systems

MDPI

Article

Adaptation in E-Learning Content Specifications with Dynamic Sharable Objects

Ignacio Gutiérrez [1], Víctor Álvarez [2,*], M. Puerto Paule [1], Juan Ramón Pérez-Pérez [1] and Sara de Freitas [2]

[1] Department of Computer Science, University of Oviedo, Campus de Llamaquique, Facultad de Ciencias, Oviedo 33007, Spain; nacho@ijgm.es (I.G.); paule@uniovi.es (M.P.P.); jrpp@uniovi.es (J.R.P.-P.)
[2] Office of the Pro-Vice Chancellor (Learning and Teaching), Murdoch University, Perth 6150, Australia
* Correspondence: V.Alvarez@murdoch.edu.au; Tel.: +61-8-9360-6965

Academic Editor: Jon Mason
Received: 24 February 2016; Accepted: 2 June 2016; Published: 8 June 2016

Abstract: Dynamic sophisticated real-time adaptation is not possible with current e-learning technologies. Our proposal is based on changing the approach for the development of e-learning systems using dynamic languages and including them in both platforms and learning content specifications thereby making them adaptive. We propose a Sharable Auto-Adaptive Learning Object (SALO), defined as an object that includes learning content and describes its own behaviour supported by dynamic languages. We describe an example implementation of SALO for the delivery and assessment of a web development course using Moodle rubrics. As a result, the learning objects can dynamically adapt their characteristics and behaviour in e-learning platforms.

Keywords: e-learning systems; dynamic languages; learning object; content specifications

1. Introduction

The first Internet-based learning systems appeared over two decades ago and, since then, universities worldwide have incorporated online educational resources to complement classroom teaching [1]. As a result of the supply and demand of online resources among educational institutions, there is a technological need for content interoperability that has motivated the development of reusable Learning Objects (LOs) and communication/collaboration services [2]. The concept of learning object has been supported by content specifications such as the Sharable Content Object Reference Model [3], IMS Learning Design [4] and IMS Common Cartridge [5]. Among these standards, SCORM is currently the most popular. A more recent specification, Tin Can Api [6] introduces some improvements over the previous, like the ability of tracking learners' progress. These specifications, or *de facto* standards, make use of formal languages to describe LOs as an ensemble of teaching/learning resources, including a hierarchy and navigation structure that students follow to acquire the expected knowledge and skills.

From a technological perspective, Learning Objects are based on the paradigm of object orientation, which allows the creation and aggregation of digital components (called objects) that can be reused in different contexts and in many ways. The LO specifications employ a model that allows serialising and storing objects, and thus LOs can be created and consumed by any tool that supports the specification.

The current LO models are not conducive to dynamic adaptation for users or for systems. From a computer science viewpoint, one of the technological limitations of current approaches is the inability to serialise 'own-behaviour' in such objects. This restriction prevents the possibility of adapting behaviour dynamically and targeting more personalised and contextual educational e-learning settings. The effects of enhancing personalised and contextual e-learning can be particularly significant when combined with emerging e-learning paradigms, such as mobile and ubiquitous learning [7], which occur in adaptable contexts, and thus benefit from real-time adaptation. In addition, students spend a significant amount of time searching for online information to acquire knowledge and skills. This task of selecting the most relevant information can be cognitively very demanding and cause students to stress and have a negative effect on students' engagement and attrition. One way to help students to engage with online learning is to make the systems sensitive to the needs and context of the user. Recent research shows that the effectiveness of adaptive learning increases when the systems introduce mechanisms allowing for learner control [8,9].

Traditional online educational systems have either none or statically implemented adaptation, such as the case of conditional activities and groups/roles customisation in Moodle [10]. In order to address this gap, this paper introduces the creation of learning objects that can dynamically adapt their behaviour in addition to being interoperable across platforms. We have designed and implemented what we call a SALO (Sharable self-Adaptive Learning Object) [11] that is fully autonomous and able to express its content and adaptive behaviour so that it can be shared between different Learning Management Systems (LMSs).

Creating SALOs adds a level of difficulty, as it requires technical knowledge, but in our view this can be compensated by advantages in re-using the object across different units. Under this approach the LO becomes completely independent and from a technical point of view we can speak of "classless" objects, each one unique and able to behave self-sufficiently. We achieve this goal by using dynamic languages that enable storing objects' behaviour in a simple manner. The flexibility of dynamic languages allows us to generate objects that are "free" from their classes, can experience changes and are dynamically adaptable to the context in which they run.

