
Design Patterns for Learning Objects: the Case for Socratic Method

Ismar Frango Silveira, Maria Alice Grigas Varella Ferreira,
Carlos Fernando Araújo Jr

Centro de Ciências Exatas e Tecnológicas – Universidade Cruzeiro do Sul (UNICSUL)
Av. Dr. Ussiel Cirilo, 225 – Vila Jacuí – São Miguel Paulista
08060-070 – São Paulo – SP – Brazil

ismar.silveira@unicsul.br, malice@pcs.usp.br,
carlos.araujo@unicsul.br

***Abstract.** Millenniums ago, Socrates had stated the basis for today's philosophy only by using criticism to set up concepts and rules discussing them with his disciples. This approach, named Socratic Method, has been proven to be also a teaching-learning strategy. Nowadays, even though web-based, collaborative Distance Learning systems are becoming industry standards for representation and deployment of online content for courses, some features can be extracted from Socratic Method in order to answer a recurrent questioning about how technologies could be correctly applied on supporting different pedagogical strategies in order to provide massive, high-quality distance learning content. This paper shows a collection of design patterns for developing fine granular, highly reusable learning objects to be used to build learning resources in computer-supported learning environments.*

1. Introduction

By using critical reasoning, fifth-century Athenian Socrates (469-399 BCE) set all the standards for subsequent Western philosophy. More than this, his dialectic method undoubtedly influenced the way people build knowledge by gathering pieces of conclusive information only through questioning. From Plato to Vygotsky, the acceptance of the notion that knowledge must be built instead of transmitted was early shaped by Socrates' discussions with his disciples. Centuries had passed since Socrates was given poison hemlock to drink, but several questionings still remain from that era. Perhaps most of them can be resumed on a simpler one: how to achieve high-quality, massively-scalable education?

Today's computer-assisted learning systems barely use Socratic-like methods, since they usually rely on instructional, teacher-centered, static content-based teaching approaches. Constructive methods, in other hand, face knowledge as an entity that must be built, even though by other means than questioning-answering Socratic strategies.

2. The Socratic Method

Three main features can be extracted from Socratic Method [Seeskin, 1987]. First, the method is skeptical, in order that it starts by assuming no previous specific knowledge

on certain subject. This skepticism was perhaps inherited from Socrates' contemporaneous Sophists, but while Sophist's principle of ignorance on something was definitive, the Socratic one is provocative, being a crucial element to start knowledge building process. It roughly contrasts with web-based, static content-modeled courses, which usually are presented as sources for uncontestable "truth", and knowledge is meant to be built from these "truths".

Second, it is a conversational strategy, since dialectics are used in order to gather knowledge by means of people's thoughts confrontations. Jean Piaget (2001) stated that knowledge was built from the interaction among apprentices and the objects that represented the content to be learnt, but he didn't take in account conversational issues. Lev Vygotsky (1998), in other hand, stated that social interaction would precede and even supercede interaction with content. In nowadays terminology, while Piaget maintains the intra-psychical knowledge building, Vygotsky statements were about the inter-psychical process of knowledge building.

Last, Socratic Method aims to allow for empirical concept pursuing, since it criticizes proposed definitions for concepts by analyzing particular instances of them. Such bottom-up approach, if applied to web-based courses, would lead to the necessity for specific exercises that must be previously applied in order to target the right topics, thus offering adequate content to students, according to their needs. Table 1 compares these three points of Socratic Method with common instructional distance learning courses.

Table 1 – Comparison among Socratic Method and instructional courses.

Socratic	Instructional
Skeptical	Omniscient
"I know nothing except the fact of my ignorance"	"The truth is inside me"
Conversational	Expositive
Knowledge is built through human interactions	Students are meant to interact mainly with available content
Empirical and Deductive	Assimilable
Concepts can be acquired through examples, as well as new examples can be instanced from concepts	Concepts must be assimilated from available contents.

Next item will detail these three points of Socratic method compared to web-based distance learning, simultaneously proposing some design patterns to achieve Socratic-like experiences in web-based courses.

3. Patterns for Learning Objects

According to Larman (2004), a pattern is a reusable solution template that organizes proven solutions into categories, since they will probably recur in a different problem domain. Roughly speaking, a pattern is a solution for a given problem or a problems' class.

