ALT-J, Research in Learning Technology
Vol. 14, No. 1, March 2006, pp. 117–129

Accessibility and adaptability of learning objects: responding to metadata, learning patterns and profiles of needs and preferences

Steve Green*[a], Ray Jones[b], Elaine Pearson[a] and Stavroula Gkatzidou[a]
[a]*University of Teesside, UK;* [b]*London Metropolitan University, UK*

The case for learning patterns as a design method for accessible and adaptable learning objects is explored. Patterns and templates for the design of learning objects can be derived from successful existing learning resources. These patterns can then be reused in the design of new learning objects. We argue that by attending to criteria for reuse in the definition of these patterns and in the subsequent design of new learning objects, those new resources can be themselves reusable and also adaptable to different learning contexts. Finally, if the patterns identified can be implemented as templates for standard authoring tools, the design of effective, reusable and adaptable resources can be made available to those with limited skills in multimedia authoring and result in learning resources that are more widely accessible.

Introduction

Drawing on our experiences of creating and using learning objects for teaching Java as a first programming language (the Java Project), we will:

- Explore the potential of accessibility profiles in the context of the development of learning objects.
- Demonstrate how anonymous user profiles can influence the design patterns and templates of learning resources and how new accessible learning objects can be derived from them.

*Corresponding author. Accessibility Research Centre, School of Computing, University of Teesside, Middlesbrough TS1 3BA, UK. Email: s.j.green@tees.ac.uk

ISSN 0968-7769 (print)/ISSN 1741-1629 (online)/06/010117–13

DOI: 10.1080/09687760500479811

By focusing on user profiles and design patterns we will argue that accessible (and usable) learning objects can be achieved by designing for adaptability. This is a departure from the well-documented and frequently proposed approach to designing for accessibility: that of universal design (sometimes called design for all). It does, however, reflect current developments. For example, The IMS (2004) 'AccessForAll' specification currently proposes an adaptability model for digital resources and services, which attempts to match resources and services to users' needs and preferences. Although the motivation of the 'AccessForAll' specification is primarily to ensure the accessibility of resources, it also, almost as a by-product, ensures that resources are more usable and better designed. The new proposals move away from the view that web resources can be sensibly designed to meet the needs of everyone through strict adherence to standards and universal design principles. There is a new recognition that addressing all issues of accessibility, multiculturalism and language in a single standard could never be realised: one group or other would inevitably be excluded. However, we argue that the idea that resources should be adaptable (and adapted) to the needs of individuals or groups is eminently sensible.

In exploring issues of accessibility, adaptability and learning objects, the term 'accessibility' is used to mean availability to a wide audience including those with specific needs; 'adaptability' is used in the context of the ease with which a digital resource can be modified to meet different user requirements, and the terms 'learning patterns', 'learning objects', 'resources' and 'components' represent a loose hierarchy of reusable elements that together form a learning solution.

Metadata and profiles of needs and preferences

Dublin Core metadata

Dublin Core (DC) in its simplest form is an interoperability standard for catalogue information on digital resources. Books, journal articles, webpages, videos, digital images, learning objects or information relating to almost anything anyone might wish to reference could be considered a digital resource. The Dublin Core Metadata Initiative[1] proposes an abstract model consisting of a description made up of one or more statement about a single resource and optionally the unique resource indicator of the resource being described. Each statement is made up of a property and a value (or value unique resource indicator). One advantage of DC over other standards is that it is relatively simple, consisting of only 15 core properties, and consequently forms the basis of many other standards particularly for archives or digital repositories (e.g. the Open Archive Initiative [2004] metadata harvesting standard). Additionally, the core properties or elements are fairly obvious things such as title, subject, date, creator, contributor, and so on.

DC statements can be encoded directly within the object or resource itself typically if it is an XHTML document. More commonly a separate catalogue is held to contain the DC metadata elements and reference information. Essentially, the DC elements typically form a separate database often referred to as a resource catalogue. In the case of the 'Java Project', learning object references and metadata are held in a learning resource catalogue (LRC3).

