Scott et al. 2016).

Somehow, we converge with Pashler et al. (2008) who conducted experiments that yield suggestions about the weak sustain of the theory, regarding the fact that more experiments based on a strong scientific design over a broad scope of samples and extended in time are necessary for arriving to a well sustained conclusion.

The results obtained in the VLE experience encouraged us to extend our experimentation and Moodle appears as the natural choice for supporting it through ILS inference.

2.2 Inferring personal learning styles and the *TeachBot* metaphor

Theoretically a teacher is supposed to first know the student's given by the individual ILS preferences and then adapt the course presentation according such preferences. To get this prior information we decided to emulate the behavior of a teacher in the process of knowing a student. Thus, during the teaching, each action taken by each student is incrementally observed. In this way the student database should be expanded with each interaction, increasing the knowledge about student's behavior in different courses and reapplying the machine learning approach iteratively, given an observation threshold, a more accurate ILS profile is built. This strategy also would emulate the test-retest one reported by several researchers (Rosati and Felder 1995; Zywno 2003; Seery et al. 2003).

A Bayesian Network (BN) represented a very malleable mechanism for inferring Learning Styles. The BN formalism is based on the Bayes theorem,

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

which relates conditional and marginal probabilities. It yields the conditional probability distribution of a random variable A, assuming we know: information about another variable B in terms of the conditional probability distribution of B given A, and the marginal probability distribution of A alone. In this case the variables are contained in the ILS vector that is defined by four dimensions:

<Perception, Input, Processing, Understanding >.

A BN also represents a particular probability distribution of the joint distribution over all the variables represented by nodes in a directed acyclic graph where nodes represent

random variables and arcs represent probabilistic correlation between variables (Jensen 1996). That is, the probability of a student to have an *Intuitive* preference can be assimilated to the level in the scale that results of filling the ILS questionnaire contrasted against with the level that appears the *Sensing* preference. The same reasoning holds for the other dimensions.

The essential problem resides on how to infer a reasonably accurate value for each dimension in an incremental way. In consequence, what clues are supposed to lead to such inference value? For example, one objective is to infer whether a student prefers visual or verbal material, and then the hints should be related to the student's actions while taking the courses. In this case the obvious clue is the format of the didactic material chosen by the student regarding its graphic, written presentation, spoken, etc. Middle provides, in this way, support to help identify the most likely learning style that defines the student's profile.

Thus, for example, we need to record how many examples the student analyze before correctly applying the topic to be learnt. This is an indication that the subject has to feel better through practical examples to understand an abstraction. This suggests then how intuitive or sensitive the student could be. And we use the same procedure for each indicator defined by Felder's learning styles.

In order to infer dynamically the personal ILS vector of a student in Moodle is necessary extending its functionality for incrementally building the students' profile based on the observation of the behavior during their learning process. This implies developing new software components implementing this new functionality that ideally should be *seamlessly* added. This *seamlessness* strongly depends on the *flexibility* of the underlying software architecture for extending it as well as the mechanism provided to non-intrusively intercept the interactions of the user if there is any.

3 Analyzing Moodle architectural flexibility

In order to define a common vocabulary and make it easier the understanding of the analysis the following section briefly presents the basics architectural concepts that will be used to describe the limitations we found and the solutions we propose. Readers used to these concepts can skip to section 3.2.

3.1 Software architecture

Software Architecture refers generically to the division of a software system into a set of components, their externally visible relationships and the patterns that defines the collaboration among them to accomplish a given functionality (Bass et al. 2013).

Since its systematized definition at the end of the nineties Software Architecture has became in the standard vehicle to describe and design software systems, particularly large scale ones. As it provides a common vocabulary not just to model the software global design through its structures and its behavior attending business goals, but also *prescribes* analyzable solutions for broad domains in function of the so-called Architecture Styles and Quality Attributes.

3.1.1 Architecture styles and object-oriented frameworks

A Software Architectural Style is an *abstract* description of functional components and the way they interact to accomplish a given functionality that are usually found as the underlying organization of many software systems.

These styles are collected into a catalog that includes the most common organization forms as Client-Server (i.e., distributed systems), Repository (i.e. data centered systems), Layered Systems (i.e., operating systems), Multi-Tier (i.e. enterprise systems), Event-Driven (i.e., control systems), Microkernel (i.e. service-oriented systems), among the most common. Normally large systems follow a combination of these styles that define its *concrete architecture* by establishing interface protocols through which the concrete functional components accomplish the system function.

When a concrete architecture describes abstractions of a particular domain it presents the important opportunity of both code and design reuse. Depending on the implementation technology that materializes (Campo et al. 2002) the architecture they are referred to as Product-Line Architectures (Coplien et al. 1998) or Object-Oriented Application Frameworks (Fayad et al. 1999). Frameworks, for short, became in the standard way to provide infrastructure platforms. Due to its constant development the obtained experience led to several catalogs of design patterns (Gamma et al. 1995) (Fowler 2002). Design Patterns are recurring reusable design solutions enabling systematic design methods based on the proper combination of such previously known solutions, as happens in other engineering disciplines.

3.1.2 Software architecture quality attributes

Software architectures are characterized through Quality Attributes (QA). QAs define different dimensions of *desirable* computational properties that the *design* of a system must satisfy. These properties, formerly referred to as non-functional requirements, are related to the structure and interaction of the software components that compose the system (internal or constructive perspective) as well as the externally perceivable behavior of the running system (external or dynamic perspective).

In the external classification, performance, availability and reliability, among others, quantify time-related system properties (i.e. response time, time between failures, etc.) whereas security and usability, for example, define quantifiable human-related system properties (i.e., use easiness, robustness against hacking). In the internal one, among many, extensibility, flexibility, understandability, readability, and reusability are examples of the most important related to our research.

Extensibility refers to how simple it is to add new functionalities or features to an existing architecture. For this purpose different mechanisms can be used when designing the architecture. Flexibility, often treated as a synonym, refers to the mechanisms themselves that enables a greater ability to introduce different implementations of such functionalities (statically or dynamically). Some authors refers to these QAs as Adaptability, but adaptability, under our view, is most related to the ability for using the same architecture to solve problems in different domains not modifying it to cope with problems of the new domain. Modifiability, on its part, becomes a key QA when a Product Line Architecture or a Framework is going to be developed or used.

Strongly related to these QAs appear Understandability and Readability. Both of them involve the ways in which the architecture is described for enabling its extension. Understandability speaks of how much effort is needed to grasp the abstract both static and dynamic structures of the system. Readability is about how clear and comprehensive it is the architecture description through adequate notations as vehicle to make easier the understanding. Modifiability is highly dependent on these two tight related QAs. Extensibility hence is also influenced by them.

As many of these properties can conflict between them (i.e., Performance vs. Modifiability is a common tradeoff) designing architecture implies to *decide* ways that provide an adequate balance among different tradeoffs. Understanding these decisions is essential for the easy extension of any architecture, so traceability plays a crucial role that must be supported by good-quality documentation.

