KMi

Knowledge Media Institute

Talking About Multimedia: A Layered Design Framework

Josie Taylor, Institute of Educational Technology, OU

Tamara Sumner, Knowledge Media Institute, OU

Andrrew Law, Open University Production Center, BBC

KMI-TR-53

November 12, 1997

The Open University

Talking about Multimedia:

A Layered Design Framework

Josie Taylor
Intitute of Educational Technology
The Open University
J.Taylor@open.ac.uk

Tamara Sumner
Knowledge Media Institute
The Open University
T.Sumner@open.ac.uk

Andrew Law
Open University Production Centre
BBC
AndrewL@oupc.bbc.co.uk

These days there are numerous examples of educational multimedia; e.g., Homer [15], Virtual Microscope [39], Art Explorer [9], Encarta™ and so on. While these are all "educational multimedia", most users, multimedia designers, and educators would agree that these examples represent very different types of applications. However, most people would probably have a difficult time cogently explaining *why* they are different and, more importantly, the implications of these differences both for effective use and for future design practice.

There are obviously differences in curricula content – a system on biology versus a system on classical Greek poetry. But there are also numerous other differences, such as the types of media featured, the way the media is being used, and different pedagogical approaches. How can we gauge which systems are well-designed or appropriate for a specific group of learners? How do we compare different designs? How do we decide what to do when designing our own educational multimedia?

Part of the problem in knowing how to explain and compare differences between systems lies in the fact that they are never used in a vacuum – they are always embedded in particular learning situations. Such situations range from Saturday evening 'edutainment', to after-school computer labs, to traditional classrooms, and, of particular importance to us, to distance learning. Furthermore, a multimedia system is usually not just an isolated box on a table, but part of the larger environment of books, videos, paper and pencil, teachers, other students, and curricula; some of which may be crucial for the system's successful use. Thus, to critique a system, you need to understand its intended context of use – what it is designed to do, how learners are expected to use it, how its use fits into a broader curriculum, who the target users are, what supporting materials are needed and so on.

Here, we develop a layered analytical framework for describing critical aspects of *multimedia systems and the learning situations* in which the systems are used. The purpose of this framework is to develop a vocabulary for discussing design and educational issues that can be shared by the many different stakeholders involved in educational multimedia design and deployment (e.g., educators, designers, technologists, users). Once an application is described using this vocabulary, we can analyse and compare it with other applications in a systematic and consistent fashion. This methodology will enable us to:

- understand which types of applications are more effective in given learning situations;
- anticipate the kinds of additional support applications may need to achieve the desired educational outcomes in different kinds of learning situations;
- critically assess design options and make informed design decisions to better promote the user experiences that the educator or designer intends.

Ideally, our framework will allow *designers* to express their intentions, evaluate the design implications of their decisions, and understand the strengths and limitations of their approach. It will enable *educators* to assess, prior to deployment, the best uses of different types of applications and what types of additional support might be required. It will help *users* to describe their experiences with different applications, to anticipate what is possible with different applications, and to be more informed consumers of educational multimedia.

In the remainder of this report, we describe and illustrate our framework using a detailed analysis of the Galapagos Pilot project. We will also use the framework to illustrate important design issues to be considered when creating new educational multimedia.

A LAYERED FRAMEWORK

How might a framework for describing educational multimedia be composed? Not surprisingly, some levels will be necessary for describing computer system-level issues; i.e. how media resources are stored, organised and delivered to learners. Other levels will be needed for describing educational considerations; i.e., what range of learning tasks are possible and what support is provided for accomplishing these tasks.

We propose seven basic categories for describing and analysing educational multimedia applications – delivery and use platforms, software issues, resource organisation, task syntax, task semantics, teaching strategies and educational aims (see Table 1). We have organised these categories into layers to emphasise how design decisions made at lower levels affect what is possible at upper levels and vice versa. This mutual inter-dependency between levels will be discussed further at the end of this report.

Table 1: Layered Dimensions of Multimedia

Layer Name	Definition and Examples
1. Educational Aims	What is being learned? Aims can reflect: • Curriculum objectives, • Assessment requirements, • Needs of target population.
2. Teaching Strategies	High-level strategies for achieving the educational aims, including: • Range of tasks learners can engage in, • Pedagogical approach, • Important relationships to other learning materials and objectives.
3. Task Semantics	Refers to the knowledge and skills learners need to turn high-level task goals into achievable objectives. Educational systems may contain methods, techniques, and structuring devices to support learners in understanding and enacting the task semantics. These can include: • Motivational aids, • Task decompositions, • Narrative, • Self-assessment questions.
4. Task Syntax	Means by which learners interact with the system to enact task semantics. Key aspects of systems are: • Navigation support, • User interface affordances.
5. Resource Organisation	How media resources are organised in the system. Some examples include: • Flat organisations, • Highly interconnected organisations, • Hierarchical organisations.
6. Software Issues	Critical features of the software system such as: • Programming Environments, • Resource Formats.
7. Delivery and Use Platforms	Important aspects of the learner's computational hardware. This can include: • hard disk and memory expectations, • network access, • accessories such as CD-ROM.

Generally speaking, the lower three levels (platforms, software, resource organisations) describe system-level media management issues and the upper levels (syntax, semantics, strategies and educational aims) describe how media can be used to provide different forms of educational support. In practice, as we shall see in our examples later on, not all layers are present in every application and the distinctions between layers can be blurred. In this report, we focus on the upper four levels - the ones which users or learners can manipulate, and ones which relate directly to the tasks users are engaged in. However, in order to define the framework and set the stage for later discussions, we need to say a few words about each level.

Level 1: Educational Aims

The first question to ask when analysing educational systems is "what is being learned?" Broadly speaking, educational aims can reflect a variety of sources such as personal learning goals, institutional or professional curriculum requirements, and socially accepted practices in a particular field [5, 17, 29]. For our purposes, we define educational systems (as opposed to ones which entertain, edutain or infotain) as systems which support the curriculum within which learning is to take place (whether for children or adults). This is true even in cases where the system allows users the freedom to construct their own learning agendas and follow their own regimes.

While we acknowledge that humans are sufficiently well-equipped as learners that they can find educative experience using a stick and pile of sand, a sophistication we would like to introduce is that users of multimedia systems should not feel that they are indeed working with a stick and a pile of sand. Educative systems should contain substantial coherent content which can be accessed in principled ways corresponding to good educational practice. This doesn't mean they can't be fun to use, and entertaining and so on. Other non-educational systems (sic) may be extremely edifying in many ways. But frequently such systems are not expecting the learning that takes place to be assessed, whereas we are.

As with any software system, it is critical to understand *who* the intended users are. At this level of analysis, the goal is to succinctly describe the anticipated background skills, knowledge and special needs of the target user population. Such a description is a vital first step towards a user-centred design approach [22]; i.e., one which is motivated by the needs of specific users rather than technical possibilities. Soloway notes that learners have additional special needs and proposes a design framework called "learner-centred design" with four distinctions – context, tasks, tools, and interface [32]. The layered framework we are defining here is also a learner-centred design framework, but one that is targeted at the special needs of distance education. Note that issues considered at this level of analysis may or may not be part of the system – this point will be taken up and expanded later.

Level 2: Teaching Strategies

The next level in the framework describes high-level teaching strategies for achieving the educational aims. Most teachers know what they want their learners to do and have devised a programme of work based on the curriculum. Furthermore, the teacher or instructor will (should) understand any given task in context – i.e., its relationship to curriculum goals, its relationship to other domains, probable background knowledge of the students, what the basic structure of the task is, and the sorts of study materials which are relevant to the task. In traditional classrooms, a central role of the teacher is to bridge the gap between broad educational aims and the day-to-day activities of individual students [28]. In classroom settings, students helping each other is also an important source of 'bridging' support.

In a distance education setting, teachers and other students are not co-present on a daily basis, so this bridging support must be embedded in the educational materials. The role of levels 2 and 3 in our framework is to analyse whether a particular educational system effectively provides this bridging support or if *additional* learning materials are needed. At this level of analysis, the objective is to concisely describe the teaching strategies or overall plan of the system. Key parts of a plan include: (1) the range of activities that learners can engage in, (2) the pedagogical stance taken by these activities, and (3) efforts to relate the activities to other learning materials or fields of study. Let's consider each of these parts in turn.