2. Background

2.1. Adaptive Educational Systems

It is crucial for teaching and learning processes that systems manage information according to the needs and context of each individual student. To this end, adaptive educational hypermedia (AEH) include the variables and a user model to adapt system navigation and content, goals, knowledge level, background, interests, preferences, stereotypes and cognitive preferences [12]. The user model can contain behavioural, cognitive and affective data [13]. Adaptive systems can also include an expert and domain models, indicating the content and knowledge to be taught and relationships between domain elements [14].

AEH relies on the use of methods and techniques of adaptive hypermedia [15] to include adaptation in educational systems. Approaches to achieve this goal include: using sets of rules, proprietary objects and closed formats that belong to the application and the introduction of new languages for the definition and creation of adaptive processes [16–18].

Among the previous approaches, the use of rules provides relatively simple and direct ways to achieve system adaptation; however, the resulting adaptation cannot be used by other systems. With regards to the creation of proprietary objects in each application, the absence of common formats prevents the exchange of such objects between different systems. The use of adaptive languages provides a step forward in this area. The level of expression of these languages is high, but it is also strongly dependent on static user types. The system does not have the ability to add new variables during execution time, nor is it capable of expressing complex behaviour. Another drawback of using proprietary languages consists of forcing Unit Coordinators to learn it, as well as e-learning systems to interpret it. In this situation, it would be preferable to use a well-established language and eliminate the additional complexity of learning and interpreting a new language.

2.2. Web-Based Online Learning Platforms

The term e-learning system encompasses software applications such as Learning Management Systems (LMS), Learning Content Management Systems (LCMS) and Virtual Learning Environments (VLE). In general, a Web-based e-learning platform is a software application that relies on the Web to give support to online teaching and learning processes [1]. Online learning platforms and the newest Massive Open Online Courses (MOOCs) offer an integrated software environment for management and exchange of educational experiences. This includes tools for creation, maintenance and distribution of online courses and facilities for student enrolment, maintenance and reporting, as well as more specific tasks for system administration.

In terms of software architecture, LMSs have evolved from "black-box" and monolithic design to service-oriented architectures (SOA) [1]. E-learning frameworks give support to LMS development and are based on SOA architectures. These frameworks include lists of service descriptors and layers of components, development guidelines and, in some cases they even provide the implementation for some of the services. The main idea behind this type of architecture is to promote interoperability among the different components and between platforms, as well as obtain and incorporate external service providers (e.g., Google services). Content interoperability is achieved through well-defined specification APIs, such as the Runtime Environment (RTE) for SCORM. The implementation of these APIs is conditioned by the underlying software architecture. In learning platforms with a modular architecture like Moodle, this development is integrated into a single module, whereas in service-oriented LMSs, such as Sakai ([19]), the development is based on services that may be independent from the learning platform.

2.3. Content Adaptation

In general, e-learning specifications adapt the courses using navigation and content sequencing. The SN (Sequencing and Navigation) module, that is common in most specifications, defines a set of rules that follows the information and preferences stored in a user model to allow the LMS to display certain content and provide a personalised navigation structure to the course. The sequence is adapted by a static algorithm that triggers a set of rules, and operates on the basis of the interaction of the student with the virtual learning environment. To take a technical example, each time a student achieves a learning objective (*i.e.*, the result of a test) then a process is launched to show the content that follows in the sequence. This way of processing learning objects as static containers of information, leaves them unable to enact functions with their associated content. Instead, the LMS acts as the external engine that handles the LOs.

In our view, a promising alternative consists in providing LOs with 'self-decision' in such a manner that they are invested with dynamic decision-making capabilities. This would enable self-control over display, adaptation to user actions and allow LMSs to focus on other functionality.

2.4. Dynamic Languages

The characteristics of dynamic languages make it possible to know and change the structure, behaviour and environment of an object dynamically [20]. It is important to note that these characteristics are shared also by scripting languages. Both terms can be used interchangeably, but more strictly a dynamic language is an evolution of scripting languages that incorporates a number of technical improvements, such as modularity, object orientation and access to databases.

The main feature of dynamic languages is to allow dynamic modelling, which is a requirement for some applications, such as those that are highly dependent on their context. Dynamic modelling allows changing the very nature of software and its usage. The ability of these languages to modify the internal code while running makes it possible to change data and behaviour dynamically. This feature may allow the user model to alter in real-time, both for LMSs and LOs, in order to provide highly adapted and personalised learning environments.