Patterns exist for both design and implementation phases in software engineering process. Design patterns provide high-level, elegant, reusable solutions to commonly encountered programming challenges. Implementation patterns are

predicates on the implementation of a software module whose main characteristics are: purposeful, simplicity and easy automation. Implementation patterns have been extensively used to describe implementation strategies for design patterns. This work deals only design patterns. General design patterns catalogues could be found in Gamma et al. (1995) and Alur (2003). These catalogues generally describe design patterns through well-defined templates, composed by the following parts: Also Known As, Motivation, Problem, Forces, Solution, Example, Implementation, Consequences, Related or Interacting Patterns and Known Uses, suggested by Gamma et al. (1995). Currently, there are some templates that are even more detailed than Gamma's, for instance, the ones used by POSA (Pattern-Oriented Software Architecture) [Buschmann et al, 1996]. On the other hand, other researchers, like Tidwell (2003) and Welie (2003) use simpler templates in interaction area. In this work we simply describe the main aspects of the patterns, without any formalism.

Learning objects are defined here as “any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning” [Wiley 2000]. Development of learning objects usually include a wide range of tasks, like instructional design, multimedia design and text production, as well as audio, video, graphics and pictures' conversion to machine-readable formats. In addition to these tasks, there are other ones, like course authoring, software development, content integration and testing, modification/adjustment, training; and so on. Some of these tasks can be systematized in order to be described for a set of design patterns.

Last section's arisen problems lead to some questionings about the ways Socratic Method could be applied in order to achieve more dialectic, adaptive, knowledge building-driven pedagogical strategies that would be able to go beyond expository, static, purely instructional distance learning approaches. Following subsections will show some software design patterns to be applied on elaboration of learning objects. These patterns were first briefly described by Silveira et al. (2003), being proposed as efficient solutions for the three dilemmas that had been shown at Table 1.

3.1. Skepticism vs. Omniscience

The main reason for why Socratic Method is considered as skeptical is the fact that it looks for pure truths beginning from the complete scratch, since there is an assumption that no pre-knowledge about subjects under criticism can be used as a complete demonstration for any reasoning that could arise from criticism. Instructional teaching-learning processes, in other hand, are based on a sort of “omniscience”, since all knowledge about some subject is supposed to have already been represented somewhere as a learning object, or a set of them. Web-based learning systems are examples of this fact, because even when some specific information about a subject is not found among course's available learning objects, the Internet wideness is an invitation to search for such information at somewhere else, since it is supposed to be already implemented. Clearly, this approach doesn't guarantee knowledge building, as well as it doesn't assure that such specific information will be found in an way that would fulfill, in a proper and exact manner, those students' needs and requirements. In other hand, would learning objects that fit to a learning situation be useful at other learning conditions? In order to solve these problems, two patterns will be proposed.

First, *Fine Granularity* pattern is meant to solve the problem of developing high-reusable learning objects. This pattern states that learning objects must be fine-granular enough to represent an atomic piece of information. Otherwise, they can only be containers for other learning objects.

For instance, HTML-based hypermedia learning objects tend to have a very poor granularity, since they aren't structured according to their types, relationships or goals. Thus, learning objects' reuse is strongly discouraged, unless finer granularities are provided when designing learning objects. Figure 1 shows an UML class diagram that depicts different media types that are commonly used when building HTML webpages, and relationships that are usually found among them.

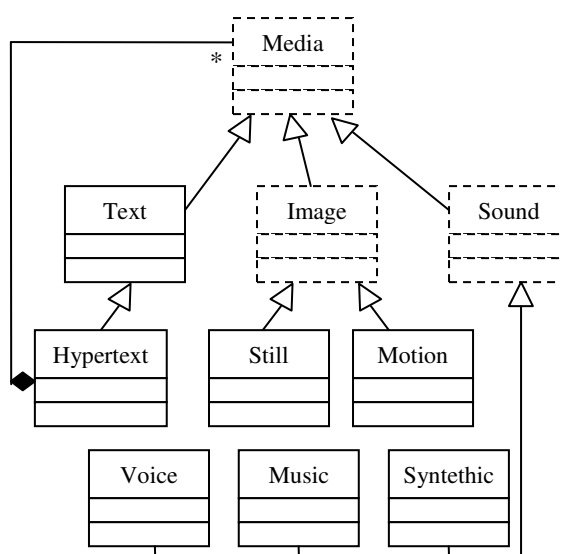


Figure 1: UML Class Diagram for HTML-based content

As seen in Figure 1, Hypertext is indeed Media's children's more susceptible to have poorest granularity, since the former maintains a composite – and not aggregate – relationship with the last, which means that lifetime of all learning objects that are contained into a Hypertext object are tied to this one's lifetime.