It is currently believed that *what learners must do* in order to understand is at least as important as the design of the instructional materials; i.e. how the materials are organised, explained, and presented [13]. At this level, we need to note what high-level activities or tasks are present in the system. Are learners watching nature videos, listening to musical selections, comparing two law cases, taking field notes, analysing specimens, or solving mathematical problems? Specifically, we need to focus on verbs that accurately characterise what learners are actually doing when using the system.

It is also important to characterise the pedagogical approach to these activities. For instance, there are many ways to approach the activity of solving mathematical problems. Many computer-aided learning systems are rooted in behaviourism and contain instructional models based on knowledge transmission and drill-and-practice repetition [4]. That is, students are presented with small chunks of information in a pre-determined order and the system helps them to practise skills such as multiplication or addition before proceeding to the next chunk [2]. Such systems may be appropriate for helping learners gain proficiency and automaticity in skills such as multiplication tables but are inappropriate if the educational aims include teaching independent problem-solving skills.

Other educational systems are broadly based on constructivism [24, 26, 33]. These systems assume learning is the result of active participation on the part of the learner in settings which resemble the settings where the skill will later be used [27]. The LOGO language is perhaps the most well-known constructionist system for teaching mathematics. According to Papert, the inventor of LOGO, the language enables students to actively explore and directly experience their mathematical ideas by constructing interactive programs in areas that are personally meaningful and interesting to them [24, 25].

Finally, many newly developed educational systems, particularly those coming out of research labs, are influenced by social-constructivism [32, 38]. Social-constructivists view learning as a process of acquiring the shared practices of a scholarly community and assume that much learning will occur indirectly through a learner's enculturation and participation in the joint efforts of social groups. An example in a field related to mathematics is the use of the PuckLand system by small groups of students studying beginning physics [40].

The purpose behind this brief survey is not to provide a comprehensive explanation of theoretical models of instruction. Rather, our intention is to remind readers that *all* educational systems are rooted in a particular model, either explicitly or implicitly by design. It is important to be clear about what model you are getting when buying or designing educational systems because each model makes different demands on the supporting materials or learning situation. For instance, the LOGO system has been deployed around the world with widely differing results. However, Harel has shown that through careful design of the learning situation to create a rich social support environment, the language can be effectively used to teach fractions to fourth grade students (nine and ten year olds) [12].

Finally, when analysing teaching strategies in a system, efforts to relate the material being studied to other learning materials or fields of study should be characterised. At a practical level, if systems refer to other materials such as books or videos, then students need access to these or substitute materials. If systems systematically refer to related fields that students may have no knowledge of, then supporting materials providing this information may be needed.

At a theoretical level, it is important to make connections across different domains and fields of study. First, such interdisciplinary connections help learners to see the broader context and importance of their often highly focused studies. Second, recognising similarities and differences across domains is an important step towards generalising learned skills and knowledge; creating these abstract categories and representations and being able to apply them to new situations is the goal of most scholarly endeavours [15]. However, as Detterman has so eloquently argued, this transfer of knowledge across domains doesn’t just take place effortlessly and automatically [8]. If systems do not explicitly make these connections, then other learning materials need to. As an example, establishing these discipline-spanning connections is the goal of the new Saturday morning Open University television format where educational programmes from different faculties are shown together and woven into a common theme.

Before defining the remaining layers of the framework, let’s examine how these upper layers are realised in a specific educational system – the Galapagos Pilot CD-ROM.

Case: The Galapagos Pilot - Levels 1 and 2

The Galapagos Pilot Tutorial project is funded by the Open University Science Faculty to examine the processes of multimedia production for science. One of the designers of the multimedia package created an extended tutorial to help his students with basic ideas in evolution. The structure he developed for tutorials was used as the basis of the multimedia package’s design. The discussion here focuses on the Pilot project which underwent developmental testing [37]. Based on experiences gained by this Pilot, the package is currently being redesigned for use in the S103 science foundation course.

Figure 1: Stunning imagery woven together with a tight narrative forms the core of the Galapagos Pilot project.

Level 1: What are the educational aims?
The aims of this package are two-fold. First, its primary aim is to introduce students to some of the observation and note-taking skills required in fieldwork, relating this activity to that engaged in by Darwin in his exploration of Galapagos. The students' activities take place against the stunning visual background of the islands (Figure 1) which animals illustrate so vividly the processes which students are studying. Students are encouraged to relate their observations to evolutionary theory, but the teaching of that theory is not the main focus of the program. These aims are not overtly articulated within the program; however, other materials in the course provide the required context and prior knowledge.

The second aim of this package is to gently introduce entry-level undergraduate students both to academic study and to using computer technology. The target audience is several thousand students in the Foundation Level science course at the Open University. These students differ widely in their background knowledge and previous academic experience. Many students may need very explicit support to re-introduce them to academic study. Additionally, most students in the course will have little or no previous experience with computer technology.

Level 2: What are the Teaching Strategies?
The students' main tasks are to (1) observe and categorise the finch species on the islands and (2) relate their findings to Darwinian evolutionary theory in order to draw some conclusions about how finches

arrived in the archipelago and later diversified. To accomplish this task, students:

- listen to narrative about the islands while viewing still images,
- watch videos of expert commentary on important sub-topics,
- enter data into a spreadsheet representing their 'field' notes,
- compute simple statistics from their data using the spreadsheet,
- closely examine pictures of finches,
- read essays on related background material.

Keeping with the 'gentle introduction aim', the overall task is relatively short and highly decomposed and structured. The package explicitly tries to get students to relate their activities to the historical context of Darwin's time period by including an essay on 'religion and evolution.' The teaching tradition behind the Pilot lies between behaviourism and constructivism– students are using skills in a simulated environment that resembles a field trip, and they must draw together their findings to reach conclusions about various issues presented to them. To this extent, the system is constructionist in its approach. But at the same time, students are relatively constrained by a tight structure which provides security and performs a hand-holding function. The tasks and sub-tasks are set and there is implicit order in their presentation. To this extent, the system also embodies behaviourist principles.

Level 3: Task Semantics

The terms syntax and semantics typically refer to language, and an important part of language is the interplay between meaning (semantics) and form (syntax). Similarly, we use these terms to emphasise that the process of understanding what to do, and how to do it, is not a one-way event. There is a continuous interaction between the system and the learner; learners try to articulate a task structure reflecting their prior knowledge and experience, and the system itself prompts learners to act in particular ways. Often, learners are not even aware that this dialectical process is going on.

We use the term 'task semantics' to refer to the implicit (or sometimes very explicit) knowledge learners use to understand tasks; that is, the knowledge and skills necessary to turn high-level goals into achievable objectives, to define boundaries and gauge when a task is sufficiently complete, to link important features of the domain to one another, and so on. Our analyses at this level focus on two things: (1) articulating the knowledge and skills learners need in order to understand *what* to do and (2) describing how the system or other aspects of the learning situation supports learners to enact and understand the task semantics.

The task semantics learners need to know reflect important properties of the domain or field being studied. These properties can include critical perceptual and conceptual skills for recognising formal arrangements [9], procedural skills for decomposing tasks and taking actions [30], problem-solving techniques and strategies [1], a repertoire of previous cases to compare with [14, 21], socially-accepted practices in the particular field [5, 17], and important declarative knowledge. Understanding the knowledge and skills learners need in a given situation requires knowledge of the domain or field being studied, experience in instructional design, and some understanding of the principles of cognition and learning. A full treatment of this topic is the goal of many undergraduate and post-graduate education programmes and is thus beyond the scope of this report.

Educational systems can play a useful role in helping learners construct a coherent task semantic; at the very least, systems should not hinder this process. Unfortunately, while teachers are usually (though not always!) aware of the limitations of learners' abilities to generate sensible task semantics, multimedia designers frequently over-estimate them. Often, learners lack vital support for activities essential to task completion or support for recognising and understanding key aspects of what they are doing. At this level of analysis, the objective is to assess whether the system provides the necessary support for the target learner population and to characterise at a high-level the support methods, techniques, or devices used by the educational system. Here, we will consider three useful categories for analysing system support – (1) techniques for promoting reflection, (2) structuring devices helping learners through tasks, and (3) methods for keeping learners motivated.