Accessibility profiles

As well as knowing what resources they are dealing with and their properties, resource creators may also be interested in characterising users so they can match available resources to them. This is often referred to as user profiling. User profile information can be used for accessibility purposes, but in many instances context or preference may be equally important. For example, a car driver needing access to web or PDA information may be in a similar position to a blind user in that neither can handle primarily visual material but could probably usefully access audible descriptions or instructions. Consequently it may be possible to identify this profile as a non-visual or auditory profile. Equally, other contexts may require large text (e.g. visual presentations to large audiences or to someone with a visual impairment). A non-auditory profile might mean that audio material needs a transcription and video needs captions or subtitles. In broad terms there is a need to be able to define a user's contextual profile as a set of requirements for services and resources. Typically the profile will define the user's human–computer interaction requirements in terms of visual, auditory and tactile components. The three main elements of the profile will be:

1. Display or output (typically visual but could be an auditory screen reader or tactile Braille display).
2. Control or input (typically keyboard and mouse but could be switches, touch-screen, joystick tactile devices or an auditory voice recognition system).
3. Content (primarily visual, auditory media or textual components that can be read or transformed into auditory components by a screen reader).

In the IMS 'AccessForAll' proposal (IMS, 2004; Nevile, 2005), an 'adaptability' element is employed to identify a set of user needs and preferences. This is considered an important enough extension by the Dublin Core Metadata Initiative to consider incorporation into the DC standards itself.

Profiles of needs and preferences

The IMS 'AccessForAll' initiative also proposes an anonymous profile of needs and preference (PNP). The profile is anonymous in that there is no need to know who the user is or why they require the specified support. Also, choices and options are considered just as important as absolute needs; for example, a user might express a preference for Braille output, but with an indicator that auditory substitution is also acceptable but visual elements are unacceptable. Another profile might express the need for large text or a screen reader. In essence this is encoded using a predetermined structure and vocabulary (see Figure 1).

display	screenReader
	screenEnhance
	textReadingHighlight
	braille
	tactile
	visualAlert
	structuralPresentation
control	keyboardEnhance
	onscreenKeyboard
	alternativeKeyboards
	mouseEmulation
	alternativePointing
	voiceRecognition
	structuralNavigation
	codeinput
content	alternativesToVisual
	alternativesToText
	visualText
	alternativesToAuditory
	learnerScaffold
	personalStyleSheet
	extraTime

Figure 1. IMS 'AccessForAll' PNP vocabulary

The proposed structure and vocabulary encourages the designer to check and build for adaptability. For example: Is the display fully accessible to a screen reader? Is mouse emulation or alternative pointing catered for? Does the content cater for personal style sheets? These factors and more can be coded into the adaptability element using the items identified in Figure 1.

The important point is that, in principle, if digital resources are associated with rich metadata and a detailed profile of needs and preferences are available, there exists the necessary pre-requisites for designing and developing accessible applications through adaptable resources. Before considering in detail the process applied, the following section takes a closer look at the application area under consideration.

Learning objects and metadata

Criteria for reusing learning objects

The arguments in favour of the reuse of learning resources include economic ones (Downes, 2001) and those of quality (Jones, 2004). However, in order for learning resources to be effectively reused they should be designed with reuse in mind: they should be cohesive and decoupled from other resources (Boyle, 2003). In other words, a reusable learning object should be concerned with a single topic (i.e. cohesive) and not be unnecessarily linked to (i.e. decoupled from) external resources; thus, a learning object should be 'an independent and self-standing unit of learning content' (Polsani, 2003, section 2.2).

In order to be as widely reusable as possible, learning objects should also be context free. However, freedom from a particular learning context may reduce their pedagogical effectiveness; therefore reusable learning objects, while being free from any particular context that would restrict their reuse, should still be adaptable to the specific context in which they may be used (Jones, 2004).

Patterns

Expert practitioners use their experience of solving problems in the past to build on and create new solutions in new situations.

> One thing expert designers know not to do is to solve every problem from first principles. Rather, they reuse solutions that have worked for them in the past. When they find a good solution they use it again and again. Such experience is part of what makes them experts. (Gamma *et al.*, 1995, p. 1)

These reusable solutions that Gamma *et al.* refer to are design patterns.

> Each pattern describes a problem which occurs over and over again in our environment, and then describes the core solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. (Alexander *et al.*, 1977, p. x)

Here, Alexander *et al.* were referring to patterns in architecture and town planning, but core solutions in any area of design can be described in terms of patterns. It is these patterns that will be extracted from the learning objects referred to earlier for reuse in creating new learning resources. The aim in doing this is to help learning object designers by providing them with a set of design ideas in a structured way that will clearly articulate the design problem and its solution (Goodyear, 2005). This will enable the designer to produce effective and adaptable learning resources in an efficient manner.