3.2 An insight into Moodle extensibility

As we stated above Middle objective is try to simulate the behavior of a teacher or coach observing and registering the way a given student interacts with a Moodle course and to provide support according the learning style. For that it is necessary to intercept all the interactions of the student using Moodle and register each event in order to build and feed his/her Bayesian network. This would be the main design driver behind Middle.

There are two possible ways for extending Moodle to implement the necessary functionality to try achieving the *teachbot* metaphor: use the Moodle's Inspire framework for machine-learning-based techniques or implement the Middle functionality from scratch. In both cases it is unavoidable first *understand* Moodle architecture in order to afterward understand the Inspire one for reusing it.

3.2.1 On understandability and architecture style

As we stated above documentation is crucial, particularly if a product is intended in its conception for being highly extensible, as Moodle is supposed to be. In this sense, the provided documentation, although enough in amount, lacks consistency and uniformity regarding description languages and modeling techniques.

For example there are many drawings and charts of different contributors using different notations from informal text and drawings to more UML complaint charts. There is no structured explanation how the Domain Model is mapped to the reference architecture. This implies that much effort is needed in terms of time spent on finding and understanding the abstract behavior, in other words, the architecture itself.

Architecture styles and concrete architecture As having being developed under the Open Source philosophy it is evident that Moodle did not follow a formal architecture design methodology. So, the effect of the agile idea that the software architecture “naturally” emerge from implementation leads to a limitations when extensibility and particularly flexibility is one important design driver.

The fact of adopting PHP as the initial development 'platform serves as an additional condiment. PHP is a popular platform for fast development that does not requires a strong formation and experience beyond programming. However, it does not represent

the best choice when evolution and maintenance are involved, simply because it was not born to cope with ever evolving big and complex developments.

The concept of having a “functionality Core” get easily depleted when new functionalities beyond the original intent want to be added. This lead to generate new versions that sometimes obliges to re-implement plug-ins to adapt them to the changes (this was our case when Inspire was included into the core and was impossible to build the solution without an unaffordable cost at this time, as we explain further below).

Beyond this, although the architecture is supposed to be designed with extensibility and flexibility as main drivers it’s somehow unexpected to find that the descriptions are fairly informal and is described as if it would respond to different architectural styles from which it really is based on.

It is rather clear that the Repository architectural style is the predominant style following a Client-Server interaction model. Moreover, as one of the main design drivers was flexibility, the plug-in model, typical of Microkernel pattern, is a good decision, but documented as the “Core “and explained as if it were a Layered Architecture (abstractions stack), when it is a classical Three-Tier (User Interface <=> Application <=> Data Management) one.

These technical misconceptions unnecessarily complicate the understanding of the architecture, particularly when the user needs at the end to inspect the implementation code in order to grasp the overall working structure.

Events: The main drawback The main confusing and limiting factor behind understanding Moodle design resides in that it is described as an Event-Based architecture. The fact of using a class named Event for just logging system activities does not make it an Event System and so it is far from take full advantage of the benefits of such style from the flexibility point of view.

The name Event-based derives from the fact that it stores information in the database almost all the external or internal ones interactions produced into the system which are considered events. This concept is slightly different, but essentially important, when architectural interactions are involved; Event-based does not mean Event-Driven at all, but is very easy for non-aware researchers misinterpret this difference and then start to fruitlessly seek inexistent dynamic interactions triggered by such events.

In an Event-Driven Architecture events are first class active entities which trigger customizable behaviors depending on each specific variations of the expected behavior. This built-in capability would offer a powerful tool for enabling mechanisms to observe the behavior of users of a computer system at different levels of granularity.

Providing Moodle with this feature it is impossible without a thorough redesign and implementation. So, the possibilities of using it as an active assistant get complex and almost unattainable. For example, to observe the navigation behavior within a course that involves several activities would be necessary to modify all the plug-ins implementing each activity. In our case it took almost 1 week of code inspection just to add a menu option.

Thus, the only logical alternative left is to use Moodle as a data collector for further analysis through data and text mining techniques on its database. This alternative, however, does not come easy without a considerable effort too. Much of the referential integrity rules and object-relational schema information is wired into the Core and it is

necessary to recover it through code inspection. This fact adds another cumbersome error prone and time consuming work,

3.2.2 Redefining inspire

The Inspire project aims at to extend Moodle with machine-learning techniques in order to provide insights into student's behavior. Inspire was a framework under development,² that provide an abstraction of a common machine-learning classification systems, plus abstractions related to courses.

As being implemented in PHP it also gets difficult to understand it, even more because its design does not follow standard quality design concepts. In principle it is not necessary to understand Moodle architecture in the very detail, but a new challenge is to understand Inspire. At the time we try reusing it, it came accompanied by documentation somewhat detailed, but informal and mostly based on textual descriptions. So code scavenging was the only way in order to try to maximize code reuse, minimizing paradoxically the potential that a framework is supposed to provide.

Figure 1 shows the recovered abstract flow chart that involves the main components, depicted over the provided UML class diagram.

The Model class represents a combination of indicators and objectives used for prediction.³ Once the Model component is defined, the Analyzer component starts the data collection process, taking into account the Target component and the time range defined in the Time Splitting model. Once the samples have been generated, the Indicator component is responsible for carrying out the calculations corresponding to its definition, and then being joined by the Dataset_Manager, which provides the necessary Dataset for the prediction processors (through Processor) to generate the model predictions.

Discovering this control path was a time consuming routine task. For the development of Middle, the analysis of learning profiles of Felder's vision through Bayesian networks need to be considered, and this show some other design limitations of Inspire.

Middle must provide flexibility by allowing the configuration of the dimensions of the learning styles, adding or removing indicators, specifying how the dimension of the profile of the trainees is analyzed. This should provide great adaptability to different types of courses and enable to test a variety of combinations of indicators with which evaluate the chosen dimension, be it "perception", "input", "processing" or "understanding", following the guidelines of Felder learning styles, or any other definition of learning styles or student models.

A problem arises with the definition of the Inspire TimeSplitter abstract component. Its name leads to a misunderstanding, because it indeed represents simple time or date ranges, splitting the time is not its real function. So the provided interface is confusing. The same problem arises with the naming of other abstractions, as Processor (too much generic because it does not process anything, it is just a hook defining a not too much clear protocol). These aspects among others, as a too much static control-flow, and

² Nowadays it has been added to the Moodle Core and is called Analytics.

³ In our case, we don't "predict" a Learning Style but we "infer it" based on student's interactions observation.

