Design of e-learning objects for CALL in SYNTACTICS

Dr. Nick Armitage, University of Sunderland, Sunderland, U.K., Nicholas.Armitage@sunderland.ac.uk
Dr. Chris. Bowerman, University of Sunderland, U.K., Chris.bowerman@sunderland.ac.uk

Abstract
The production of electronic course content for delivery on Virtual Learning Environments which employ the growing number of standards and specifications in this area are intended to enable the electronic teaching material to be re-used. This is accomplished by providing clearly defined approaches to describing and organising content so that the material can be made available and discovered on a number of learning platforms. However this confuses the re-use of the object with pedagogic re-use. We look at this issue, call for a greater use of software engineering expertise and the return of control of content creation to the classroom tutor. To this end we also present Syntactics - a computer-aided language learning (CALL) system which focuses on a novel form of electronic re-use of content.

1. Introduction
Syntactics is a computer-aided language learning (CALL) system which focuses on a novel form of electronic re-use of content. Re-use of electronic content is much in vogue and a gamut of e-learning standards have arisen since the 1970s in order to facilitate this. The standards rest on providing common mechanisms for the packaging of content so that it can be unpacked and used on a variety of electronic, virtual learning environments (VLE). It is argued that the appropriate description and packaging of electronic content will enable it to be re-used. However this argument is fundamentally flawed: interoperability may enable a learning object (LO) to be re-deployed on another VLE but this is not the same as pedagogic re-use. There are therefore two aspects to re-use: software re-use and pedagogic re-use. Both these areas of e-learning are ripe for further investigation.

In terms of software engineering it should be possible to draw on the experience gained in building re-useable software based on lower level objects which can be re-combined to form conglomerate objects of greater complexity and capable of increased functionality. Lessons learned in terms of the cohesion and decoupling need to be applied to the design of e-learning objects. However these may not simply apply without adaptation. Content is different from program code and it is quite possible that there will be trade-offs in this area e.g., passing user interface data may increase usability but requiring the passing of large numbers of parameters between learning objects is likely to create barriers to re/use.

What is needed is a greater focus on the pedagogical re-use of content. In order to accomplish this the authoring of the learning objects needs to be put back in the hands of the tutors. There are two key approaches to this: limited and full access. Limited access would permit a restricted customisation of learning objects such as more minor modifications to canned text whilst a full-access approach would allow the complete re-authoring of content by tutors using authoring tools which would permit content to be created, re-used and shared (in full or in part) with others.

This implies a fundamental re-think of the content production lifecycle. There needs to be a move away from the need for intensive technical knowledge of software engineering to a framework for content development which permits tutors to be supported in terms of sound pedagogical approaches.

It is with this in mind that our Syntactics system has endeavoured to put the teacher back in control of the teaching materials by providing an environment in which they can take ownership of the electronic teaching materials which they create, re-use them and share them
in full or in part with others. Such an approach moves the focus of development to a flexible framework for content production and delivery. Syntactics makes use of a different type of learning object centred around XML technology which is hidden from the tutor by means of a friendly editing interface. The Syntactics approach is focussed on the pedagogic requirements of the tutor who enters their own content, exercises, self assessments and feedback information on a local machine before making it available to learners. The content is packaged in a re-useable form which can be despatched to learners and also be cut and pasted (in whole or in part) by the tutor.

2. The nature of re-use
Reusing things is not new! Teachers reuse successful teaching material both because of its fitness for purpose and also because it enables better use of their resources. For example, similar examples and illustrations might be used with several classes, similar experiments might be run each year to illustrate the same point, standard texts might be used and so on.

Software engineers also soon realised the need to reuse program code to save machine resources on the relatively puny early computers (Clements 95). Today it is not so much the computer resources that are scarce but the human resources hence the desire to reuse program components has become one more related to effort. The key driver at the moment in the area of eLearning is that of software reuse hence we shall look at key issues here and then turn to potential conflicts with teachers needs to reuse educational material (so called pedagogic reuse).