Promoting Reflection

Simply put, 'promoting reflection' means providing learners with the means and opportunity to critically think about what they are doing and what are the important points. Students will often lack the necessary knowledge and skills for recognising important issues or errors in their thinking, and will often miss important opportunities for reflection [9, 11, 15, 34]. Studies have shown that one difference between 'good' students and 'poor' students is the quality and quantity of self-reflection the students engage in [6]. Good students are continually thinking about and assessing the appropriateness of their approach and the validity of their actions and results; 'poor' students tend to muddle through activities without recognising key issues or where they went astray. Systems and educational materials that promote reflection should help *all* students to: (1) potential flaws in their approach or important points to consider, (2) identify critical knowledge or actions relevant to the task, and (3) apply this knowledge or skills to the task at hand.

In more traditional educational media such as textbooks, a common method for promoting reflection is self assessment questions (SAQs). Typically, SAQs are located throughout the text at key locations to get students to stop and reflect on what they have just read. Designing effective self assessment questions is a challenging task and helpful discussions on how to proceed can be found in [19]. Borrowing from existing media, many educational multimedia systems also employ the SAQ approach. Some extend this approach by checking student responses for correctness and reminding students about questions they have not yet answered.

Other systems have tried more novel approaches that take better advantage of the potential of new media [36]. For instance, in the Art Explorer system [9], students are asked to group paintings several times into different categories of their own choosing. The system analyses the student's grouping to detect which paintings the student perceives as being similar or different. The system presents these results back to the student (see Figure 2), shows how other students grouped paintings, and asks them try again to generate more categories. The aim of presenting the analyses back to students is to get them to question and rethink their original categories.

Figure 2: The Art Explorer system encourages self-reflection by analysing and presenting the student's own categories back to them.

As the examples above illustrate, there are many different ways for educational multimedia to promote learner reflection. When analysing a specific system, focus on the explicit mechanisms, if any, embedded in the system. Specifically, analyse the mechanisms in terms of *what* and *how*. What skills and knowledge does the system promote reflection on, and do they correspond with key points of the task semantic? How does the system promote reflection? Does the system encourage self-reflection; i.e., get learners to re-examine their efforts from a new perspective or in light of new information?

Structuring Devices

A large part of instructional design has traditionally involved careful consideration of how information should be broken up, organised, and presented to the learner. This emphasis on structuring arose from the dominant educational model (knowledge transmission), a model supported by, and recapitulated in, serial modes of presentation in existing teaching media (i.e., written books, spoken lectures, audio tapes, videotapes). Using these media in 'transmission' mode, the challenge for designers was taking complexly structured and interrelated information and rendering it into a meaningful, serial form.

With the advent of hypermedia, we are no longer confined to creating educational materials for serial presentation. Indeed, multimedia systems can now reflect the structures and interconnections important to a particular field or domain [10]. Such 'hypermedia' systems can support learners to follow links to related material or even to construct their own personally-meaningful links [3]. However, with all this new found freedom, there is a cost: one of the pressing design challenges is ensuring that learners don't 'get lost in hyperspace'. That is, systems should help learners in: (1) finding relevant information, (2) sense-making (constructing a meaningful 'story' from their explorations), and (3) enacting (knowing what to do next in the face of seemingly infinite options).

Here, we use the term 'structuring devices' to refer to techniques employed by systems to help learners in finding, sense-making, and enacting. In the audio-visual arts, narrative is a common structuring device used to help listeners or viewers weave a sensible story out of complex information. In the realm of multimedia, the MENO project [16] is exploring the many ways in which narrative can usefully underpin learning activites. Maps and task decompositions are other useful structuring devices. Maps, for instance, can present an overview of the concepts being covered or the resources present in the system. Concept maps can help learners get a sense of the larger-picture of what is being studied and help to contextualise individual activities [31]. Maps can also give learners a Gestalt view of what is covered by the system, and thus assist them in understanding where they been in the system and where they have missed. For instance, the map of the Galapagos archipelago shown in Figure 3 acts as a navigation interface and a guide to the resources that are available to be explored. Task decompositions are basically 'to do' lists and come in many forms, such as checklists [18] and agendas [35, 41]. The Galapagos system also employs a form of task decomposition in its help system (see Figure 6).

Some systems, such as general purpose multimedia encyclopaedias, contain few structuring devices. Other systems, such as the Galapagos Pilot, use several devices to help novice students accomplish complex tasks. When analysing systems at this level of the framework, note the various structuring devices employed by the system and try to determine if they will be adequate to support the target learner population. Will students be able to find the information they need? If students are stuck, how will they find out what to do next? If the necessary structuring devices are not present in the system (which is often the case), supporting materials may need to be constructed to fill this gap.

Motivation

As Soloway points out, one difference between educational systems and professional productivity software is that "the student's initial interest and continuing engagement cannot be taken for granted" [32]. Csikszentmihalyi expresses this concern a little more strongly, claiming that the chief impediments to learning are not cognitive, but stem from students' lack of motivation and lack of joy from learning [7]. Unfortunately, how to motivate learners is a complex and poorly understood topic.

Constructionists claim that students are motivated when they work on authentic problems of their own choosing [25, 32]. Others point out the need for fun in learning [9] and have even analysed computer games as a source of inspiration on motivational techniques [20]. For challenging tasks, providing pacing and completion information can be critical – instilling students with a sense of progress and accomplishment can help mitigate frustration and ensure their continued engagement.

Despite a paucity of recognised techniques for motivating learners, it is still important to consider what motivational support is provided by the system. Does the system motivate learners to begin a task or does it overwhelm them from the very start? Are there aspects of the system that encourage learners to stick with challenging tasks or feel good about their accomplishments so far?

You may be wondering why level three warrants such extensive treatment compared with the other levels. In our experience, providing explicit support for task semantics is vitally important, yet all-too-frequently overlooked in multimedia design. Educators tend to focus on their area of expertise, articulating educational aims, teaching strategies, and task semantics. Multimedia designers tend to focus on their system-oriented areas of expertise – user interface design, resource organisation, and hardware and software issues. It is in this vital area of providing support for task semantics that the two cultures meet and often fail to communicate during system design. Hopefully, the previous discussion has shed some light on how this gap can be bridged in educational systems. Before continuing with the next level, let's examine the task semantics in the Galapagos Pilot.

Case: The Galapagos Pilot - Level 3

What are the task semantics?
Students are asked by the program to visit each island and note the finch species to be found on each one along with details of habitat and feeding habits. Students are posed questions about how the finches could have arrived in Galapagos which can only be answered by inspection of all the available resources (expert comment, essays, animations and related evidence in video clips). The CD-ROM supports a tutor-marked assignment where the ultimate goal is to derive a theory of the origins of the finch species. Specifically, students must write an essay answering the question: did one species of finch arrive on the islands (does evolution alone account for the diversity of observed species) or did multiple species arrive initially?

How are the task semantics supported in the system?
Students do not have to derive their own task semantic or consult external resources to complete the task. The task semantics are conveyed explicitly as part of the system and all necessary resources are delivered within the system.

The program has been structured episodically – periods of time are spent watching, then doing, then watching again, and so on. The teaching sequence consists of four main interlocking segments as follows:

Introduction:
The system begins with contextual material delivered on video clip by the course tutor. The tutor briefly discusses why Madagascar and Galapagos are both unique island communities and how each are

prototypical examples of particular forms of island ecosystems. Selecting Galapagos begins the following sequence of activities:

Episode 1:
Looking: The sequence opens with some motivational material: stunning pictures show the beauty of the islands and the diversity of wildlife is highlighted. These pictures are set against voice-overs quoting from Darwin's journals. Understanding difficult nineteenth century language is made easier by simultaneous on-screen text display as the actor speaks. Against a video collage of various animals, we hear and feel Darwin's amazement when faced with the wildlife of Galapagos, and with the gradation and diversity of structure of the finch population in particular. The commentary introduces the idea of evolution.

Doing: The next segment provides information about the geographical location of Galapagos, and the commentary advises students to take a tour around the islands and collect information about the finch species, habitat and feeding habits on a spreadsheet. Selecting an island from the map in Figure 3 causes the necessary information about finches to be displayed in a pop-up window.