Figure 1 can be extended in order to have different hypertext kinds than that HTML-like, for instance, structured hypertext. More than this, high-generality containers than hypertext-based must be considered, since learning objects types can vary.

HTML-like web courses can avoid being “omniscient” by adequately using granular learning objects that must be presented to apprentices in a manner that would encourage knowledge building instead of information gathering. Using structured containers instead of HTML-like hypertext, learning objects could be arranged in a way that would allow a semantic approach for searching operations, which would be a helpful feature including for reusing purposes. Figure 2 shows this approach implemented as an adaptation for a part of Figure 1's diagram.

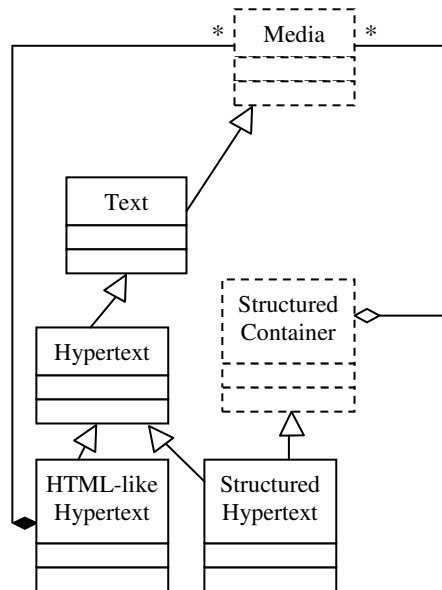


Figure 2: UML Class Diagram adapted for Fine Granularity pattern

Figure 2 shows a different relationship among Structured Container and Media, providing lifetimes' lower coupling. Obviously, this pattern alone is not enough to provide as much skepticism as Socratic methods, but at least it ensures that a course can be planned to force apprentices to elaborate, by themselves, mental structures whose evolution would bring to them other learning objects.

Socratic dialectics implementations, however, are not a new. Early 1970's ITSs (Intelligent Tutoring Systems), like James Carbonell's SCHOLAR [Carbonell 1970] directly implemented Socratic dialectics to teach South America geographical issues.

Nevertheless, our goal is not to propose direct implementations of Socratic Method by means of question-answer interactions, but through an indirect approach of it, trying to get apprentice's model based on feedback that are perceived at each interaction with learning objects.

In spite of the fact that it isn't a complete skeptical approach, since it begins with small, discrete pieces of information, according to criticisms that are done on it, new learning objects are proposed to the apprentice, which leads to adaptation problems.

How to select and present learning objects to apprentices that are adequate to their specific needs? This problem can be solved by adopting a second pattern, named *Learner-Adaptation*.

The Learner-Adaptation pattern can be described by the following general rule: all learning objects must be created and structured in a way that they can be adapted to students' actual learning requirements.

Figure 3 shows an UML class diagrams for this pattern.

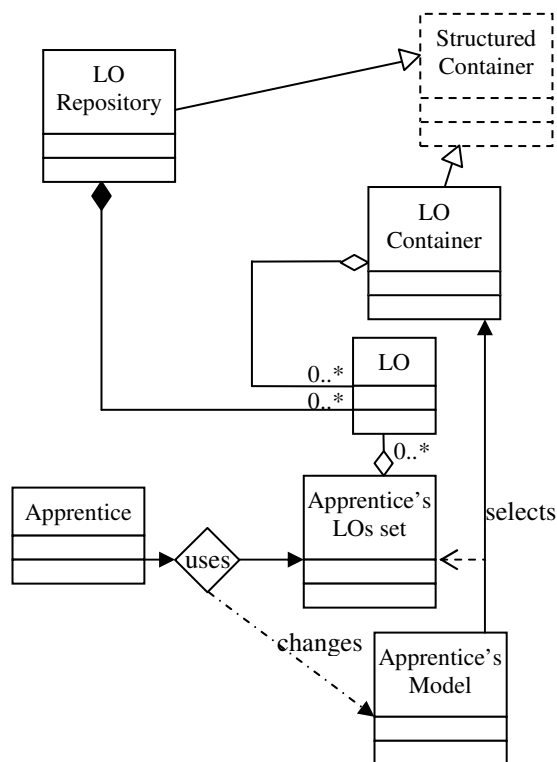


Figure 3: UML Class Diagram adapted for Learner Adaptation pattern

Analyzing Figure 3 shows that applying Fine Granularity pattern is the first step to achieve Learner Adaptation. Based on this, Apprentice's Model selects adequate learning objects from a LO Container, thus compounding an Apprentice's LOs set. Apprentice's using of these learning objects can provoke changes on Apprentice's Model.