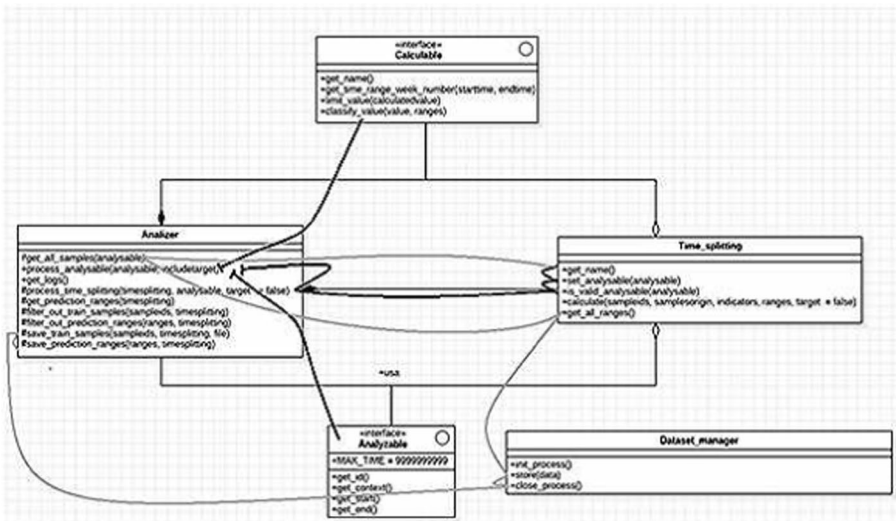


Fig. 1 Abstract component-connector view of Inspire deduced through code inspection

meaningless hook methods, made that the real reuse of Inspire was reduced to simply avoid the necessary burden to define a Moodle plug-in due to time frame limitations.

3.3 The middle architecture

In this context, Middle define a general dimension that uses the configuration of the rest of the dimensions to show a general learning profile of the learners based on the four dimensions. This component can be specialized to test other student models.

In this way a different abstraction is created by groping functionality of three Inspire components to maintain the general protocol of the framework. So, a component named Learning Style that defines the models of learning styles appears. A Style Dimension component that specifies how the different models are configured based on the objectives, time ranges and the definition of samples, a Learning Dimension component that represents the information that is taken from the system through indicators, created based on the characteristic behaviors of the different dimensions of learning, and a Bayesian Classifier component that allows predicting through a Bayesian network the type of style that a particular learner can have (See Fig. 2).

The value presented by the dimension chosen for each apprentice will be derived from the set of defined *indicators*. The value returned by each indicator will be related to some characteristic that influences the corresponding dimension, analyzing the data obtained based on Felder’s learning styles.

The components of Middle’s architecture are detailed below. It must be noticed the inheritance paths of base component (bracketed names in Fig. 2) are maintained for operational issues, but the real implementation overrides almost the 60% of the predefined implementation by Inspire, forcing in some cases their semantics in order to maintain the protocol.

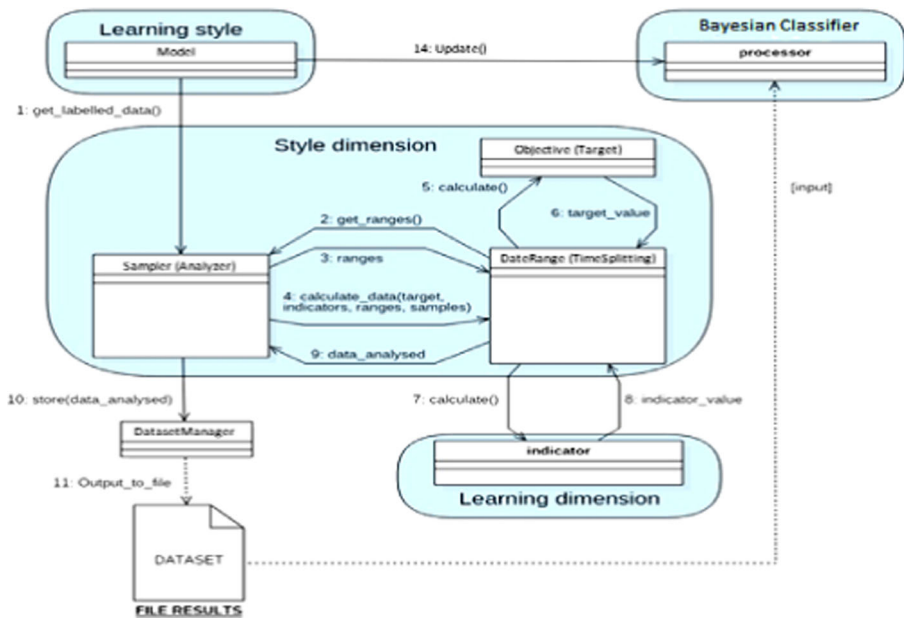


Fig. 2 Abstract Middle components diagram

3.3.1 Sampler component

The Sampler component is responsible for creating dataset files that will be used by the classifier processors. The base component maintains the responsibility for most of the basic functions. However, the definition of an abstract functionality that is key, *get_all_samples()*, which specifies what a sample is stands out. A sample can be any Moodle entity: a course, a user, an inscription, an attempt at a quiz, etc. The samples do not represent anything by themselves; they are just a list of identifiers. They make sense when combined with the Objective and Learning Dimension components. This is another issue the Inspire framework user has to deal with, a major one indeed.

Another issue is scalability since the amount of calculations to be performed could be large. So, this makes its execution be made at the Course level and then, the resulting datasets are merged once all the courses on the site have been sampled. This leads to introduce further modifications beyond the Inspire level. Again, if Moodle would had a true Event Driven Architecture providing a rich set of events this issue would be easily solved by using, for example, a Chain of Responsibility pattern to decide the granularity of re-calculation depending on the event type. And, going towards a more complex and elegant solution, it could combine this with a Strategy pattern to dynamically select the policy to be taken under different system load circumstances as the simplest reusable design solutions (Gamma et al. 1995).

3.3.2 Objective component

This component defines the semantics of the data analysis wanted, and also defines the actions to be carried out depending on such analysis. The objectives depend on Sampler

instances because they provide the samples that are needed. Samplers are entities other than Objectives because they can be used in several objectives. Each objective needs to specify which class of Sampler has to be used.

A callback defined by the objective will be executed once new classifications begin to arrive so each objective will have control over the inference results (implementing in this way an event system behavior).

For our application of learning styles, the definition of the objectives is based on the different dimensions of Felder's vision, extending the Target class with discrete values according to the corresponding dimension. For example, for the perception dimension, the values are: "high sensitive", "medium sensitive", "low sensitive", "low intuitive", "medium intuitive" and "high intuitive".

3.3.3 Learning dimension component

The responsibility of these indicators is simple. They make a calculation from a sample provided by Indicators specifying the data set, which it is necessary for the calculation.

Middle defines a set of potential indicators based on characteristic behaviors defined in the different dimensions of Felder. This does not imply that an indicator related to the Perception dimension cannot be related to another dimension.

3.3.4 Bayesian classifier

A Bayesian Classifier is an inference processor, adapting the Processor class of Inspire. Basically, is a machine learning back-end that process the datasets generated by the calculated indicators and objectives.

Communication between inference processors and Moodle is through files, because these processors can be encoded in PHP, Python, or other languages or even in cloud services.

This is perhaps one of the most limiting factors from the perspective of a flexible design because it would be simpler to establish a common protocol following the Command pattern or perhaps a Strategy one, among other possibilities.