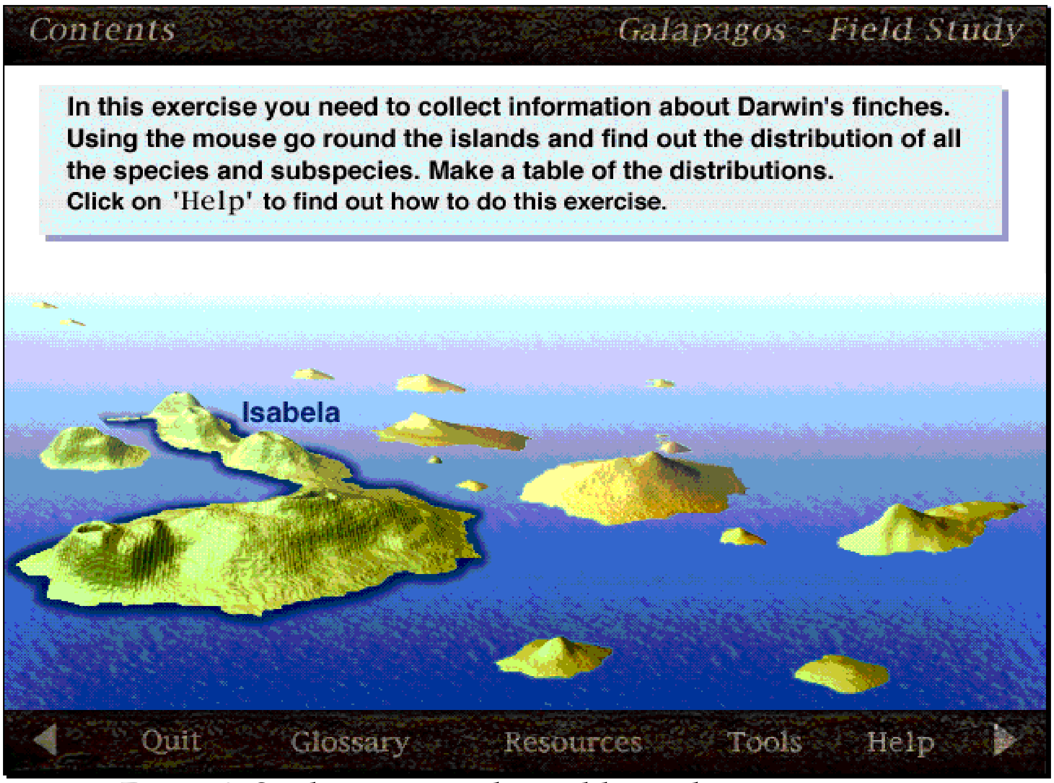

Figure 3: Students can explore additional resources concerning the finch population by selecting any island on the map. In this case, the Isabela Island has been selected.

Episode 2:
Looking: Having completed this exercise, students then sit back to look at video footage about the formation of the islands showing spectacular volcanic sequences, graphics of submerged volcanic processes, how the

islands have arisen from the sea and what they would have looked like prior to being colonised by plant and animal life.

<u>Doing:</u> The question is posed first by the spoken commentary: how did the finches get there? Students are directed to several video clips of expert witnesses (discussing prevailing winds and ocean currents, and useful information from other related research), a short essay on finches, and a world map. Successive screens then pose a series of questions which students answer by selecting from multiple choice response boxes.

Episode 3:
<u>Looking:</u> Encouragingly, the narrative line now takes over again and talks about how Darwin made mistakes when he was a young field biologist. Further stunning photo-images of birds are shown as a prelude to the task which faced John Gould, the person who actually recognised the birds as finches. From the tiny shrivelled scraps of field specimens Darwin brought back to England, Gould produced impressive lifelike paintings which we are shown.

<u>Doing:</u> Students are invited to 'have a go' at Gould's task and group the birds by dragging pictures of Darwin's specimens into piles. A video of an expert on classifying birds is available to help with this task. When students are done with their grouping, they are shown Gould's original classification, and then a much more recent one – Lack's tree. Students are able to view their own original grouping by clicking back. Students are invited to generate hypotheses about why the groupings are different and to speculate on what information the later studies had which the earlier ones didn't aided by a video clip talking about the importance of field work, and an essay on Religion and Evolution. The next section asks students to use the data from their spreadsheets to complete a table (Figure 4) from which they can identify the total number of species/subspecies, and count up how many are endemic to each island group.

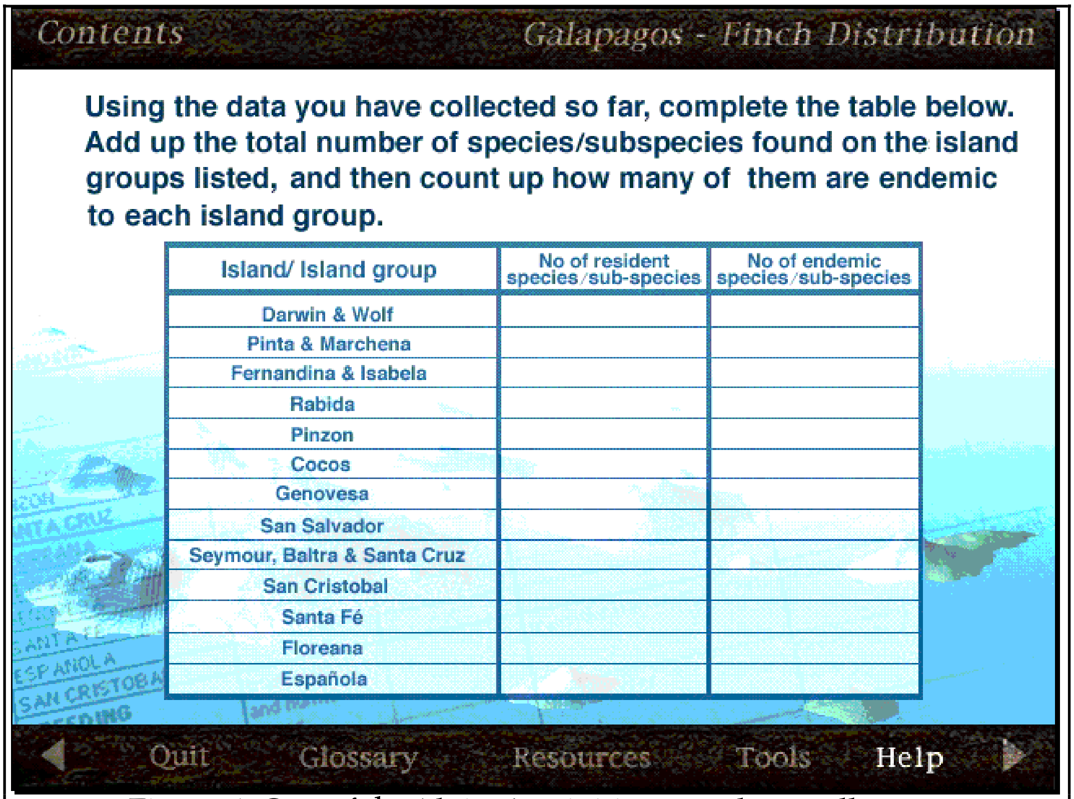

Figure 4: One of the 'doing' activities – students tally data from their spreadsheet exercise and enter the figures into this table.

Episode 4:
The task now becomes working out percentages of sub-species endemic to the island group.

How does the system promote reflection?
Galapagos relies primarily on SAQs at the end of 'looking' segments in an effort to get student's to think about what they have just seen and heard. Each SAQ is answered by choosing from a selection of multiple choice boxes. The system discourages users from advancing without answering the questions. A few SAQs asking more difficult and reflective questions are posed in advance of the material. The classification task (see Figure 5) also attempts to encourage reflection by getting students to compare their groupings with the experts of previous eras and to reflect on why the groupings may be different.