Next section will discuss another dilemma, this time related to conversational issues of Socratic Method.

3.2 . Conversation vs. Exposition

In this section, the problem to be discussed is how conversational strategies could replace expositive ones in distance learning courses?

A possible solution for this is the *Agent-Agent Interaction* pattern, which affirms that interaction among apprentices and teachers – named learning agents – can allow learning objects to be shared among agents, thus decentralizing learning objects' managing responsibilities.

Figure 4 shows an UML Class Diagram which depicts the Agent-Agent Interaction pattern.

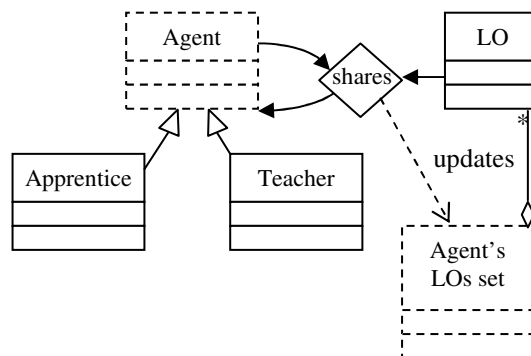


Figure 4: UML Class Diagram adapted for Agent-agent interaction pattern

Certainly the most evident characteristic of Socratic Method is its dialectics, which represent the basis over every criticism is done in order to endorse knowledge building. For most of distance learning supporting tools, however, Socratic dialectics are barely more than a theory, since a large portion of applicable teaching-learning activities is seriously affected because of the lack of interaction tools that would support such dialectics.

In addition, apprentice-centered teaching-learning philosophies state that learning process must not focus on teacher as content creator and explainer, stressing student's role as chief on learning process, thus agreeing with Vygotsky's theories about inter-psychical knowledge building [Vygotsky 1998], which are widely based on Socratic dialectics.

The ability of sharing learning objects becomes closer to Socratic dialectics if agents are also allowed to create their own learning objects. This point leads to another discussion about the last of three dilemmas showed at Table 1. This discussion is taken at next section.

3.3 . Deduction vs. Assimilation

Considering that learning processes will succeed by means of pedagogical strategies application over available learning objects, it is possible to admit that the supported number of learning activities can go up and down, according to the amount of learning objects that are deployed by teachers.

In fact, when having in mind teacher-centered instruction model, this is what really happens. Learning possibilities goes until the limit that is imposed by repository's size, since learning activities goes on while there are unnoticed learning objects, or untried kinds of handling them. Thus, new activities will be possible only if new learning objects are shared.

In instructional, teacher-centered strategies, all learning objects deployed by him must be assimilated by students. Even if they were also allowed to deploy learning objects, it doesn't assure learning object's authoring, since apprentices could be just deploying someone else's object, thus reproducing omniscient patterns, not being able to put into effect skeptical skills they are meant to obtain.

The problem being discussed can be resumed as follows: how to ensure that students are able to deduce new concepts from other ones? A possible solution to this is the *Hybrid Authoring* pattern, which allows them to build new learning objects through interaction among them, or among them and other learning objects. Figure 6 shows an UML class diagram that represents such pattern.

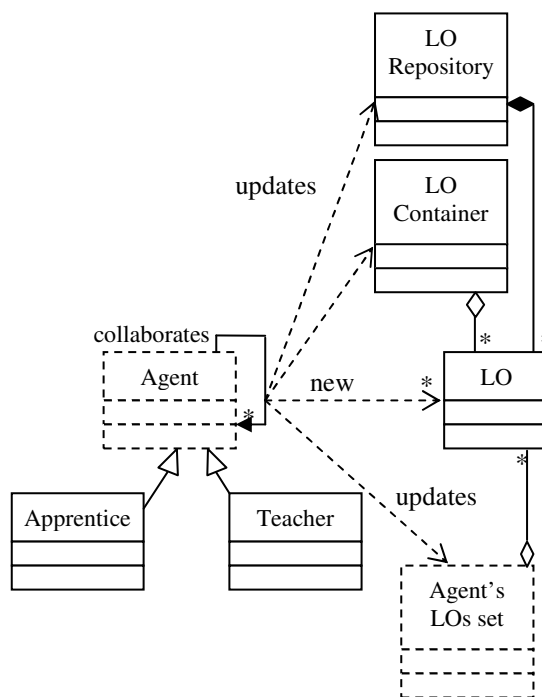


Figure 5: UML Class Diagram adapted for Hybrid Authoring pattern

Hybrid Authoring stands for the fact of that every learning agent can contribute with their own learning objects, which can be pre-built content or some kind of content made on-the-fly. However, it must be pointed that hybrid authoring is not solely a sort of juxtaposing of static and dynamic authoring, since this kind of authoring brings new possibilities referred to the collaboration among learning agents through the sharing of learning objects.