4 Learning profile analysis model

Personalizing education is the motivation for the creation of the Middle architecture presented in this article. Thus, knowing each student is an important part of it.

The design driver of this component was emulating the activity of a teacher by observing and interacting with a student and learning how he/she learns. Each student's behavior is recorded and analyzed in order to deduce his learning style.

The first link in this chain is the student, when interacting with the course in which he is participating. This interaction happens while reading texts of the different modules of the course, carrying out proposed exercises, sending messages to their classmates, among other activities. Each time the student performs any of these actions, data is generated that is stored in a database in the form of activity logs (also known as logs).

As time goes by and students continue to participate in a course, the volume of these records increases. Then, once the database has a minimum amount of data (for example, when a course or section ends), it is acceptable to start using the indicators.

The indicators provide a measure of specific student interactions with an online course. For this, the indicators query the basic actions of the students on the courses. These queries return a number that represents a number of interactions, or the timing of those interactions, or some analysis of characteristics of those interactions. The output of the indicators on the other hand will always be following the same standard, a number between 0 and 1.

4.1 Classification of didactic resources

In order to analyze the behavior of the students, it is necessary to know the type of teaching resources that the teacher himself poses. Moodle doesn't define an inclusive identifier for the different aspects a teacher can gather using alternative teaching resources. So, to cover different learning styles, it is necessary to identify characteristics of each proposal. For them, the teacher have to declare features such as which topic he/she is specialized in, what type of resource is using (theoretical approach, exercises or general information), how is the presentation of such resources (referring to whether it is textual, auditory, multimedia, or other).

In relation to sections and topics of a resource, it should be borne in mind that several resources may be in the same section but include different topics, or vice versa, may comprise the same topic and belong to different sections.

With this custom classification provided, it is not only assured to have a unique identifier per resource, but also to take advantage of the identifier to obtain enriched information from it.

4.2 Specification of dimensions and indicators

A short description of the four dimensions with the most important indicators for each one is presented below. This presentation highlights the main limitation discussed above. The lack of an adequate API interface makes necessary to directly access the raw Moodle database in order to get the needed indicators. This aspect obliges to understand a rather complex data schema that does not follow conventional restrictions of normalization (i.e. foreign keys). Besides, any change made on new versions can make that the whole application does not work properly or not function at all. For illustration purposes, two brief examples of the SQL implementation for processing indicators are shown further below.

4.2.1 Perception dimension

What kind of information do students preferably perceive?

In his writings Carl Jung introduced the idea that by mean of the senses or intuition people tend to perceive the world (Alvarez Gómez et al. 2006). The sensitive one involves observation and obtaining data through the senses. The intuitive one realizes the perception indirectly through subconscious speculations, imagination and forebod-ings. Although we all use both types, most people tend to use one style over the other.

Then, a sensitive student prefers to learn with facts, they are more patient, careful and good at memorizing, often likes to solve problems with well-established methods, without complications or surprises. The intuitive one prefers the conceptual and theoretical to the concrete, feels more comfortable with abstractions and mathematical formulations, they like innovation, work fast and dislike repetition.

- View actions Indicator

This indicator captures the readings made by each user in the courses he took, taking into account the proportion of teaching resources he used for each module. A sensitive student tends to interact with most resources, while an intuitive student tends to use few resources. In other words, this indicator informs the degree of intuitiveness or sensitivity that each user has based on the amount of readings they have in their activity log. A user who records a large number of readings is associated with a sensitive profile, because he needs to use a greater amount of resources, in order to obtain enough facts to build a concept according to his learning style. The intuitive one, on the other hand, given that he prefers more conceptual and theoretical material to those specific, is associated more with a user with a small amount of resource readings.

It is important to define the context that makes sense to use this indicator. The courses must allow the exploration of the different teaching resources without restrictions or obligations. This does not imply that the course does not have a structure, but that the student can read the material in the order they want, according to their learning style.

In order for the indicator to perform the necessary calculations and obtain a result in relation to the number of views a student has, it is required to obtain the following values:

- Maximum amount of resources used in a course. To perform this query, the record of all view or review actions grouped by user is taken. Then, the number of actions per user is calculated and the maximum of them is taken.

```
SELECT MAX(amount) AS maximum
FROM (SELECT userid AS userid, U.firstname AS names, U.lastname AS lastName, COUNT(*) AS amount
      FROM (SELECT L.userid as userid, L.objecttable AS objecttable ,
      L.objectid AS objectid FROM mdl_logstore_standard_log L
      WHERE (L.action = 'viewed' OR L.action = 'reviewed')
      AND L.contextlevel = 70AND L.objecttable IN (
      SELECT name FROM mdl_modules)
      AND L.courseid = ?
      GROUP BY L.userid, L.objecttable,
      L.objectid
      ORDER BY l.userid) AS list_mod_users
INNER JOIN mdl_user U ON (userid = U.id)
GROUP BY userid) AS list_view_modules
```

- Amount of resources used by a student in a course. The amount of activity records of the type “view” or “review” is calculated for each user.

```

SELECT userid AS userid, U.firstname AS names, U.lastname AS lastnames, COUNT(*) AS amount
  FROM (SELECT L.userid as userid, L.objecttable AS objecttable , L.objectid AS objectid
        FROM mdl_logstore_standard_log L
        WHERE (l.action = 'viewed' OR l.action = 'reviewed')
        AND l.contextlevel = 70
        AND l.objecttable IN (SELECT name FROM mdl_modules)
        AND l.courseid = ?
        AND l.userid = ?
       )
 GROUP BY L.userid, L.objecttable, L.objectid
 ORDER BY l.userid) AS list_mod_users
 INNER JOIN mdl_user U ON (userid = U.id) GROUP BY userid

```

Through these values, a value can be set between [0–1], which establishes the degree of “view” events that a student had in relation to the maximum number of “view” events that a user with a “student” role had in that course.

- Exercise actions indicator

It captures the amount of actions on exercises in which each student participates, whether creating, editing, deleting, etc. Based on the classification of the proposed resources, the number of exercises in which the student participated is calculated. This implies any completion or delivery of the course exercises. A student who has more interaction with resources classified as exercises reflects a sensitive student’s behavior. In turn, a student who does not register a lot of interaction activity with the exercise proposed by the teacher reflects a more similar behavior to the intuitive student.

This indicator returns a value between [0–1] that represents the degree of participation that a student has with respect to the resources classified as exercises, comparing it with the one that had the greatest participation. A low value of this indicator tends to an intuitive student profile.

- Time evaluation indicator

This indicator determines the average resolution time of a quiz for a given student in a course. A sensitive learner tends to be more cautious in solving an evaluation, taking more time in this process. An intuitive apprentice tends to solve the evaluation in less time in his eagerness to work fast, taking less time in reviewing his answers.

For this, the average time in minutes that a given apprentice takes to solve a questionnaire and the maximum average time of resolution of that questionnaire considering all the students is consulted:

- Maximum average resolution time for an evaluation in the course.
- Average resolution time for an apprenticeship evaluation in a course.