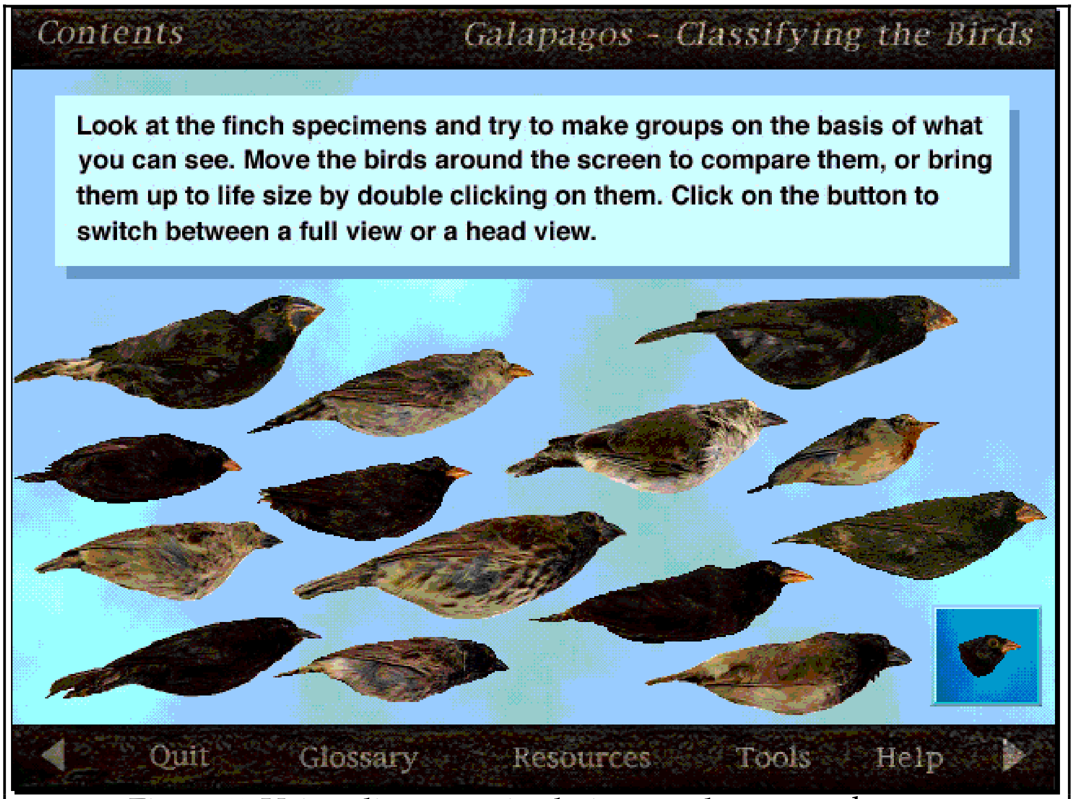

Figure 5: Using direct manipulation, students sort the finches into groups representing those likely to be the same species.

What structuring devices are used by the system?

Narrative is the primary device used by the system, though maps and task decompositions are also used in smaller amounts. The system presents itself in such a way that its televisual heritage is obvious. There is a strong narrative flow at the points of watching, and the activities are highly structured. Keeping a strong narrative line and presenting very clear task instructions enables the target learners to stay on track, to not get lost in the system, and to accomplish the tasks in sequence with minimal distraction.

At the end of these narrative sequences, learners are given a task to perform. At these points, there are some opportunities for browsing and branching, particularly in looking around the available resources. In some cases, maps are used to indicate the possible areas and resources to explore (see Figure 3). If the learner gets stuck on how to proceed, detailed task instructions (see Figure 6) are available by pressing help.

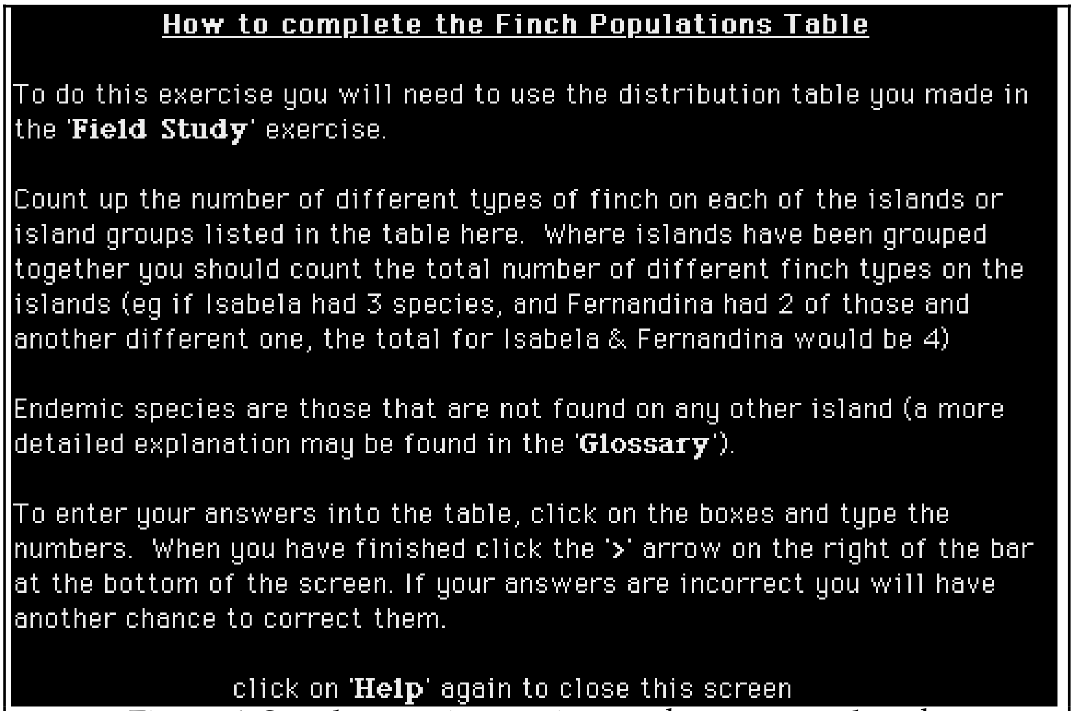

Figure 6: Step-by-step instructions on how to complete the table shown in Figure 4 are provided in the help system.

What motivational techniques does the system employ?
The visual delights of the islands are used as motivational ploys to keep students going, while also communicating the fact that simple observation of wildlife can be misleading unless a systematic approach to cataloguing data is taken. Students are given encouragement through periodic reminders of Darwin's many mistakes in his early attempts at fieldwork.

Level 4: Task Syntax

The task syntax level is the one at which the user is envisaged as operating the multimedia system – typically learners would be engaged in exploring the system to discover ways of actually achieving the task goals set up in the previous level. Essentially, this level is the appropriate one for critique of the user interface. Here, we will focus on two main aspects of the interface – *navigation* support and *other affordances* supporting the task semantics. The term 'affordances' encompasses both the *actual* properties of a particular technology that determine how it can be used and the *perceived* properties that influence how people will actually use it [23]. Clearly, in our case, it is desirable for interfaces of educational systems to help learners in perceiving appropriate task semantics. Thus, when designing or critiquing educational systems, the task semantics and task syntax levels are often deeply intertwined.

Navigational Support

In most multimedia systems, the interface furnishes users with a representation of the resources which are available, and provides mechanisms for searching, browsing, collating, inspecting, annotating, or watching the multiple forms of media. In so doing, the provided affordances encourage users to behave in certain kinds of ways and hopefully these correspond to the activities designers were expecting users to engage in.

A large category of interface mechanisms focus on supporting navigation, i.e., finding and moving through resources via searching, browsing, or linking. For example, in an on-line encyclopaedia the typical task is locating specific information in a large information space. Thus, an encyclopaedic format encourages and supports rich forms of searching and perhaps limited forms of browsing to help find related information. Other interface mechanisms are more task or domain-specific. A virtual instrument, for instance, offers very task-specific affordances. The Virtual Microscope [39] supports the close examination of a small number of rock samples in different kinds of polarised and unpolarised light. In this case, obviously, a battery of tools to enable complex searches of large databases is irrelevant. Designers, therefore, need to know what kind of activity is the principal focus in a given system and should provide interface support appropriate to this activity. Although a variety of other activities could also be enabled, they should be regarded as subsidiary to the main function.

Given the diversity of possibilities in a user interface, it is impossible to offer a complete and well-defined analysis checklist. Indeed, this is an area offering the potential for unlimited design creativity. However, some useful things to focus on are the types of navigation supported in a system and types of representations of the resources provided by the system. That is, does the system provide the user with an overview of available resources? Does the system indicate to the user where he or she has been and what still remains to be discovered? Does the system support the user in developing a coherent structural overview of the system? And, can users find the resources they need at appropriate points in their tasks?