3. Our Proposed Solution: Shareable Auto-Adaptive Learning Object (SALO)

We define a Shareable Auto-Adaptive Learning Object (SALO) as a learning object that may describe and adapt its own behaviour dynamically, be independent of the running platform, context-customisable and reusable by other e-learning systems. Additionally, we believe a SALO must have the following characteristics [11]:

1. Provide content based on user needs and context.
2. Incorporate a context-aware behaviour.
3. Be reusable and independent of changes in the learning platforms.
4. Be interchangeable, shareable and compatible with current content specifications and platforms.

In our view, adding dynamic capabilities to learning objects clearly suggests the use of dynamic languages for systems development. The first three features require the theoretical foundations for reflective and introspective capabilities currently offered by dynamic languages. These features imply changes in the current design and development of modular e-learning platforms. However, such changes are compatible with existing development frameworks, as they define a set of services that are independent of the programming language.

With the goal of achieving interoperability between systems, our first proposed version of SALO contains dynamic data and behaviour packaged using the SCORM specification. In this manner, SALOs can be used in SCORM compliant environments, which currently include a large number of learning and training management platforms. In its current form, SCORM describes only the packaging of the content of a learning object, whereas the behaviour is indicated as metadata which is in turn processed using the engine specification defined in the module SN (Sequencing and Navigation) and implemented in the specific learning platform. This approach for adaptive LOs limits the scalability of the solution, as it becomes highly dependent on the tool with which the LO is created and managed.

The life cycle of a SALO includes the editing process and the runtime process. To describe this life cycle, we identify four main steps:

- Choice of educational resources that determine the content that is rendered.
- Determine the behaviour using the methods and techniques of adaptive hypermedia, and an implementation with dynamic languages.
- Packaging a SALO following the same process as a regular SCORM object. This process is transparent to users and allows learning platforms to use the SALO.
- Enabling a dynamic adaptive learning environment (Figure 1), provided by the learning platform plus an adaptation engine.

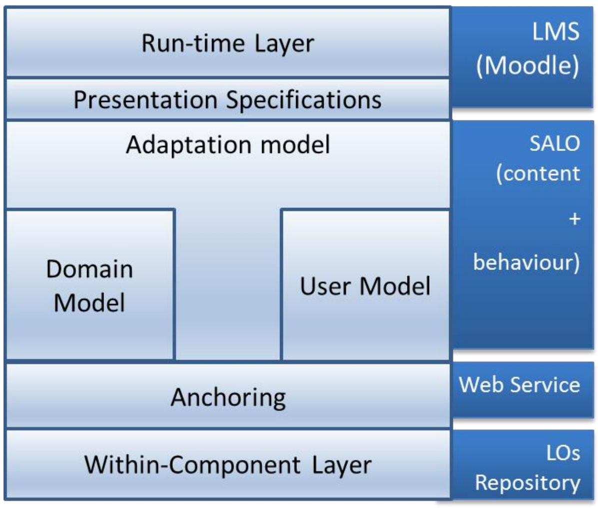


Figure 1. Dynamic adaptive learning environment (adapted from [21]).

A fundamental difference with traditional adaptation methods is that the adaptation engine does not need to be implemented in the learning platform, and it is instead available in the same SALO.

The process consists of two phases: (a) Loading and (b) running the SALO. (a) In order to obtain the necessary information to render the LO, the loading stage obtains educational resources that can be stored online or locally. At the same time, the process retrieves the SALO's behaviour and generates the methods needed for the system to run the SALO henceforth; (b) During runtime, SALOs make it possible to introduce new behaviour through programmed dynamic methods that will make the learning object mutate under the specific conditions of the user, context and learning environment.

4. Implementation Example: Using SALOs in an HTML Course

In this section, we present the actual design and implementation of Shareable Auto-Adaptive Learning Objects through a functional prototype that has been developed to show it feasibility and enable its future evaluation in authentic learning situations. The prototype has been designed to be introduced in a university course on web technologies. The course design is shown in the figure below, along with the navigation. We assume the course is offered in Moodle and graded using rubrics. A rubric is a criteria-based assessment currently available in Moodle, but not in SCORM. We have made rubrics available in SALO to enable lecturers to perform regular assessments and provide feedback to students. The marks and feedback from the assessment allows progressing to the next module or asking students to revise previous concepts.