Hybrid authoring is achieved when dynamic authoring can be done over static, commonly pre-built, pre-deployed content. However, this way of authoring also includes the dynamic insertion of new content into some learning context. This kind of authoring is generally obtained by mixing classic static authoring tools with CSCL (Computer-Supported Collaborative Learning) systems like PIAGET, in order to allow the organization and management of pedagogical content, integrating static and dynamic content in a well-structured, concise context.

4 . Conclusions

A very large range of work has been done in the recent years about Learning Objects, and an increasing number of tools are being continuously developed in order to make the processes of creation and deployment of pedagogical content for web-based courses effective. Nevertheless, there are various such systems that are, in a subtle analysis, barely more than toolkits for supporting authoring processes, together with deployment tool for HTML-based, static web pages that encapsulate all content that is expected to

accomplish course curricula. In many cases, the final product is nothing more than a widely available, but static, non-interactive, hardly reusable online material that is often closer to e-books than to learning object-based electronic contents. Besides, the whole software engineering process beneath this development is often done in an *ad hoc* manner, being hardly systematized. In order to address this specific problem, the present paper showed four patterns – Fine Granularity, Learner-Adaptation, Agent-Agent Interaction, Hybrid Authoring – for applying on development of Learning Objects. Such patterns were based on Socratic Method, which encourage dialectics as learning process' main foundation.

Future work includes a more complete description of these patterns according to POSA's templates, which is currently being built. Further work will also include some ontology-driven patterns that would allow the building of architectures for classifying all hybrid-generated learning objects, in order to improve their retrieval, for instance. Practical examples of learning objects built under these patterns, being able to cope with metadata standards, like LOM (Learning Object Metadata) [IEEE, 1999], SCORM (Sharable Content Object Reference Model.) [ADL, 2004], are also being planned for the future.

5 . References

- ADL (2004) SCORM 2004 Documentation, <http://www.adlnet.org/scorm>. Visited September 26, 2005.
- Alur, D. et al. (2003) Core J2EE Patterns, 2nd Edition, Prentice Hall.
- Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M. (1996) Pattern-Oriented Software Architecture - A System of Patterns. John Wiley and Sons Ltd, Chichester, UK
- Carbonell, J. R. (1970) "AI in CAI: An Artificial Intelligence Approach to CAI", In IEEE Transactions on Man-Machine Systems, v. 11, pp.190.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading.
- IEEE Learning Technology Standards Committee - LTSC (1999) Learning Object Metadata: Draft Document v3.6, <http://ltsc.ieee.org/doc/wg12/LOM3.6.html>. Visited September 26, 2005.
- Larman, C. (2004) Applying UML and Patterns, 3rd ed. Sebastopol, Prentice Hall PTR.
- Piaget, J. (2001) The Psychology of Intelligence, London, Routledge.
- Silveira, I; Ferreira, M. A. G. V.; Araújo Jr., C. F. (2003) "From Socrates to PIAGET: Patterns for Distance Learning". In.: Proceedings of the 3rd IEEE Conference on Advanced Learning Technologies, pp. 402-403, Athens, Greece.
- Seeskin, K. (1987). Dialogue and Discovery: a Study on Socratic Method, New York, State University of New York Press.
- Tidwell, J. (2003) User Interface Patterns and Techniques, <http://time-tripper.com/uipatterns>. Visited September 28, 2005.

-
- Vygotsky, L. S., Roeber, L. W. (ed.) (1999). The Collected Works of L.S. Vygotsky : Volume 6: Scientific Legacy (Cognition and Language: A Series in Psycholinguistics). Plenum US, Cleveland
- Welie, M. van, (2003) Patterns in Interaction Design, <http://www.welie.com>. Visited September 28, 2005.
- Wiley, D. A. (2000). "Learning Object Design and Sequencing Theory". PhD Thesis. Provo, UT: Brigham Young University.