In simple terms, therefore, a pattern defines a common problem and proven solution, which an experienced designer may choose to adopt. In practice, patterns are implemented as standard sequences of learning components (here 'pages') compounded from learning objects and resources. They are generally implemented as a selection of reusable frameworks with slots for the learning components. For example, a particular type of learning activity might best be presented using a sequence composed of a text-based explanation of the principles involved, an illustrative animation, a review question and a practical exercise. This learning pattern might be presented in a framework with four separate pages each embedding a separate learning object. This learning pattern forms a solution to a learning event, which can be reused in designing an appropriate learning sequence.

Practical application: the Java project

The design and successful use of learning objects to complement a first-year degree course in the programming language Java is well documented as part of an innovative blended learning approach to the teaching of programming to first-year computing students (Boyle *et al.*, 2003). The learning objects were designed by a team of subject experts and a multimedia developer who authored them in Flash. Each learning object was self-contained and was intended to be used either as a stand-alone resource or as an embedded link within a compound learning object. Each object was made up of a sequence of frames, which themselves consisted of smaller, self-contained, multimedia content. The aim of each learning object was to explain a particular programming concept in the Java programming language through the use of examples. The main learning objects, their content and their sequencing were designed to follow a particular pedagogical method. The effectiveness of the learning resources was recognised by being one of a small number of recipients of the European Academic Software Awards in 2004.[2]

Analysing learning objects

The Java project implemented a simple sequence of 'pages' of multimedia content. Each of the pages was of a particular type, or design, and served a specific purpose. Each learning object comprised a similar sequence of pages. User control over the sequence consists of simple forward, back and rewind navigation buttons. In a typical learning object, the first page provided the title and a short description of the topic to be covered. This was followed by pages of different types from simple static text and images to pages that consisted of synchronised sequences of animations and text. The general format of each sequence was:

1. The title page.
2. The concept page.
3. The example page.
4. The detailed explanation page.
5. The practice page.

A typical sequence consisted of at least five steps employing five basic page types. In some learning objects, steps 3 and 4 were repeated with different programming examples and, often, there was more than one practice page.

Learning objects as patterns

A major aim of our work is to provide design patterns that will enable new learning resources to be created from existing successful resources. From the analysis of the learning objects used in the Java project, a pattern can be detected in the sequence of pages that make up each learning object. The individual page types can also be regarded as patterns that describe a particular learner activity. Thus the main pattern may be termed the sequencer pattern and it can be regarded as a compound pattern (i.e. one that is made up of other patterns).

Buschmann *et al.* (1996) define design patterns, in part, as describing a particular recurring design problem that arises in particular design contexts and presents a well-proven scheme for its solution. The learning object components, described here, have been extracted from existing successful learning resources and they are described in a technology neutral manner, thus they can be considered as design patterns. However, Buschmann *et al.* also suggest that patterns should be described in a consistent and structured manner in terms of their context (the situation that gives rise to the problem), the problem itself and the solution to that problem. This advice follows the lead of Alexander *et al.* (1977), who describe their patterns in a similar manner and who refer to a collection of such patterns as a pattern language. It is also important that the learning object components that are defined using these patterns fully take into account the need for reuse and adaptability. Thus these aspects need to be addressed in any documentation of a pattern. (An example of such documentation (as adapted by the authors) can be found in Appendix 1.)

The analysis of the original learning object results in the identification of a pedagogical pattern that represents a simple framework into which a series of activities can be located. The implementation of the pattern is a simple page sequencer that could be used for any simple linear sequence of activities. The pedagogy is thus not enforced; it is left up to the designer to select the appropriate activities and the correct ordering of those activities. The activities themselves are also patterns and are implemented as templates into which appropriate content can be inserted. This content can also be accompanied by metadata that help search engines find it, define its properties and indicate its level of adaptability. Based on these metadata, an automated service can be identified to adapt the learning object components or patterns to different user requirements.

Discussion: towards adaptable learning objects

Learning object metadata

The provision of metadata was not of prime concern during the initial development of the learning objects in the Java Project. However, DC metadata are typically held within a separate catalogue. Consequently, identifying the main features of the resource, its learning objectives, media content, author and other relevant details can be a second stage refinement based directly on a learning object or learning pattern description. Depending on the context in which the object is expected to be used, a range of descriptors may be indicated. In our context those descriptors that have appropriate matches to the anonymous PNPs (and which will generally appear in the newly proposed ‘adaptability’ tag) are liable to be the most relevant.