In this example, we create two SALOs (Figure 2). The first SALO, named "Concepts", contains an introduction and examples of general Web aspects. The second, "Basic development", is oriented to explain client technologies. The navigation uses the default sequential design in SCORM using CAM (Content Aggregation Model). That is, once the first module is completed, the student is allowed to start with the second. When both modules are completed the students are asked to undergo an evaluation.

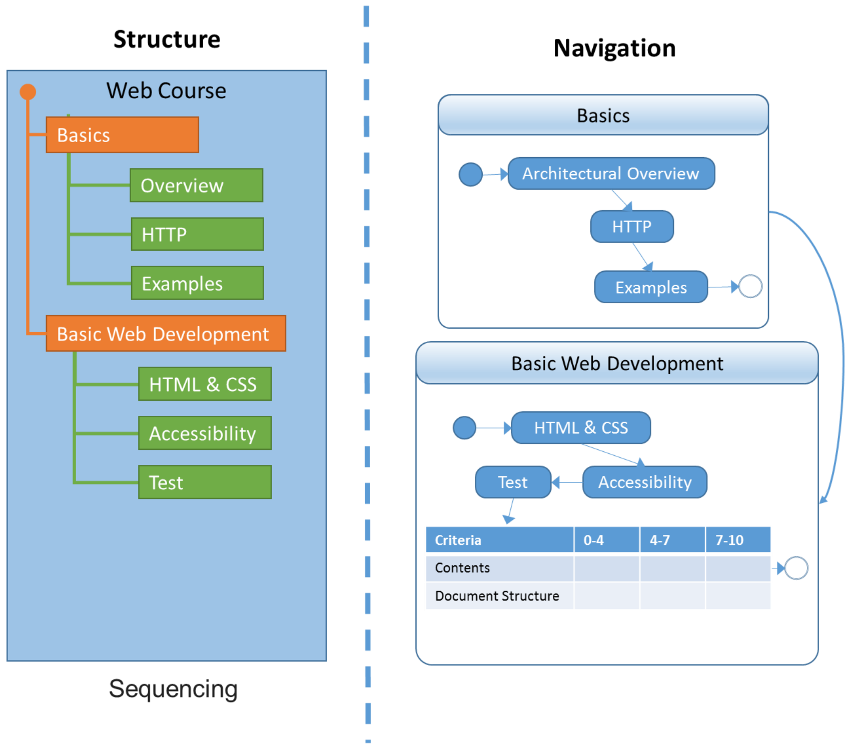


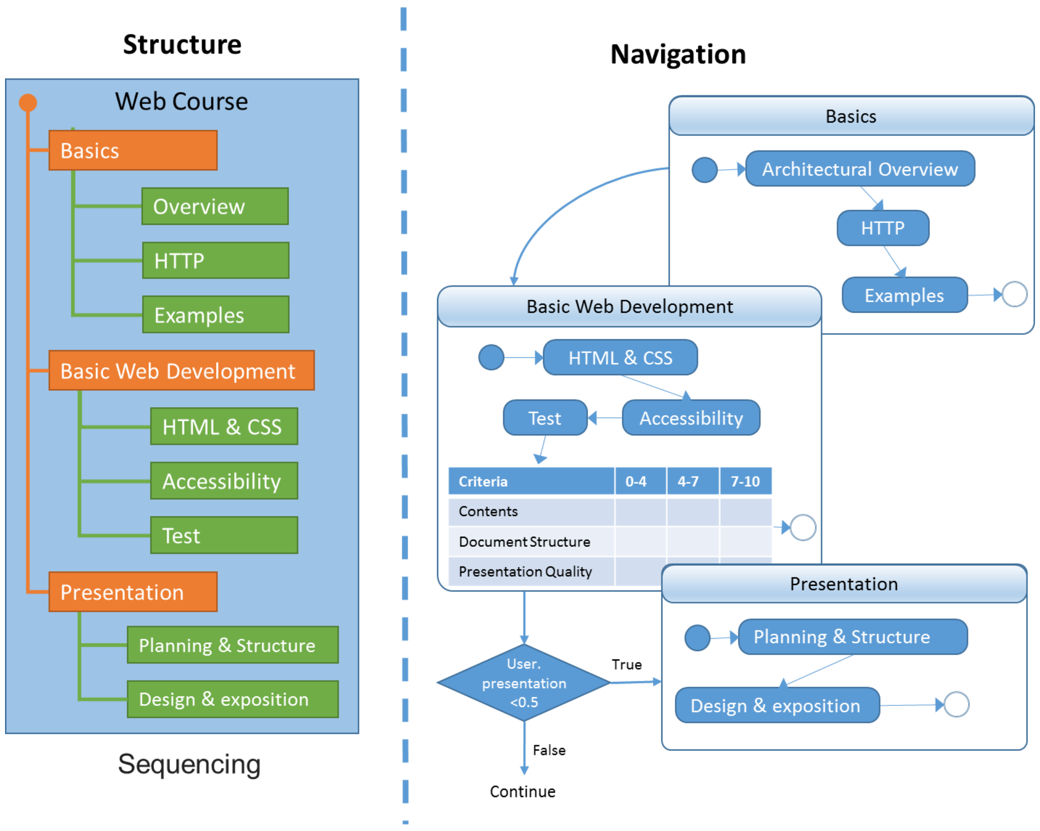
Figure 2. Shareable auto-adaptive learning objects in the HTML course.