2. 1 Software Reuse
The BCS Reuse Group (BCS 2006) define Software Reuse as “the planned use of Software Artefacts in the solution to multiple problems”. Initial moves to produce components were made in the production of subroutines which would be used as black boxes by programmers saving them much effort. These initiatives soon became more principled as Dijkstra raised the importance of the structuring of programs and pointed out that independent elements could be produced separately. Then Parnas introduced the notion of being able to hide the information contents of components (the implementation) so that as long as their means of connection (interfaces) with the outside world was well understood, side effects were avoided and the code performed in given ways on failure. Given these factors it was possible to subdivide the system into parts which could be rewritten, replaced and reused. These notions have lead to the object-orientated approaches of today where it is no longer the writing of the individual components but their integration which is the key task (often in particular architectural frameworks). The degree of integration is typically measured in terms of cohesion (the tightness with which related program features are grouped together to make it tighter and more maintainable) and coupling (the relatedness of components to each other). The general ideal for a good system is one with high cohesion and low coupling since this makes the code perspicuous and easy to maintain. However, there are tradeoffs since the most cohesive code is a single related program with no sub components and hence no saving of human resources. Today, in modern computer systems the typical trade-off is towards high coupling with classes whose methods are highly related in both the nature of and the activities which they perform.

For content production there is also a need to ensure well-motivated software designs in which the components of content are sensibly subdivided and integrated. Current learning standards such as the IEEE 1484.12.1-2002 Learning Object Metadata (LOM) standard are clear about the need to subdivide learning material into smaller
subcomponents but are less clear about how these can be integrated (e.g., no coupling mechanisms are currently specified). As in traditional software engineering there is a complexity trade-off with increased coupling leading to increasingly large systems which are less easy to re-use and with less coupling meaning that there is little meaningful re-use possible. Perhaps part of the way out of this bind is to make use, as originally intended by AICC (AICC 2006), of content management and virtual learning system to integrate objects. But to this end we will need to see a greater debate in terms of the nature and appropriateness of creating interfaces between learning objects.

2.2 Pedagogic Reuse
Most CAL, and its modern day equivalent elearning, content has favoured a limited, rote-style of teaching with little scope for the tutor to customise the content for particular groups, their learning goals and needs. In the area of Computer-Aided Language Learning (CALL) it has lead to imitative, audiolingual teaching methods which are variants of traditional teaching approaches. Perhaps the reason for this is due to a misunderstanding of the degree of customisation of learning that takes place in the classroom in general and the language teaching classroom in general:

“However good the material is, most experienced teachers do not go through it word for word. Instead, they use the best bits, add to some exercises and adapt others. Sometimes, they replace textbook material with their own ideas – or ideas from other teachers and books – and occasionally they may omit the textbook lesson completely” (Harmer 1998)

A peculiarity of language courses which accentuates the need for customisation is that they vary enormously in length and content because they are tailored to particular purposes, are hence more unpredictable and render planning in detail beyond a few days is unrealistic. Perhaps at an extreme end of the need for tutor customisation, language teachers need considerable flexibility in readily re-useable, customisable content.

The approach typically taken in the emerging educational standards such as the IEEE 1484.12.1-2002 Learning Object Metadata (LOM) Working Group, which has produced a standard on the nature of learning objects for learning content is based on a fairly traditional view of object-orientation in which the reuse is in the integration of existing components of “code” (here learning content):

“For this Standard, a learning object is defined as any entity, digital or non-digital, that may be used for learning, education or training” (LTSC 2002).

What this omits is the need to pedagogically reuse the content in different situations with different (or the same) group with bespoke customizations to meet the needs of particular learners and their tutors goals. At the heart of this omission is a misalignment in terms of the expected design approach for content.