Taking the averages of the questionnaire resolution times, a value can be obtained between [0–1] related to the time a student takes in the resolution of a questionnaire compared to the maximum time it took for a user in the same course.

4.2.2 Processing dimension

How does the apprentice prefer to process the information?

According to Kolb (2014), the mental process through which perceived information is converted into knowledge can be divided into two categories: active experimentation and reflexive observation. Active experimentation involves doing something in the external world from information. Discuss it, explain it or try it in some way. Reflective observation involves examining and manipulating information in an introspective way.

Active learners tend to work better when they can experience and manipulate things, that is, they process information better if they can do something active with it. So they also tend to work better in groups. Instead, reflexive learners prefer to think things through before acting and working alone or with another person at most.

- Interaction message indicator

It captures the amount of actions related to the start of an interaction. Any action of creation or modification of any object related to a forum or a chat is considered.

In this case, it is necessary calculate: 1) maximum amount of chat and forum interactions in a course; 2) number of chat or forum interactions of a student in a course. This indicator informs about the degree of initiative that an apprentice has to participate in activities that involve group work and collaborative work, based on a value between [0–1].

4.2.3 Input dimension

Through which sensory modality is the information more effectively perceived?

The visual student prefers to see what he is learning through graphs, diagrams, images, timelines and videos. The verbal learner is most successful when he hears or reads the information. When the student “reads”, he sees written words, but the brain generally translates them into their spoken equivalents and processes them very differently from how they process truly visual information.

The indicators proposed for this dimension make use of the classification proposed above to be able to recognize which activities are textual, auditory or multimedia. Thus, the variation between them is given by the field that establishes that classification.

- Only text activities, only hearing activities, only multimedia activities indicators

These three indicators represent the amount of activities in which the apprentice intervenes, which only contain text, or only audio, or multimedia content. Based on the classification of the proposed activities, the interaction of the learner is analyzed.

4.2.4 Understanding dimension

How does the student progress in his learning?

The sequential student prefers to have the information in a linear and orderly manner. The global learner prefers to see the whole picture first, tends to learn in big jumps, absorbing the material almost randomly.

- Reading jumps indicator

This indicator tracks the reading order that a student records about the course activities. For this, the amount of “jumps”, which a student performs in reading different activities, is taken into account. Based on the record of readings on sections presented by an apprentice; If the distance of a section and the immediately preceding section in the record is greater than 1, then a section break is counted.

It is normal, in training areas, to present the content in a logically orderly progression, where the pace of learning is determined by the program or calendar. When a unit is covered, an exam is applied to the trainees and passed to the next unit. Many apprentices are accustomed to this way of learning sequentially, dominating the contents in the order in which they are presented. Others, however, have difficulty learning that way, they learn by blocks. Sequential learners follow linear reasoning for problem solving, while global learners make intuitive advances and may not know how to explain how they arrived at the solution.

Traditional schools have often difficult experiences with global students. Since these students do not learn continuously and predictably, they tend to feel out of the progress of their classmates and then unable to meet the expectations of their teachers. Some eventually feel discouraged with training and resign. However, global learners are excellent for synthesis, multidisciplinary researchers, and systemic thinkers who manage to see connections and relationships where no one else sees them (Felder and Silverman 1988). We give these students a chance, detecting their particularities in a way that makes traditional education easier.

5 Bayesian network instantiation process

In order to build the Bayesian network, it is first necessary to define its structure and establish what variables are involved and how they interact with each other. The resulting Bayesian network consists, on one hand, of a set of independent variables called indicators that provide information on the level of probability of some trainee’s characteristic, based on the events recorded in the Moodle platform. On the other hand, there are the dependent variables that represent the tendencies of the students to different dimensions of Felder’s personality.

The probability that a student’s preference is classified as intuitive or sensitive, in the case of the perception dimension, will depend on the observed behavior of the student, measured through the proposed indicators that influence that dimension. In this way, there is a dependence on the perception dimension with respect to the group of corresponding indicators, as can be seen in Fig. 3.

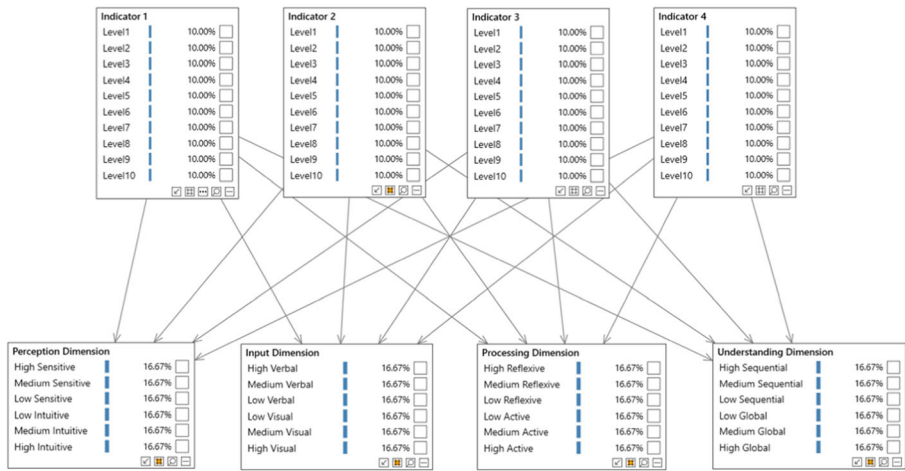


Fig. 3 Middle Bayes Network

To approximate this probability, you must first experiment with a group of students so that at the end, based on their behavior, their trends in the different dimensions of the Felder model are known.

$$P(\text{Perc} = HI \mid \text{Ind}_1, \dots, \text{Ind}_i)$$

For example, if we want to know what is the probability a student has a tendency to be High Intuitive (HI) in the perception dimension, it is necessary to calculate the probability that it is HI in the perception variable, given the experience recorded in the different indicators influencing this dimension.

Once the structure of the Bayesian network is defined, the conditional probability tables should be established (see Fig. 4). To meet this goal, the data collected from the Moodle database is used.