The context described above includes automated continuous evaluation and system feedback about the learning level of students, concept development and learning speed. In terms of adaptive learning, SALOs enable runtime content adaptation, changes in the navigation (further explained on next section), as well as individualised assessment methods (rubrics) adapted to the student's abilities and level of understanding. Altogether, these features enable real-time feedback, content, navigation and assessment adaptation, as well as reinforcing dynamic decision making.

4.1. Design Requirements

4.1.1. Adaptation Requirement 1: Adapting Learning Content, Navigation and Assessment "on the Go"

This requirement allows overcoming the problem of modifying an online course while it is taking place as it may occur in, for instance, the following situation. Suppose a lecturer realises that students have difficulties with oral presentations and wants to provide support to presentation skills by adding both a module "Presentation" with new contents, as well as an evaluation of oral presentations in rubrics. She can do so by creating a new learning object named "Presentation" (Figure 2). In addition to introducing new contents with no technical updates or re-starts/re-loads (Figure 3), the lecturer can modify the rubric to include new aspects to the evaluation process, *i.e.*, presentation quality. In this way the lecturer is able to include completely new learning content and navigation without causing any disturbance in the system or to the students.



Criteria	0-4	4-7	7-10
Contents			
Document Structure			
Presentation Quality			

Figure 3. Adding a new "Presentation" learning object at runtime.

4.1.2. Adaptation Requirement 2: Re-Use of a Course in a Different Learning Scenario

Now suppose the following case: with the course in progress, and after the changes mentioned in the previous example have taken place, the Lecturer decides to introduce two new variables: the time each student can commit to the course, and an estimation of the time needed to learn each module. Using these two variables, the students with time restrictions, *i.e.*, those who have a regular job and study at the same time, can better schedule their time and choose the contents to study. Having this information displayed in learning indicators [22,23] allows teachers to make dynamic decisions about the course and the students [24]. In our example, the lecturer, who knows the time constraints and individual needs, can modify the navigation rules of the SALO to adapt the time an individual student spent on a particular content. Changes at this level are possible while keeping access to the Moodle course, without needing to duplicate contents or wait for a student group to complete the course to modify and adapt it for next group.

4.2. Prototype Implementation

For this example, we have created a prototype where the domain model is a web course (see Figure 2) and the user model is the learner, with the consideration of the following parameters: age, gender, presentation score and time spent in the module. The variables have been chosen to fit the example. Age and gender are static variables, and the grade obtained from the rubric, as well as the time to complete the assignment, are dynamic variables, subject to changes during the course. In order to adapt the user model, we have based our design and implementation on the following methods and techniques of Adaptive Hypermedia:

1. Conditional text: It is implemented using all (four) variables. The format of the text is adapted and the resulting paragraphs are displayed.
2. Expanded text: It expands the text presented accordingly to the level of knowledge of the student.
3. Notes: Adapts the presentation of links according to users' interests.
4. Link ordering: Management of links with respect to the level of knowledge and interest of students.

There are two ways of providing dynamism to a SALO object. The first type is concerned with the navigation and represented by pre-requisites and post-requisites, and the second is applied directly over the content of the object.