Other Affordances Supporting the Task Semantics

The framework we are describing can help in the design process by making clear the close relationship between task syntax and task semantics. Decisions made at the task semantics level about the way in which students will go about a given task using the system have repercussions for design at the syntax level. Problems often arise when these repercussions are not explicitly described or acknowledged. For example, suppose we are designing a system in which students are expected to develop independent learning skills, where they make their own decisions about how to proceed with a task given only the resources with which to complete it. In this case, clearly, the interface needs to provide them with all the tools required to complete the task. However, since developing learning skills is also an objective, the system should also provide support tools for the decision making process learners will be engaged in (e.g. tools to enable them to proceed, go back, interrupt, re-start, make notes, insert electronic bookmarks, or tags, and so on). In this case, the resulting interface will need to give users access to and control over these various features and yet be carefully designed so as not to intimidate novices. On the other hand, let's assume that our students already have the necessary skills for independent learning. In this case, they should just be able to get on with the task, with little or no interference from the system. These learners will not be expecting the system necessarily to 'do' anything specific - they will approach it with a plan of action and all the affordances for independent learning are there. However, a novice user might find such as system's lack of interactive engagement puzzling, difficult to overcome and ultimately daunting.

Let's suppose in another case that we are designing for a group of learners who are not so confident, and who are new to using computers (as in the Galapagos Pilot scenario). The decision made at the task semantics level is that the material is going to be packaged and presented within a supporting framework that has a strong narrative thread running through it. In this case, the system will need to take the initiative in certain circumstances and designers need to ensure that users are not left looking at 'dead' screens with no idea of what to do next, or where to go. The system needs to deliver the right materials at the right time and to actively elicit the user's engagement with it. Learners in this situation will not need to avail themselves of a complex array of tools as did the independent learners, because the decision-making process has been done elsewhere. In this system, the needs for the interface shifts to focusing on motivation, entrancement, specific subject matter, or particular teaching points. This in turn means that the screen organisation must be simplified and designed so as not to be intimidating to a new user. The affordances in this case will be different, but compatible with the planned task semantics so that students don't end up frustrated at not being able to complete their tasks. Our independent learners might find such a streamlined presentation irritating to use, however, because they might want to break out of the narrative line and do their own thing.

So it is clearly crucial that the relationship between task semantics and task syntax is well articulated in the design process, and that careful consideration is given to the target user population's needs and expectations.

Level 5: Resource Organisation

All computational systems have formal structure at the machine level – data or information in computers is stored in machine code, and this code is successively interpreted by the computer to produce a virtual machine. Virtual machines provide higher-level, abstract representations that enable us, as users or designers, to recognise and manipulate objects such as text, video clips, audio tracks, pictures etc. These 'resources' are the basic tools of multimedia.

At this level in the framework, we are concerned with understanding how resources are organised in the system. Ideally, resources are organised intelligently to maximise the kind of interface we want our multimedia system to have, to optimise preferred routes through the material and to help users orient themselves to the resources available. Specifically, when analysing a system at this level, think about drawing an overview picture or map of the system. What shape would this picture be? How are different resources grouped and connected? Can users even generate a coherent representation of the system ?

As the last question indicates, we want to take a user-centred perspective on organisation rather than a developer-centred perspective because the two are not always the same. That is, how a designer wants users to perceive the structure of the system is usually different from how the system is actually implemented. When trying to answer the questions posed above, think about what an experienced user would draw rather than what one of the system implementors would draw. This distinction between representations which are system-oriented and those which are matters of interpretation is an important one.

Typical multimedia development teams are multidisciplinary, consisting of educators, domain specialists, graphic designers, and technologists. Various members are likely to have differing views on the relative importance of these perspectives. This can give rise to conversations where team members are all using a language to describe what they are doing which has only apparent consensus – words such as 'structure', 'organisation', 'access', 'file' and so on may tacitly be taken by all team members to refer to the same thing, but in fact are underpinned by quite different concepts.

Case: The Galapagos Pilot - Levels 4 and 5

Level 4: Task syntax support

In keeping with the design goal of a simple narrative structure, the range of possible user actions is very limited. During “looking” intervals, users can control the rate of progress through the presentation by clicking on the forward arrow button (). Pressing the backward button ()jumps back to beginning of the episode. When users reach the ‘doing’ part of the episode, there is the potential for more exploratory activity. The ‘doing’ sessions either launch a spreadsheet package (Claris Works™) or provide information stores within Galapagos for students to explore. Additional materials supporting the doing activity can be accessed via the toolbar shown below (Figure 7). The archipelago map in Figure 3 is an example of a nice navigation mechanism that also directly supports the task semantics.

Figure 7: This toolbar is available during all ‘doing’ activities.

Level 5: Resource Organisation

The structure of the Galapagos Pilot is hierarchical. Each rectangle in Figure 8 represents a page in the system; each page may contain still pictures, video clips, and of course audio clips of the ongoing narrative. The diagram as a whole represents a typical page organisation within an episode. As you can see, the pages are organised in a linear fashion during ‘looking’ sessions, with a small amount of branching at the end of an episode during the ‘doing’ part. There are relatively few connections between the pages. The entire Pilot system consists of four such episodes that are themselves linearly sequenced. Thus, jumping to material in the middle of the third episode requires quite a lot of button pressing to progress through the first two episodes!

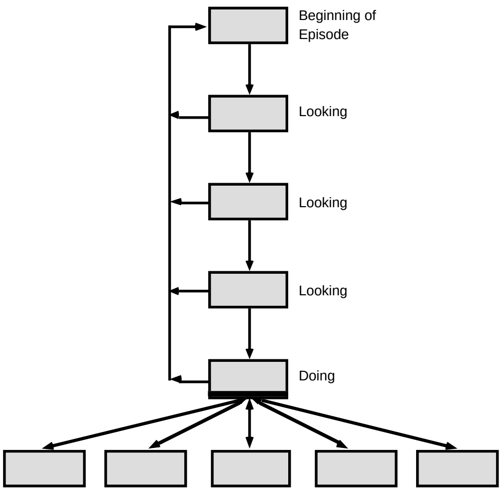

Figure 8: This diagram conveys a general picture of the resource organization in the Pilot project, including key navigational links built into the system

Level 6: Software Issues

The next two levels span a potentially broad range of technical issues. As we shall see, in an analogous way to the semantics and syntax levels, the software and hardware issues covered in levels 6 and 7 can be complexly intertwined. Software issues do not play a large role in analysing existing systems, mainly because decisions made by designers get reflected in the hardware requirements for using the system; i.e., statements about the memory or disk space needed. However, software issues play a crucial role in designing and creating multimedia systems. Here, we will briefly discuss factors affecting one of the biggest software issues facing any design team – the choice of programming development environment.

The programming environment chosen for building a system may seem to be a very low-level decision, but it has a big influence on what the final system can be, primarily because not all environments are well-suited for all types of applications. Some environments, such as Director™, Toolbook™ and mTropolis™, are better suited for more presentational forms of multimedia. Other more general purpose programming languages, such as C++ or Visual Basic™, may be better suited for creating more productivity tool-oriented applications. Also, if users are interacting with a shared service over a network, then networking languages such as Java™ may be a better choice. As with any over-generalisation, there are many exceptions to these broad heuristics. The Virtual Microscope blurs the presentation /productivity distinction, yet it was created using Director™. A streamlined version of the Microscope was then created for use over the Internet by taking advantage of the Shockwave™ plug-in that enables Director applications to run in most standard Internet browsers.

Other factors also influence the choice of programming environment. These can include ease and speed of implementation, previous experience and know-how of the development team, and hardware platform suitability. For instance, we have found Visual Basic to be a useful tool for quickly prototyping our ideas for new systems because it provides built-in support for many media types and because our technical expert is very familiar with it. However, the resulting systems have usually required large amounts of memory to have adequate performance – more memory than we can reasonably expect our students to have. Thus, we have used Visual Basic for prototyping and then have implemented the final system using a different environment. As this story indicates, choosing suitable environments involves a complex series of design trade-offs and different environments may come into play at different points in the development lifecycle. Choosing a programming environment for multimedia systems is particularly challenging since this is a constantly changing and rapidly expanding area.

Level 7: Delivery and Use Platforms

When we look into the computer, inside the structure of databases and hypertext files and so on, the languages we see are governed by the constraints of hardware. Issues of compatibility, accessibility, performance and space occupied in memory are concepts which are machine related, have formal properties and are (to all intents and purposes) non-negotiable. When analysing existing systems, the key question to ask is will the system run adequately on the typical user's hardware platform? That is, will student's have the necessary hard disk space? Will their machines contain enough memory to ensure adequate performance? Does the system assume the existence of accessories that the student may not have; e.g. CD-ROM drives, colour printers, or high-speed modems?