2.3 Design Approaches
Plass (1998), who criticizes CALL approaches for ignoring the user in the design and evaluation process too frequently, reviews a range of techniques for evaluating the user interface of CALL software, distinguishing four basic approaches:

- the ‘craft’ approach to design considers each project to be unique and hence has no under-girding philosophy
• the ‘enhanced software engineering approach’, adapts traditional software engineering theories with user and task analysis
• the ‘technologist approach’ does not impose interface design theory on the user interface but makes use of software tools to analyse user behaviour and produce an optimal system
• and finally the ‘cognitive approach’ makes use of cognitive psychology to predict interfaces that users will find effective

Since there is no settled theory of learner performance on particular tasks the most pragmatic approach is a strong element of software engineering mixed in with the crafting approach which tutors typically need for real-world teaching. In other words whilst the overall system needs to obey good software engineering principles the content needs to be created and reused by more craft orientated approaches.

3. Syntactics
Such an approach lies at the heart of the Syntactics system which enables tutors to create elearning materials, customise them (both in terms of content, sequence and interaction) using a common user interface which requires no specialist IT skills.

In a typical scenario, a teacher might wish to provide study materials for a Greek class, would then go to the Syntactics website and select, for example, a Multiple Choice Question (MCQ) exercise. A blank template appears as a web page into which the tutor enters the task description, possible answers and feedback. The completed template is saved (others may be entered as required) and is then able to display and respond to user interactions. The content is made available to students by attaching the created files to an email and sending them to the student who, on receipt saves the attachments before opening the first one. When opened the attachments connect to the Syntactics site via the internet, where the data is displayed (using the latest version of the e.g. MCQ object) and the student can begin study. Alternatively, the teacher can put the files on the Internet and link to each one from a chosen web page.

As in LOM, this approach enables reuse and extensibility because each task type is a self-contained object (actually XML data for the blank task template used by the teacher). The file can be reused because it can be included among any list of Syntactics tasks (or indeed as part of a LOM-based course, if accompanied by a metadata file) and customised either by editing the XML data files in the Syntactics environment or by simply inserting a file (or link) into a sequence of attachments (or web links).

4. Conclusion
The LOM approach to providing elearning is clearly viable and offers benefits in terms of increased access across a variety of platforms on a just-in-time basis. However in taking away the content creation from the tutor it risks reducing the depth customisation of teaching materials by tutors and hence reducing the effectiveness of some more traditional learning experiences. For this reason we propose systems like Syntactics which, not only compares well for reuse and extensibility with LOM but, we believe, provides added value in offering other teachers the opportunity to edit existing materials in detail through a simple browser interface with no particular IT knowledge. Each task, on opening, is editable by the user. A teacher can therefore open a task by another teacher, and make small changes (e.g. correcting spelling) or more substantial ones, such as changing a language focus such as a tense from one to another. This makes it possible, especially where the LOM-based courseware does
not provide a sufficiently seamless learning experience, to provide highly targeted materials as needed in the CALL and general CAL classroom.

Similar criticisms can be levied at robot control systems in teaching environments. The control languages are often very unfriendly for users and need to be made more accessible to those who tutor these systems. One approach would be to make use of object-based editors which could shield and support the learner of the language. The tutor would be able to open up the control language component and customise it to make it easier to introduce to learners, progressively removing the scaffolding as they progressed and mastered the full language as needed.

5. Acknowledgements
This work is partially supported by EC-US Grant U8041.

6. References
AICC (Aviation Industry CBT Committee): http://aicc.org

BCS Reuse Group http://www.dur.ac.uk/liz.burd/reuse/ accessed 1/20/06

Clements, P. (1995), The American Programmer, 8(11), From subroutines to subsystems: component based software development


Syntactics: A practical application of the system for teaching a Biblical Greek is at http://www.callinguistics.com/greek/default.asp and blank lesson templates are at http://www.callinguistics.com/greek/tasks/default.asp