In the first type (Figure 4), the HTML label "rule" indicates the behaviour of the SALO. Such behaviour is described using a programming code, which resides in the same object, and it is executed when the object is first loaded in the system. The second type is embedded in HTML (code in Figure 5), marked with the special HTML tag "span" and within the tag the method is described along with the parameters and values needed to perform the adaptation. The use of the tag "span" allows the inclusion of this code without affecting any of the content, in such a way that the new behaviour is ignored by web browsers and any e-learning platform that does not include our proposed adaptation engine. This allows the SALO to be incorporated and executed in any LMS; logically the adaptation would not be shown in the absence of a SALO engine.

```
<rule id="path_to_presentation" type="nav">
  <code>
         if user.rubrica.presentation<5
                      showlink(presentation)
                      hidelink(tema3)
         end
  </code>
</rule>
```

Figure 4. Code snippet 1: Behavioural rule described in the SALO.

```
<span class="show_full_explanation" id="0_ADATIVE">
  <code>
                    If (user.remainingTime>2.5)
            echo("This text is a detail explanation……")
  </code>
</span>
```

Figure 5. Code snippet 2: A rule is directly applied to the content.

The code in Figure 4 (above) is processed by the adaptation engine. A query to the user model allows grades of the student to be obtained in the presentation. In this way, the conditional if sentence determines whether to display more information about the presentations or continue displaying the next module available in the course navigation. The methods showlink() and hidelink() are part of the adaptation environment and the LMS. In this case, because we are using Moodle, they have been implemented in PHP, but the same implementation can be done in any dynamic language.

In Figure 5 (below), the rule describes a direct action including the text to be visualised, in such a way that, if the completion time increments, the object will show a more detailed text. This code requires the use of a reserved word such as 'user' to represent the user model or 'style' to indicate changes in style, *i.e.*, font types, colours, *etc.* In this example, a method show_full_explanation() has been included in the same SALO and written by the author of the course. The method echo() is part of the adaptation engine and responsible to display the text to the student.

In our prototype the adaptation process starts when Moodle obtains a SALO to be visualised. At that moment, the adaptation engine loads the content and behaviour in memory and starts its execution, rendering a visualization of the learning content in Moodle. Once the user has finalised the interaction with the object, the next object is requested (either automatically or by using a link). The engine then evaluates the situation using the user model, as well as the navigation and behavioural rules included in the SALO, and triggers the next action by loading the required object, so the process repeats itself. When the user signs out, Moodle stores the last state.

5. Discussion

Adaptation of e-learning systems to users' characteristics and needs is a research line that has led to the development of a number of tools and adaptive methods. Research studies with users in real learning environments have been used to validate the efficacy of adaptive e-learning [25,26]. However, such adaptation has been achieved using static and pre-determined factors. The emergence of e-learning platforms was followed by content specifications and the use of learning objects. LO specifications make it possible to achieve e-learning interoperability but not dynamic adaptation. In this sense, part of the initial problem remains: LO are designed for a specific context and a static user model. The frameworks for the design of e-learning architectures offer non-flexible services, which do not favour real-time adaptation either. In order to make an e-learning system, *i.e.*, Moodle, able to incorporate a LO specification, *i.e.*, SCORM, it has to implement a sequencing and navigation module, as well as incorporate the data model. Any later modifications to the characteristics of the user model imply the need to re-program the corresponding e-learning modules.

The Sharable Auto-Adaptive Learning Object proposed in this investigation, integrates the data model, in such way that the LMS is not any more responsible for defining the behaviour of the LO, but the SALO itself determines it as an autonomous agent. In technical terms, this implies the LMS does not have to codify the sequencing and navigation of the LO, and the SALO needs to be able to represent the presentation layer of the LMS. Future versions of SALO can extend the user and adaptation models to include aspects, such as learner-controlled selection of tasks and feedback. In a recently published parallel study, we have determined how variables that are central to the learning process can be used to allow for the application of adaptive rules to different types of content [14].

The use of dynamic languages is already a solution developed for hypermedia systems through the LAG adaptation language [27] as well as for web-based systems using a domain specific language (DSL) based on Cascading Style Sheets (CSS) [17]. Our proposal is in line with these research directions, with the advantage that in this case it is not necessary to create a new adaptation language. Existing script languages, such as PHP and Python contain dynamic characteristics, allowing for incorporation of new behaviour in real time. The costs and benefits of using dynamic languages have been comprehensively evaluated [28]. However, the performance and convenience of using SALOs in learning courses needs to be evaluated and compared with other learning specifications using both an educational and technical perspectives. From an educational viewpoint, we propose to address the evaluation of the didactic effectiveness [26] of SALO contents following an inter- and intra-subject experimental design to allow measuring changes in students' performance over their instruction time, comparing the adapted and non-adapted learning contents. Technically speaking, we understand one of the disadvantages of this proposal is that it requires interest in computer programming and the understanding of these programming languages. This is also the case for LAG, that requires learning a new programming language and DSLs, which requires expert knowledge.