When creating multimedia systems, the challenge is to design an interesting and educative system that works well within the constraints of the typical user's hardware platform. Thus, a crucial early step in the design process is to characterise the target user's anticipated platform and to bear these constraints in mind as the design unfolds. Also, it is important to distinguish between the delivery and use platforms since the two are not always the same. The term 'delivery' connotes how you will actually get the multimedia system to the user, whereas 'use platform' refers to how the student will access the system in normal use. For instance, some software applications are delivered to users via CD-ROM but must be installed on the user's hard disk before they can be used.

Case: The Galapagos Pilot - Levels 6 and 7

It is perhaps a little unfair to analyse the Galapagos Pilot at these two levels since one purpose of project was for the designers and creators to gain experience in multimedia system development. Thus in its original conception, fitting within the constraints of student home platforms was not a large issue. However, in order to illustrate the framework, we will go ahead and talk about how these levels are realised in the Pilot.

Level 6: Software Issues

The Galapagos Pilot was created using Director™. Several aspects of Director made it promising for this project. First, its scripting model makes it well-suited for realising the Pilot's strong, linear narrative flow. Second, unlike a general purpose programming language, Director has many high-level features for manipulating different media types, such as the many video clips which play a central role in the Pilot's stunning presentation. Finally, Director is multi-platform – systems built in Director can be made to run on both PC-compatible and Macintosh™ platforms, making the software available to a wider audience of students.

Level 7: Delivery and Use Platforms
In the case of the Galapagos Pilot, the delivery and use platforms are identical: CD-ROM. When you consider the many rich graphics, audio and video resources contained in the system, there is really little alternative. The resulting system is very large and many students would not have the necessary space to install the system on their hard disk. For the same reason, alternate delivery platforms such as the Internet would not be practical because of the sheer volume of media resources that would have to be transmitted over the telephone links available to most students.

Drawing the Levels Together
Having defined the levels, there are two key points to remember. First, in any given learning situation, levels 1, 2 and 3 (educational aims, teaching strategies, and task semantics) may or may not actually be present within the multimedia system. True multimediarity often breaks out of the electronic domain to encompass ancillary forms of learning materials, be they spoken, written, audio, or video. Some systems are intended to be used as stand-alone applications. Other systems depend upon the deployment of additional, external resources by the teacher, in order to contextualise the learning they contain. Yet other systems are intended to be self-contained, but due to constraints in the learning situation, cannot be used as the designers had anticipated (e.g., given the limitations of a school timetable, learners cannot always devote endless hours to unguided exploration). In this case, the teacher will need to find ways of providing guidance to constrain and limit the scope of exploration. This is a perfectly good way of proceeding. The only point we are making is that, ideally, the decision to have these levels covered by components outside of the system should be a *deliberate* design decision, rather than an oversight or an afterthought.

The second point is that, in designing systems, the design team needs to understand how all the levels work together or interact for a given project. As discussed earlier, there are mutual interdependencies between the levels, with decisions at one level affecting the possible outcomes at another level.

Now that we have a feel for the overall framework, let's return to Galapagos to see what insights our layered analysis offers us. This 'epilogue' analysis of the Pilot will help demonstrate the interaction between layers in terms of design issues and resulting usability.

Case: The Galapagos Pilot - Epilogue

In this concluding stage of our analysis we have the benefit of 20/20 hindsight – the Galapagos Pilot has already undergone developmental testing [37]. Here, we will review some of the major findings of this testing and see how these issues reveal themselves in our earlier layered analysis.

Finding 1: Narrative worked well.
Overall, students found the package straightforward to use and felt they would be able to successfully use it at home. The simplicity of the package was integral for establishing and maintaining the student's confidence with computer-based learning materials.

Observers credit the package's strong narrative line for the overall positive response. This very strong form of task semantic support appears to be well-suited for the needs of this target learner population, who are new to academic study and to computers. Thus, there is a good match between level 3 devices (narrative) and level 1 aims (target population needs).

Finding 2: Educational value of spreadsheet task unclear.
The first task involves transcribing information on finch populations into the Claris Works™ spreadsheet package. Most students found this task long, laborious, and not particularly meaningful from an educational perspective.

There are two interpretations of this finding, and probably both should be resolved in the next version of the Pilot. First, there seems to be a breakdown between the stated educational aims (level 1) and the teaching strategy taken (level 2). The stated aim is to give students a feel for the observation and note-taking skills involved in fieldwork. However, as our verb analysis from level 2 indicates, the task being done is one of 'entering data.' That is, students are taking data presented in one window and retyping or retallying it in another window. It is not surprising that students found this tedious and seemingly unrelated to meaningful fieldwork activities. Clearly, this task should be redesigned to better reflect the educational aims.

Another, related interpretation is that this is a breakdown at the task syntax level. Since entering data is not particular relevant to the task semantic, the user interface should be redesigned to eliminate this non-essential and error-prone activity, for instance by supporting copying and pasting.

Finding 3: Lack of support in the spreadsheet.
For several activities, students moved from the Pilot into a general purpose spreadsheet package (Claris Works™) where they were given no explicit support on how to use the package. Most students found this very

difficult, which is unsurprising given their complete lack of prior experience with software applications.

This finding stems from an unevenness in task semantic (level 3) support. The simple, narrative line in the Pilot provided very strong support, and then students were quickly transitioned into a relatively open-ended and high-functionality software package with no explicit support for their task. Either the task needs to be supported with simpler tools directly in the Pilot or additional materials are needed to help students use the spreadsheet package.

Finding 4: Problems with the map.
Many students lost track of which islands they had visited during their explorations of the archipelago (Figure 3). The result was that students kept revisiting islands and became annoyed when they had to relisten to parts of the audio track again.

This finding reveals two level 4 (task syntax) problems. The first is a problem with the navigation support provided by the map shown in Figure 3. While the map nicely shows the resources available to be explored, it does not show learners where they have already been and where they still need to go. A simple modification would be to change the appearance of islands that have already been visited. The second problem is lack of user control over audio resources. Audio resources that are very interesting the first time you hear them are usually not so interesting after repeated hearings. The interface should be modified to make playing the audio clips optional and under user control.

Finding 5: Better representations of finch specimens needed.
Students had difficulties classifying finch specimens because of a lack of adequate representations of the necessary features – namely, beaks and wing feathers. The Pilot provides a view of the beaks, but the resolution was inadequate for most students to spot the necessary differences between specimens. No explicit views of the wing feathers are provided.

Here, there seems to be a discrepancy between level 2 (strategies) and levels 3 and 4 (task semantics and syntax). The strategy calls for students to gain experience classifying finches; however, the necessary materials are not provided. At the task semantic level, the Pilot episode on finch classification needs to be re-designed to emphasise these two crucial anatomical distinctions. At the task syntax level, the media resources used in the system need to be carefully chosen to support spotting these difference across specimens. For instance, perhaps Gould's original drawings detailing beaks (see Figure 4) could be used instead of digitised photo images.

Finding 6: Perhaps a little too linear.
From the students' perspective, one perceived benefit of multimedia is the ability to easily revisit important or difficult areas in the package.

However, several students expressed a concern that, in the Pilot, getting back to a specific area often required watching everything all over again.

This difficulty indicates that the current resource organisation (level 5) should perhaps be reconsidered. At the moment, the package is extremely linear, with all four episodes proceeding sequentially and no linking across episodes. Students should minimally be provided some way of directly accessing the start of each different episode. Indeed, flattening the overall organisation, and providing some overview maps, may help students develop a comprehensible overview of the material in the Pilot.

SUMMARY

Using our framework, we have systematically analysed a particular educational application – the Galapagos Pilot. At the beginning of this report, we suggested that most people would probably have a difficult time explaining *why* various systems are different and the implications of these differences for effective use and design practice. The framework described in this report helps us to systematically describe systems and to identify the levels where key differences lie. This levelled view will enable us to compare one system with another, without confusing different discourses. The discourse of programming environments, for example, should not intrude upon discourse related to the task semantics. Similarly, educational goals will have very little direct connection with the organisation of resources, except via the teaching strategies/task semantics/task syntax hierarchy.

In addition to describing systems, this framework reminds us that any learning situation has components which need to be present in some form or other for the desired educational experience to occur. If the learning is to be assessed, it is not enough that users of multimedia systems have enjoyed themselves, or had a rewarding time. The experience must be contextualised within the course they are following. If these vital contextual cues are not embedded in the system, they must then be provided in some other way.