View Actions		Exercise Actions		Time Evaluation	
Level	P(Level)	Level	P(Level)	Level	P(Level)
1	P(1)	1	P(1)	1	P(1)
...
10	10	10	10	10	10

Indicators			Perception Dimension					
View Actions (VA)	Exercise Actions (EA)	Time Evaluation (TA)	Intuitive			Sensitive		
			Low (LI)	Medium (MI)	High (HI)	Low (LS)	Medium (MS)	High (HS)
1	1	1	$P(\text{Perc} = LI \mid VA = 1, EA = 1, TE = 1)$	$P(\text{Perc} = MI \mid VA = 1, EA = 1, TE = 1)$	$P(\text{Perc} = HI \mid VA = 1, EA = 1, TE = 1)$	$P(\text{Perc} = LS \mid VA = 1, EA = 1, TE = 1)$	$P(\text{Perc} = MS \mid VA = 1, EA = 1, TE = 1)$	$P(\text{Perc} = HS \mid VA = 1, EA = 1, TE = 1)$
1	1	2	$P(\text{Perc} = LI \mid VA = 1, EA = 1, TE = 2)$	$P(\text{Perc} = MI \mid VA = 1, EA = 1, TE = 2)$	$P(\text{Perc} = HI \mid VA = 1, EA = 1, TE = 2)$	$P(\text{Perc} = LS \mid VA = 1, EA = 1, TE = 2)$	$P(\text{Perc} = MS \mid VA = 1, EA = 1, TE = 2)$	$P(\text{Perc} = HS \mid VA = 1, EA = 1, TE = 2)$
2	2	2	$P(\text{Perc} = LI \mid VA = 2, EA = 2, TE = 2)$	$P(\text{Perc} = MI \mid VA = 2, EA = 2, TE = 2)$	$P(\text{Perc} = HI \mid VA = 2, EA = 2, TE = 2)$	$P(\text{Perc} = LS \mid VA = 2, EA = 2, TE = 2)$	$P(\text{Perc} = MS \mid VA = 2, EA = 2, TE = 2)$	$P(\text{Perc} = HS \mid VA = 2, EA = 2, TE = 2)$
...	$P(\text{Perc} = LI \mid VA = \dots, EA = \dots, TE = \dots)$	$P(\text{Perc} = MI \mid VA = \dots, EA = \dots, TE = \dots)$	$P(\text{Perc} = HI \mid VA = \dots, EA = \dots, TE = \dots)$	$P(\text{Perc} = LS \mid VA = \dots, EA = \dots, TE = \dots)$	$P(\text{Perc} = MS \mid VA = \dots, EA = \dots, TE = \dots)$	$P(\text{Perc} = HS \mid VA = \dots, EA = \dots, TE = \dots)$
10	10	10	$P(\text{Perc} = LI \mid VA = 10, EA = 10, TE = 10)$	$P(\text{Perc} = MI \mid VA = 10, EA = 10, TE = 10)$	$P(\text{Perc} = HI \mid VA = 10, EA = 10, TE = 10)$	$P(\text{Perc} = LS \mid VA = 10, EA = 10, TE = 10)$	$P(\text{Perc} = MS \mid VA = 10, EA = 10, TE = 10)$	$P(\text{Perc} = HS \mid VA = 10, EA = 10, TE = 10)$

Fig. 4 Conditional Probability Tables, Middle

6 Experimental results

To test the hypothesis proposed, controlled experiments were carried out to analyze the courses taken by the trainees using the platform. In this chapter we present the corresponding statistical analysis obtained. For this, a form based on the ILS was designed that collects information about the trainees and allows them to analyze their preferences regarding Felder's learning styles.

The form was completed by 45 university students from the computer science area. Figure 5 shows the distribution of student characteristics according to Felder's ILS.

Analyzing Fig. 6, it is clear that there is a bias in all the Felder's four dimensions. Through the normal distribution of the ILS results, an estimate of the characteristic profile of the students is obtained.

One of the first conclusions that can be drawn is that students have a dominant preference to be sensitive medium-low, are more active than reflective, with a medium tendency, have a clear bias to the visual and a neutral profile with respect to the form and the order in which they understand, that is, they do not show a marked bias towards sequential or global dimension.

These results can be considered reliable estimators of student profiles, since they are not only based on Felder's ILS, supported by several authors (Caro et al. 2015; Di Bernardo and Del 2005; Kazu 2009; Pestana and Mart 2012; Rocha and Garzuzi 2015; Zywno 2003), but they show similar results to other well-recognized publications on the analysis of student learning styles of engineering (Feldman et al. 2015; García et al. 2005, 2007; García et al. 2008; Puello et al. 2014).

For the test of the Middle tool, an experiment was carried out through the importation of a virtual learning platform course. It should be clarified that the name of the trainees is omitted to maintain the privacy of the course participants.

Table 1 shows the distribution of probabilities (from level 1 to level 10) of different indicators of the perception dimension, based on the recorded behavior of course participants.

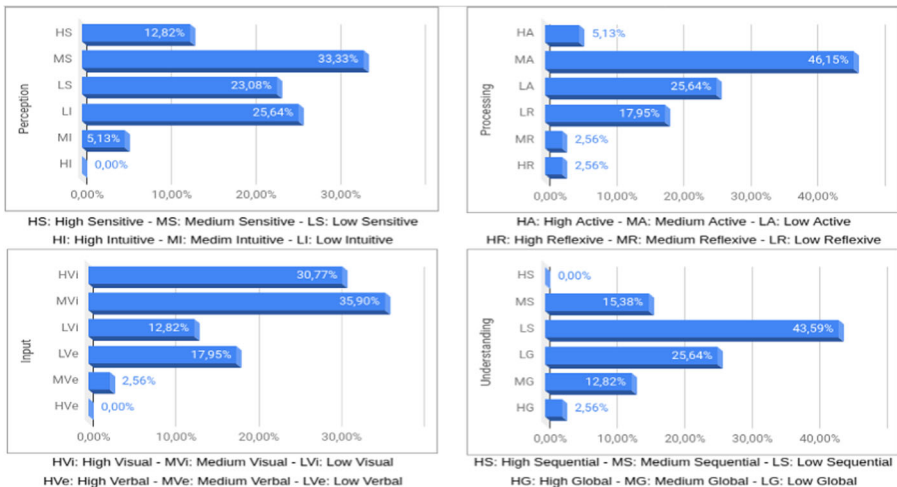


Fig. 5 Distribution of the characteristics of the learners according to Felder ILS

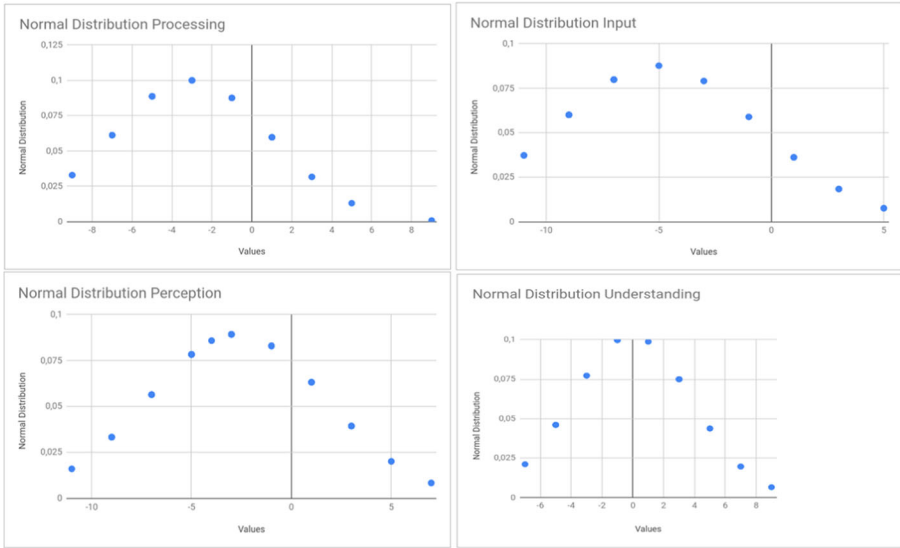


Fig. 6 Normal distribution of ILS results

Analyzing the perception dimension, considering the View Actions, Exercise Actions and Time Evaluation indicators, the high, medium or low tendency of each student is observed, to have a consistent indication of their preference, either sensitive or intuitive, as shown in Table 2.