Our study shows that integrating SALOs can favour the scalability of e-learning solutions. The main requirement is that an e-learning system needs to be implemented using dynamic languages, which is most often the case, *i.e.*, Moodle is entirely developed in PHP.

6. Conclusions

There is a current emergence in research on context-sensitive and ubiquitous e-learning systems. These systems require real-time adaptation. Our research proposes a solution for real-time adaptation in e-learning systems based on the use of dynamic languages. Dynamic languages are capable of including navigation and content adaptation but they have the drawback of having an inferior system performance when compared to static languages.

The use of dynamic languages implies changes at design and development levels in current e-learning platforms. The characteristics of dynamic languages, such as reflection and dynamic code interpretation and evaluation, offer high flexibility but also require high computational power and resources to ensure fluid execution. This requirement can be easily met by modern computer systems. In our proposal, new adaptation strategies are dynamically created according to the user model. The adaptation process is also used for allowing real-time decisions from Lecturers and Unit Coordinators, who can determine if the learning content and navigation fit the learning goals and adapt them accordingly while the system is running. When using SALOs the system can manage both static and dynamic parameters (*i.e.*, task and unit completion times, geographical location, *etc.*). In turn, the adaptation engine uses these parameters to infer (through the behaviour defined on the SALO) the dynamic adaptation to users' needs.

In this paper we propose an evolution of learning objects into sharable self-adaptive learning objects (SALOs), and we describe the characteristics and components of a SALO. The paper also shows an implementation example that uses SALOs as SCORM compliant objects making them compatible with current content specifications and e-learning systems like Moodle. The use of SALOs needs to be evaluated and compared against other content specifications. In our future work, we plan to extend and enrich this specification to incorporate SALO in e-learning platforms that are not written using dynamic languages (*i.e.*, Sakai). At the same time, future versions will integrate SALO with Tin Can Api, with the goal of offering dynamic indicators that can contribute to enhancements of teaching and learning processes.

Acknowledgments: This work has been partially funded by the Department of Science and Innovation (Spain) under the National Program for Research, Development and Innovation: project EDU2014-57571-P. We have also received funds from the European Union, through the European Regional Development Funds (ERDF); and the Principality of Asturias, through its Science, Technology and Innovation Plan (grant GRUPIN14-100).