ACKNOWLEDGEMENTS

This report would not be possible without the hard work and generous support of the Galapagos Pilot design team at the BBC/Open University Production Centre and David Robinson. We also wish to thank Peter Whalley and Ben Hawkridge for invaluable discussions on the Virtual Microscope specifically, and for sharing their expertise about designing and creating educational multimedia in general.

The following trademarked products are acknowledged here: Director is a registered trademark of Macromedia Inc., Toolbook is a registered trademark of Asymetrix, Inc., mTropolis is a registered trademark of mFactory, Visual Basic and Encarta are registered trademarks of Microsoft Corporation, Java is a registered trademark of Sun, Claris Works is a registered trademark of Claris, Macintosh is a registered trademark of Apple Inc.

REFERENCES

1. Anderson, J. R., "Learning to Program in LISP," *Cognitive Science,* , Iss. 8, pp. 87-129, 1984.

2. Anderson, J. R., C. F. Boyle and G. Yost, "The Geometry Tutor," *IJCAI-85*, Los Angeles, CA, 1985, pp. 1-7.

3. Barrett, E. and M. Redmond, *Contextual Media: Multimedia and Interpretation,* The MIT Press, Cambridge, 1995.

4. Bork, A., "Preparing Student-Computer Dialogs: Advice to Teachers," in *The Computer in the School,*.

5. Brown, J. S., A. Collins and P. Duguid, "Situated Cognition and the Culture of Learning," *Educational Researcher,* Vol. Jan-Feb, Iss. pp. 32-42, 1989.

6. Chi, M. T. H., M. Bassok, P. Reimann and R. Glaser, "Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems," *Cognitive Science,* Vol. 13, Iss. pp. 145-182, 1989.

7. Csikszentmihalyi, M., "Literacy and Intrinsic Motivation," *Daedalus (Journal of the American Academy of Arts and Sciences),* Vol. 119, Iss. 2, pp. 115-140, 1990.

8. Detterman, D. K., "The Case for the Prosecution: Transfer as an Epiphenomenon," in *Transfer on Trial: Intelligence, Cognition, and Instruction*, D. K. Detterman and R. J. Sternberg, Ed., Ablex, Norwood, New Jersey, 1993, pp. 1-24.

9. Durbridge, N. H. and M. P. Stratfold, "Varying the Texture: A Study of Art, Learning and Multimedia," *Journal of Interactive Media in Education,* Vol. 96, Iss. 1, http://www-jime.open.ac.uk/jime/01/jime-01.html, 1996.

10. Fischer, G., "Domain-Oriented Design Environments," in *Automated Software Engineering,* Ed., Kluwer Academic Publishers, Boston, MA., 1994, pp. 177-203.

11. Fischer, G., "Turning Breakdowns into Opportunities for Creativity," *Knowledge-Based Systems Journal,* Vol. 7, Iss. 4, pp. 221-232, 1994.

12. Harel, I. and S. Papert, "Software Design as a Learning Environment," *Interactive Learning Environments,* Vol. 1, Iss. 1 (March), pp. 1-32, 1990.

13. Kintsch, E., M. Franzke, P. Haley and W. Kintsch, "Principles of learning in multimedia educational systems," Technical Report 95-06, Institute of Cognitive Science, University of Colorado at Boulder, 1995.

14. Kolodner, J., *Case-Based Reasoning,* Morgan Kayfmann, San Mateo, CA, 1993.

15. Laurillard, D., "Multimedia and the changing experience of the learner," *British Journal of Educational Technology,* Vol. 26, Iss. 3, pp. 179-189, 1995.

16. Laurillard, D., L. Plowman, J. Taylor and M. Stratfold, MENO project (Multimedia, Education and Narrative Organisation), http://www-iet.open.ac.uk/iet/MENO/meno-home.html, 1996-1998.

17. Lave, J., "Situated Learning in Communities of Practice," in *Perspectives on Socially Shared Cognition*, L. Resnick, J. Levine and S. Teasley, Ed., American Psychological Association, Washington, DC, 1991, pp. 63-82.

18. Lemke, A. C. and G. Fischer, "A Cooperative Problem Solving System for User Interface Design," *Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence,* , pp. 479-484, 1990.

19. Lockwood, F., *Activities in Self-Instructional Texts,* Kogan Page, London, 1992.

20. Malone, T. W., "What Makes Computer Games Fun?," *Byte,* , pp. 258-277, 1981.

21. Nakakoji, K. and G. Fischer, "Catalog Explorer: Exploiting the Synergy of Integrated Design Environments," *Software Symposium'90 (Kyoto, Japan),* , pp. 264-271, 1990.

22. Norman, D., *User Centered System Design, New Perspectives on Human-Computer Interaction,* Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

23. Norman, D. A., *The Psychology of Everyday Things,* Basic Books, New York, 1988.

24. Papert, S., *Mindstorms,* Basic Books, New York, 1980.

25. Papert, S., *The Children's Machine,* Basic Books, 1993.

26. Repenning, A., "Programming Substrates to Create Interactive Learning Environments," *Journal of Interactive Learning Environments,* Vol. 4 (Special Issue on End-User Environments), Iss. 1, pp. 45-74, 1994.

27. Resnick, L. B., "Introduction," in *Knowing, Learning, and Instruction: Essays in honor of Robert Glaser*, L. B. Resnick, Ed., Lawrence Erlbaum, Hillsdale, New Jersey, 1989, pp. 1-24.

28. Riel, M., "A Functional Analysis of Educational Telecomputing: A Case Study of Learning Circles," *Interactive Learning Environments,* Vol. 2, Iss. 1, pp. 15-29, 1992.

29. Sachs, P., "Transforming Work: Collaboration, Learning, and Design," *Communications of the ACM (Special Issue on Representations of Work),* Vol. 38, Iss. 9, pp. 36-44, 1995.

30. Schank, R. and Y. Jona, "Empowering the Student: New Perspectives on the Design of Teaching Systems," *The Journal of the Learning Sciences,* Vol. 1, Iss. 1, pp. 7-35, 1991.

31. Smee, P., "Computer Based Mapping: Why a computer is a useful tool with which to draw thought maps," CITE Technical Report , The Open University, Center for Information Technology in Education, 1994.

32. Soloway, E., S. Jackson, J. Klein, C. Quintana, J. Reed, J. Spitulnik, S. Stratford, S. Studer, J. Eng and N. Scala, "Learning Theory in Practice: Case Studies of Learner-

Centered Design," *Human Factors in Computing Systems (CHI '96)*, Vancouver, Canada (April 13-18), 1996, pp. 189-196.

33. Stahl, G., T. Sumner and R. Owen, "Share Globally, Adapt Locally: Software Assistance to Locate and Tailor Curriculum Posted to the Internet," *Computers in Education*, Vol. 24, Iss. 3, pp. 237-246, 1995.

34. Sumner, T. and M. Stolze, "Integrating Working and Learning: Two Models of Computer Support," *AI & Society*, Vol. 10, Special Issue on Hypermedia and Workplace Learning, pp. 70-78, 1996.

35. Sumner, T. and J. Taylor, "New Media, New Practices: Experiences in Open Learning Course Design," *Submitted to: Human Factors in Computing Systems (CHI '98)*, Los Angeles (April 18-23), 1998.

36. Sumner, T. R., N. Bonnardel and B. Kallak Harstad, "The Cognitive Ergonomics of Knowledge-Based Design Support Systems," *Human Factors in Computing Systems (CHI '97)*, Atlanta, Georgia (March 22-27), 1997, pp. 83-90.

37. Taylor, J., M. Oliver and S. Hatzipanagos, "The Galapagos Tutorial Developmental Testing Report," Technical Report 55, Programme on Learner Use of Media, Institute of Educational Technology, Open University, 1995.

38. Vygotsky, L., *Thought and Language,* MIT Press, Cambridge, 1989.

39. Whalley, P., "Imagining with Multimedia," *British Journal of Educational Technology,* Vol. 26, Iss. 3, pp. 190-204, 1995.

40. Whitelock, D., E. Scanlon, J. Taylor and T. O'Shea, "Computer Support for Pupils Collaborating: A Case Study on Collisions," *CSCL '95 (Computer Support for Collaborative Learning)*, Bloomington, Indiana (Oct. 17-20), 1995, pp. 380-384.

41. Wroblewski, D., T. McCandless and W. Hill, "DETENTE: Practical Support for Practical Action," *Human Factors in Computing Systems (CHI '91)*, New Orleans (April 27 - May 2), 1991, pp. 195-202.