In the case of the student 1, it is observed that it presents a very low level of average sight actions per week, low amount of actions related to course exercises and average times of resolution of very low questionnaires, being classified with a high tendency towards intuitiveness. This not only provides the teacher with the profile of this student, but also feeds the Bayes network, increasing the probability that this student is very intuitive, given that the View Actions, Exercise Actions and Time Evaluation indicators have levels N1, N2 and N1 respectively.

Table 1 Distribution of the probability level of indicators, dimension perception

Level/Indicator	View Actions	Exercise Actions	Time Evaluation
L1	0,15	0,08	0,15
L2	0,08	0,23	0,08
L3	0,08	0,00	0,08
L4	0,31	0,08	0,23
L5	0,00	0,31	0,15
L6	0,08	0,08	0,15
L7	0,23	0,08	0,00
L8	0,00	0,08	0,08
L9	0,00	0,08	0,08
L10	0,08	0,00	0,00

Table 2 Student trend, perception dimension

Name	Perception Dimension	Indicators		
		View Actions	Exercise Actions	Time Evaluation
Learner 1	High Intuitive	L1	L2	L1
Learner 2	Medium Sensitive	L7	L8	L8
Learner 3	High Intuitive	L1	L1	L2
Learner 4	Low Intuitive	L4	L5	L4
Learner 5	Medium Sensitive	L7	L7	L6
Learner 6	Low Sensitive	L6	L5	L5
Learner 7	High Intuitive	L2	L2	L1
Learner 8	Low Intuitive	L4	L5	L5
Learner 9	Low Intuitive	L4	L5	L4
Learner 10	Medium Sensitive	L7	L6	L6
Learner 11	Medium Intuitive	L3	L4	L3
Learner 12	Low Intuitive	L4	L2	L4
Learner 13	High Sensitive	L10	L9	L9

Finally, Table 3 shows the distribution of probabilities of the dimensions, the result of the analysis of the behavior of the participants of the course mentioned above.

These results, as already stated, are compatible with other studies that allow us to affirm that the autonomous detection mechanism developed is a contribution to the autonomous generation of personalized courses. In short, the result of what is usually done manually through the test has been achieved. Moreover, the possibility of enhancing this result by tuning it using MBTI scores associated to each dimension as complement brings an additional tool to refine neutral values that usually appears when projected to the whole analyzed group due to the test nature⁴.

7 Related work

The idea of applying machine learning techniques to automatically detect students Learning Styles has had a growing interest particularly in the last decade. The many approaches existing in literature are well described in several articles (Zine et al. 2019; Pashler et al. 2008; Feldman et al. 2015) each one providing inclusive characterization and classification approaches.

Most of the cited works are oriented automatically discover student Learning Styles, and some of them use in different forms the Moodle platform mostly relying on log-mining and alternative machine-learning methods (Kika et al. 2019; Abdullah 2015; Surjono 2014; Karagiannis and Satratzemi 2018). These articles describe interesting approach, but do not provide any insight about the adequacy of Moodle to support the proposed extensions.

⁴ This aspect, under research, is beyond the scope of this paper

Table 3 Distribution of dimension probabilities

Probability distribution of dimensions							
perception	Input		Processing		Understanding		
High Sensitive	0,077	High verbal	0,077	High Reflexive	0,077	High Sequential	0,308
Medium Sensitive	0,231	Medium Verbal	0,308	Medium Reflexive	0,231	Medium Sequential	0,077
Low Sensitive	0,154	Low Verbal	0,231	Low Reflexive	0,308	Low Sequential	0,154
Low Intuitive	0,3308	Low Visual	0,000	Low Active	0,231	Low Global	0,154
Medium Intuitive	0,077	Medium Visual	0,231	Medium Active	0,154	Medium Global	0,077
High Intuitive	0,154	High Visual	0,154	High Active	0,000	High Global	0,231

Middle makes use of log-mining but due to the limitations described, leaving open the possibility of specialize the solution in a more advanced version of Model's core that treat events as such in a flexible way.

8 Conclusions

This paper presents Middle a contribution to the autonomous recognition of learning styles of learners, based on the analysis of their behavior in Moodle, through a reformulation of the Inspire framework, which served more as an implementation hook than a reuse infrastructure.

The extension of Moodle to accommodate the Bayesian inference engine resulted more complex than what was supposed to be, generating tradeoffs that had to be solved in the most elegant architectural way that the time-frame of the project allowed.

The results of using Middle in the context of real courses on mathematics and programming were very satisfactory giving a final approximation of more than 95% of the ones obtained trough manual tests. Added to this, the flexibility to change different supports for data gathering and inference techniques in a seamless way make us feel that the project was successful beyond the initial expectations.

Despite the flaws that we describe regarding some design decision made respect to a more disciplined architecture design, we think that Moodle is an excellent vehicle to support an efficient and productive e-learning. Perhaps with some extra effort rethinking some essential architectural mechanisms it will become in the most robust open-source available platform supporting intelligent behaviors for personalized teaching.

Code availability Under request for research purposes.

Authors' contributions Analia Amandi and Marcelo Campo are Main Researchers and Directors of the project.

Julio Cesar Biset is a PhD candidate at UNICEN University, Leader of the development group.

Funding This work is partially supported by d the CICPBA (Comisión de Investigaciones Científicas de la Provincia de Buenos Aires).