Application profiles

For the purposes of this discussion we accept the extension of the definition of accessibility beyond disability, and define the relationship between a user and a resource as accessible when the characteristics of the resource as delivered match the user’s needs and preferences (Nevile, 2005). The definition of accessibility implied here is that the relationship between the user and the resource is one that enables the user to make sensory and cognitive contact with the content of the resource (IMS, 2004). According to the ‘AccessForAll’ statement the term disability has been re-defined as a mismatch between the needs of the learner and the education offered, and it is therefore not a personal trait but an artefact of the relationship between the learner and the learning environment or education delivery (Cooper *et al.*, 2005). Accessibility, therefore, is the ability of the learning environment to adjust to the needs of all learners and is determined by the flexibility of the environment (with respect to presentation, control methods and access modality) and the availability of adequate ‘alternative-but-equivalent’ content (Heath *et al.*, 2005).

The needs and preferences of a user may arise from the context or environment the user is in, the tools available (e.g. mobile devices, assistive technologies such as Braille devices, voice recognition systems, or alternative keyboards, etc.), their background or a disability (physical, cognitive or sensory). According to the ‘AccessForAll’ vocabulary, descriptions of needs and preferences are separated into display, control and content characteristics (as already described). The user’s descriptions of their needs and preferences may change according to the context or occasion.

Accessibility service

In order to achieve an accessible relationship between the resource and the user, descriptions of user needs and preferences are checked against descriptions of resource components until they match. This process involves a description of a user’s control, display and content needs and preferences being matched with a description of the components of the learning object (Nevile *et al.*, 2005). The delivery of the appropriate component will form an accessible relationship between the user and the learning object. According to the ‘AccessForAll’ metadata overview, accessible systems should be able to adjust the user interface of the learning environment, locate needed resources and alter resources properties to match the needs and preferences of the user. This may involve the substitution, augmentation or transformation of components of the resource such as changes in sensory modality. For the purposes of this paper we will refer to an abstract transformation, augmentation and substitution service (TASS), which is geared to our specific learning object application. However, this can be viewed as a special instance of an ‘AccessForAll’ service.

An example of a replacement occurs when a user accessing a learning object requires a vision-free access to the resource, and therefore need alternatives to the visual content contained in the primary resource of the learning object. As stated previously, the profile of this user may actually be the same as the profile of a sighted user accessing the learning object on a PDA while driving: the user needs to access the learning object using non-visual techniques. For this relationship to be accessible it is necessary to replace the visual element of the learning objects with components that match the user's preferences of vision-free access. It is also often the case that the original content of the resource has to be supplemented, for example with the availability of a dictionary or captions, for an aural component.

The process of retrieving and presenting accessible learning objects employs an adaptation of the 'AccessForAll' application service or TASS. This is presented in Figure 2.

In simple terms, the TASS service is triggered by the user making a request through the learning resource catalogue to the learning object repository. The TASS service checks the catalogue for the objects accessibility element and compares it with the user's profile in the PNP repository. Appropriate transformations, augmentations and substitutions are applied either directly through the local TASS managed resources or indirectly through a global 'AccessForAll' service.

In principle, transformation, augmentation and substitution may occur at any level but would typically take place at the component or learning pattern level. Where a learning object in itself is not considered accessible, its constituent components are examined. If all individual components can be transformed, augmented or