Author Contributions: Ignacio Gutiérrez and MPuerto Paule conceived, designed and performed the experiments. Juan Ramón Pérez-Pérez contributed materials. Víctor Álvarez, MPuerto Paule and Sara de Freitas wrote the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Álvarez García, V.M. Voice Interactive Classroom, a service-oriented software architecture to enable cross-platform multi-channel access to Internet-based learning. Ph.D. Thesis, University of Oviedo, Oviedo, Spain, 2011.
2. García, V.M.Á.; Ruiz, M.D.P.P.; Pérez, J.R.; Pérez, I.G.M. Presente y futuro del desarrollo de plataformas Web de elearning en educación superior. In Proceedings of SPDECE 2008, Salamanca, Spain, 20 October 2008.
3. Viano, R. SCORM. *ADL Net*; 2015. Available online: https://www.adlnet.gov/adl-research/scorm/ (accessed on 6 June 2016).
4. IMS Global Learning Consortium-LD. Available online: https://www.imsglobal.org/learningdesign/index.html (accessed on 6 June 2016).
5. IMS Global Learning Consortium-Common Cartidge. Available online: https://www.imsglobal.org/cc/index.html (accessed on 6 June 2016).
6. Tin Can API Homepage-Programmable E-learning and Experience Tracking. Available online: http://tincanapi.com/ (accessed on 6 June 2016).
7. Sharples, M. Mobile learning: Research, practice and challenges. *Distance Educ. China* **2013**, *3*, 5–11.
8. Corbalan, G.; Kester, L.; van Merriënboer, J.J.G. Learner-controlled selection of tasks with different surface and structural features: Effects on transfer and efficiency. *Comput. Human Behav.* **2011**, *27*, 76–81. [CrossRef]
9. Vandewaetere, M.; Clarebout, G. Can instruction as such affect learning? The case of learner control. *Comput. Educ.* **2011**, *57*, 2322–2332. [CrossRef]
10. Moodle-Open-source learning platform | Moodle.org. Available online: https://moodle.org/ (accessed on 6 June 2016).
11. Menéndez, I.G.; Ruiz, M.D.P.P.; Pérez, J.R.P. SALO: Sharable Auto-adaptive Learning Object. In Proceedings of the 7th International Conference on Web Information Systems and Technologies, Noordwijkerhout, The Netherlands, 6 May 2011.
12. Ruiz, M.D.P.P.; Díaz, M.J.F.; Soler, F.O.; Pérez, J.R.P. Adaptation in current e-learning systems. *Comput. Stand. Interfaces* **2008**, *30*, 62–70. [CrossRef]
13. Graf, S.; Liu, T.-C.; Kinshuk, K. Interactions between students learning styles, achievement and behaviour in mismatched courses. In Proceedings of the IADIS international conference on cognition and exploratory learning in digital age (CELDA 2008), Freiburg, Germany, 13 October 2008; pp. 223–230.
14. Sanchez-Santillan, M.; Paule-Ruiz, M.; Cerezo, R.; Alvarez-Garcia, V. MeL: A dynamic adaptive model of the learning process in eLearning. *An. Psicol./Ann. Psychol.* **2015**, *32*, 106–114.
15. Brusilovsky, P. Methods and Techniques of Adaptive Hypermedia. *User Model. User-Adapt. Int.* **1996**, *6*, 87–129. [CrossRef]
16. De Bra, P.; Smits, D.; van der Sluijs, K.; Cristea, A.I.; Foss, J.; Glahn, C.; Steiner, C.M. GRAPPLE: Learning Management Systems Meet Adaptive Learning Environments. In *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends*; Peña-Ayala, A., Ed.; Springer: Berlin, Heidelberg, Germany, 2013; pp. 133–160.
17. Montes García, A.; De Bra, P.; Fletcher, G.H.; Pechenizkiy, M. A DSL based on CSS for hypertext adaptation. In Proceedings of the 25th ACM Conference on Hypertext and Social Media, Santiago, Chile, 1 September 2014; pp. 313–315.
18. De Bra, P.; Knutov, E.; Smits, D.; Stash, N.; Ramos, V.F. GALE: A generic open source extensible adaptation engine. *New Rev. Hypermed. Multimed.* **2013**, *19*, 182–212. [CrossRef]
19. Introducing Sakai 11 | Sakai. Available online: https://www.sakaiproject.org/ (accessed on 6 June 2016).

20. Redondo, J.M.; Ortin, F.; Cueva, J.M. Optimizing reflective primitives of dynamic languages. *Int. J. Softw. Eng. Knowl. Eng.* **2008**, *18*, 759–783. [CrossRef]
21. De Bra, P. Design issues in adaptive web-site development. In Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW, Toronto, ON, Canada, 11–14 May 1999; pp. 29–39.
22. Siemens, G. What are Learning Analytics? *Elearnspace* **2010**. Available online: http://www.elearnspace.org/blog/2010/08/25/what-are-learning-analytics/ (accessed on 6 June 2016).
23. Duval, E. Attention please!: Learning analytics for visualization and recommendation. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge, LAK '11, Banff, AB, Canada, 1 March 2011; pp. 9–17.
24. Ruiz, S.; Charleer, S.; Urretavizcaya, M.; Klerkx, J.; Fernández-Castro, I.; Duval, E. Supporting learning by considering emotions: Tracking and visualization a case study. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, Edinburgh, UK, 25 April 2016; pp. 254–263.
25. Hervás, A.; García, F.B.; Peñalvo, F.J.G. A method of assessing academic learning experiences in virtual learning environments. *IEEE Lat. Am. Trans. (Rev. IEEE Am. Lat.)* **2014**, *12*, 219–226. [CrossRef]
26. Paule-Ruiz, M.P.; Álvarez-García, V.; Pérez-Pérez, J.R.; Riestra-González, M. Voice interactive learning: A framework and evaluation. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education; ITiCSE '13, Canterbury, UK, 1 July 2013; pp. 34–39.
27. Cristea, A.; Smits, D.; de Bra, P. Towards a generic adaptive hypermedia platform: A conversion case study. *J. Dig. Inform.* **2007**, *8*, 1–10.
28. Redondo, J.M.; Ortin, F. A Comprehensive Evaluation of Common Python Implementations. *IEEE Softw.* **2015**, *32*, 76–84. [CrossRef]