Compliance with ethical standards

Conflicts of interest/competing interests N/A

References

- Abdullah, M. A. (2015). Learning style classification based on student's behavior in Moodle learning management system. *Transaction on Machine Learning and Artificial Intelligence*, 3(1), 28.
- Alvarez Gómez, M., Limón, H. O., Montemayor, P. V. M., & Alvarez Lavín, A. (2006). Diagnóstico de estrategias y estilos de aprendizaje (p. 126). p. 126.
- Bass L., Clemments P., Katzman R. (2013). *Software Architecture & Practice*, (3rd edn), Addison Wesley.
- Bobadilla, J., Ortega, F., & Gutiérrez, H. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132.
- Campo, M., Pace, A. D., & Zito, M. (2002). Developing object-oriented enterprise quality frameworks using proto-frameworks. *Software: Practice and Experience*, 32, 837–843.
- Caro, M. F., Josyula, D. P., & Jiménez, J. A. (2015). Multi-level pedagogical model for the personalization of pedagogical strategies in intelligent tutoring systems. *Dyna*, 82(194), 185–193.
- Coffield, F., Moseley, D., Hall, E., & Ecclestone, K. (2004). *Learning styles and pedagogy in post-16 learning: a systematic and critical review (PDF)*. London: Learning and Skills Research Centre ISBN 1853389188.
- Coplien, J., Hoffmann, D., & Weiss, D. (1998). Commonality and Variability in Software Engineering. *IEEE Software*, 15, 6.
- Cuevas, J. (2020). Is learning styles-based instruction effective? A comprehensive analysis of recent research on learning styles. *Theory and Research in Education*. <https://doi.org/10.1177/1477878515606621> SAGE.
- Di Bernardo, G. J., & Del, M. (2005). Determining the learning styles of biochemistry students as an initial step in finding meaningful learning. *Scientific and Technological Communications*, 9. (In Spanish).
- Fayad, M., Schmidt D., and Johnson, R. (1999). *Implementing application frameworks: Object-oriented frameworks at work*. John Wiley & Sons, Inc., USA.
- Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education*, 78(7), 674–681.
- Felder, R. M., & Spurlin, J. (2005). Applications, reliability and validity of the index of learning styles. *International Journal of Engineering Education*, 21(1), 103–112.
- Feldman, J., Monteserin, A., & Amandi, A. (2014). Detecting students' perception style by using games. *Computers in Education*, 71, 14–22.
- Feldman, J., Monteserin, A., & Amandi, A. (2015). Automatic detection of learning styles: State of the art. *Artificial Intelligence Review*, 44(2), 157–186.
- Fleming, N. (1995). "I'm different; not dumb: modes of presentation (VARK) in the tertiary classroom" (PDF). Higher education: blending tradition and technology: proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA).
- Fowler, M. (2002). *Patterns of Enterprise application architecture*. Addison: Wesley.
- Gamma E., Helm R., Johnson R., and Vlissides J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison: Wesley Longman Publishing Co., Inc.
- García, P., Amandi, A., Schiaffino, S., & Campo, M. (2005). Using Bayesian networks to detect students' learning styles in a web-based education system. *Proc of ASAI*, Rosario, 115–126.
- García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education*, 49(3), 794–808.
- García, P., Schiaffino, S., & Amandi, A. (2008). An enhanced Bayesian model to detect students' learning styles in web-based courses. *Journal of Computer Assisted Learning*, 24(4), 305–315.
- Jensen, F. (1996). An introduction to Bayesian networks, *UCL Press*, 1996, pp 178.
- Karagiannis, I., & Satratzemi, M. (2018). An adaptive mechanism for Moodle based on automatic detection of learning styles. *Education and Information Technologies*, 23(3), 1331–1357.
- Kazu, I. Y. (2009). The effect of learning styles on education and the teaching process. *Journal of Social Sciences*, 5(2), 85–94.
- Kika A., Leka L., Maxhelaku S., and Ktona A. (2019). "Using data mining techniques on Moodle data for classification of student? S learning styles." In: *Proceedings of International Academic Conferences*, no. 9211567.
- Kirschner, P. A. (2017). Stop propagating the learning styles myth. *Computers & Education*, 106, 166–171.

- Knoll, A. R., Otani, H., Skeel, R. L., & Van Horn, K. R. (2017). Learning style, judgements of learning, and learning of verbal and visual information. *British Journal of Psychology*, *108*(3), 544–563.
- Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT Press.
- Lilienfeld, S., Lynn, S., Ruscio, J., Beyerstein, B. (2010). "Myth #18: Students learn best when teaching styles are matched to their learning styles". *50 great myths of popular psychology: shattering widespread misconceptions about human behavior*. Chichester: Wiley-Blackwell.
- Pashler, H., McDaniel, M., Rohrer, D., & Bjork, R. (2008). Learning styles: Concepts and evidence. *Psychological Science in the Public Interest*, *9*(3), 105–119.
- Pestana, P. B., & Mart, J. (2012). Styles and strategies for health science students. *Psicogente*, *15*(28).
- Puello, P., Fernández, D., & Cabarcas, A. (2014). Herramienta para la Detección de Estilos de Aprendizaje en Estudiantes utilizando la Plataforma Moodle. *Formación Universitaria*, *7*(4), 15–24.
- Rocha, M. I. M., & Garzuzi, V. (2015). Estrategias de aprendizaje sugeridas a partir de estilos de aprendizaje identificado. *Revista de Orientación Educativa*, *55*, 69–86.
- Rodríguez G, Soria Á, and Campo M., (2013). Virtual scrum: A teaching aid to introduce undergraduate software engineering students to scrum. *Computer Applications in Engineering Education*. New York: JohnWiley & Sons, Inc.
- Rogowsky, B. A., Calhoun, B. M., & Tallal, P. (2020). Providing instruction based on students' learning style preferences does not improve learning. *Frontiers in Psychology*, *11*, 164.
- Rosati P. A., and Felder, R. M. (1995). Engineering student responses to an Index of Learning Styles, *Proc.1995 ASEE annual conference*, Washington, DC.
- Scott, E., Rodríguez, G., Soria, A., & Campo, M. (2014). Are learning styles useful indicators to discover how students use scrum for the first time? *Computers in Human Behavior*, *36*, 56–64.
- Scott E., Rodríguez G., Soria Á. and Campo M., (2016). Towards better Scrum learning using learning styles. *Journal of Systems and Software*, vol. 111, pp. 242-253, Elsevier Science, ISSN: 0164–1212
- Seery N., Gaughran W. F. and Waldmann, T. (2003). Multi-modal learning in engineering education, Proc. 2003 Annual ASEE Conf., ASEE 31. K. C. Dee, G. A. Livesay and E. A. Nauman, Learning styles of first- and second-year engineering students, *Proc. 2003 ASEE/WFEO International Coll.*, ASEE, 2003.
- Surjono, H. D. (2014). The evaluation of a moodle based adaptive e-learning system. *International Journal of Educational Technology*, *4*(1), 89.
- Witten, I., Eibe, F., Hall, M. and Pal, C. (2016). *Data mining: Practical machine learning tools and techniques* (4th edn). Burlington: Morgan Kaufmann, Elsevier.
- Wooldridge, M., Jennings, N. (1995). Intelligent Agents — Theories, Architectures, and Languages, *Lecture Notes in Artificial Intelligence. Volume 890*, Springer-Verlag
- Zine, O., Derouich, A., & Talbi, A. (2019). A comparative study of the Most influential learning styles used in adaptive educational environments. *International Journal of Advanced Computer Science and Applications*, *10*(11).
- Zywno, M. S. (2003)., A contribution of validation of score meaning for Felder-Soloman's Index of Learning Styles, *Proc. 2003 Annual ASEE conference*, ASEE.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Marcelo Campo^{1,2} · **Analia Amandi**^{1,2} · **Julio Cesar Biset**³

✉ Analia Amandi
amandi@exa.unicen.edu.ar

¹ NICE, Facultad de Ciencias Exactas, UNICEN National University, Buenos Aires, Tandil, Argentina

² CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas, Buenos Aires, Argentina

³ CICPBA, Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Buenos Aires, Argentina