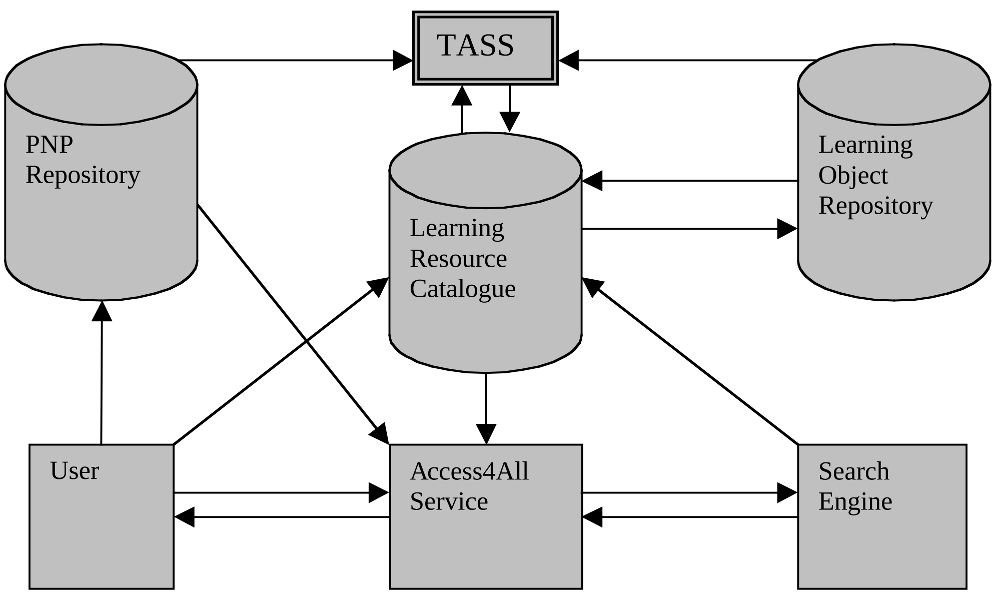

Figure 2. TASS: Transformation, Augmentation and Substitution Service

substituted then the learning object can be reconstructed as an accessible object. In certain cases, needs or preferences may require a substitute learning pattern. The learning object can then be recreated from its constituent components but employing an alternate pattern. While the primary concern is one of accessibility of resources, this feature could simply provide learners with learning patterns more suited to their personal learning styles.

Conclusions and future work

Drawing upon the work of the Java Project, the current research looks at ways to enrich and enhance these learning patterns in order to develop not only pedagogically effective, reusable but adaptable learning objects. The next stage of the research is to develop a tool to assist academics, without previous programming experience, in creating reusable, retrievable and adaptable learning objects. By incorporating these learning design patterns into a tool, an instrument can be developed to help the authors (in this case university tutors) to design and develop learning objects. This tool will also provide the opportunity for the author to assign DC metadata to the learning object in as simple a way as possible: this may include methods for intelligently inferring metadata, in order to make the process easier and quicker for the resource creator.

An interesting challenge in terms of this research is to deal with the integration of 'AccessForAll' and DC vocabularies. The development of an application profile for a set of user accessibility needs and preferences using the 'AccessForAll' vocabulary can be easily discovered using the DC records. However there is considerable disagreement on whether the PNP, which is a metadata record itself, should be integrated within the DC framework. Further work also needs to concentrate on the recursive nature of accessibility matching. The matching process (Figure 2) involves having a first 'main' resource identified and then tested against a PNP, while it would be more efficient to first decompose the resource to find an alternative to one component and to transform, augment or substitute that component, before re-assembling and re-testing the resource. Another important issue, which has yet to be resolved, is who is responsible for the substitution or augmentation service. It might be reasonable to expect a tutor employing a short video element to provide a transcript, but if a user profile expresses a preference for British Sign Language can we really expect this service to be provided by the content creator? The obvious conclusion, therefore, is that the provision of the adaptation service needs to be a collaborative venture, with additional components or alternative learning patterns provided by those with the skills to do so. In essence this can lead to the creation of a wide range of additional or alternative learning objects; some of which may be considered true alternate resources, and others as virtual objects in that they may have no physical existence but are supplied on-demand through transformations and advanced searches in response to a new PNP.

In conclusion, the newly emerging IMS and DC adaptability and accessibility standards and the proposed PNPs are set to have a profound impact on the way we view the creation of digital content as well as the way it is presented to us. This paper suggests an approach for designers of learning objects and learning patterns to respond to this challenge. Assuming that appropriate tools can be created to allow non-technical tutors to create learning objects based on these patterns, and assuming that rich metadata can be collected or inferred from those resources, then learning objects can be developed that are truly adaptable. The process of carrying out this future research and applying it to this application area will give a clearer indication to those developing the standards of whether their approach is fundamentally sound and where refinements are needed.

Notes

1. Dublin Core Metadata Initiative: http://www.dublincore.org
2. European Academic Software Awards: http://www.bth.se/llab/easa.nsf

References

Alexander, C., Ishikawa, S. & Silverstein, M. (1977) *A pattern language: towns, buildings, construction* (New York, Oxford University Press).

Boyle, T. (2003) Design principles for authoring dynamic, reusable learning objects, *Australian Journal of Educational Technology,* 19(1), 46–58.

Boyle, T., Bradley, C., Chalk, P., Jones, R. & Pickard, P. (2003) Using blended learning to improve student success rates in learning to program, *Journal of Educational Media, special edition on Blended Learning,* 28(2–3), 165–178.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M. (1996) *Pattern-oriented software architecture* (Chichester, John Wiley).

Cooper, M., Treviranus, J. & Heath, A. (2005) Meeting the diversity of needs and preferences—a look at the IMS AccessForAll specifications' role in meeting the accessibility agenda efficiently, paper presented at the *Accessible Design in the Digital World Conference.* Available online at: http://ewic.bcs.org/conference/2005/accessible/workshops/workshop3.htm (accessed 4 January 2006).

Downes, S. (2001) Learning objects: resources for distance education worldwide, *International Review of Research in Open and Distance Learning,* 2(1).

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995) *Design patterns: elements of reusable object-oriented software* (Addison-Wesley).

Goodyear, P. (2005) Educational design and networked learning: Patterns, pattern languages and design practice, *Australian Journal of Educational Technology,* 21(1), 82–101. Available online at: http://www.ascilite.org.au/ajet/ajet21/goodyear.html (accessed 4 January 2006).

Heath, A., Treviranus, J. & Nevile, L. (2005) *Individualised adaptability and accessibility in e-learning, education and training Part 1: framework.* Available online at: http://jtc1sc36.org/doc/36N1024.pdf (accessed 4 January 2006).

IMS (2004) *IMS AccessForAll Metadata specification.* Available online at: http://www.imsglobal.org/accessibility/(accessed 4 January 2006).

Jones, R. (2004) Designing adaptable learning resources with learning object patterns, *Journal of Digital Information,* 6(1). Available online at: http://jodi.tamu.edu/Articles/v06/i01/Jones/ (accessed 4 January 2006).

Nevile, L. (2005) Anonymous Dublin Core Profiles for Accessible User Relationships with Resources and Services, paper presented at the *International Conference on Dublin Core and Metadata Applications,* Madrid, Spain.

Nevile, L., Cooper, M., Heath, A., Rotherberg, M. & Teviranus, J. (2005) Learner-centred accessibility for interoperable web-based educational systems, paper presented at the *1st International Workshop on Interoperability of Web-based Education Systems, 14th International World Wide Web Conference,* Chiba, Japan. Available online at: http://l3s.de/~olmedilla/events/interopPapers/paper08.pdf (accessed 4 January 2006).

Open Archive Initiative (2004) *Open Archive Initiative metadata harvesting standard version 2.0.* Available online at: http://www.openarchives.org (accessed 4 January 2006).

Polsani, P. (2003) Use and abuse of reusable learning objects, *Journal of Digital Information,* 3(4). Available online: http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani (accessed 4 January 2006).

Appendix 1. Extracts from a 'learning objects for programming' pattern language

The following descriptions form a part of a small patterns catalogue. They are similar to the format used by Alexander *et al.* (1977) as adopted (and slightly adapted) by Goodyear (2005) but with the addition of an entry on reuse.

The detailed explanation page pattern

Context. When a new programming concept is to be learnt it is valuable to be able to see an example program, or program fragment, and to have its operation explained in an interactive way.

Problem. In order to properly illustrate a programming concept or structure it is valuable to 'walk through' an example. In a teacher-led class this is easily achieved but for self-study a dedicated resource needs to be designed that allows the learner to control the pace and progress through the 'walk-through' process.

Solution. Using a multi-frame approach, an example program, or program fragment, is displayed in one frame where its execution can be simulated by highlighting each section of the code in execution order. In a second window, the effect of the program execution is demonstrated via an animation and in a third one, an explanation of the execution is given.

The animation illustrates the code being executed by highlighting the appropriate parts of the code that are being executed. At the same time the animation is also stepped through and the text window changes at each step to describe the action of the particular line, or section, of the program code. Thus there are three parts of the page that are cycled through in synchronisation: the code, the commentary and the animation.

The learner controls the operation via a button in the explanation window: this starts the synchronised animation.

Reusabilty. (e.g. an explanation of a programming concept could be supported in two different programming languages by simply replacing the program element in one frame)

Related patterns. This pattern may be implemented by the Frameset Controller and Frameset Components.