The Life and Times of a Learning Technology System - the Impact of Change and Evolution

Claus Pahl
Dublin City University
Dublin 9
Ireland

Abstract

With the inception of the web now being more than 20 years ago, many web-based learning technology systems (LTS) have had a long life and have undergone many changes, both affecting content and infrastructure technologies. A change factor model can capture the various factors causing LTS to change. Methods for change-aware design of LTS have been suggested. The purpose of this investigation is, firstly, to add empirical results to aspects of these models and methods in order to show the relevance of such a change factor model by specifically looking at an LTS that has been developed, maintained and extended over a period of 20 years. Secondly, these results shall be used to develop a conceptual model capturing and assessing the impact of change. A key observation is an unexpectedly high impact of environmental constraints on the LTS, but also new opportunities emerging over time have had dramatic changes as their consequence.

Keywords

Learning Technology System; Interactive Learning Environments; Web-based Teaching and Learning; Software Evolution; Software Development and Maintenance; Change Impact Analysis; Software Ageing.

INTRODUCTION

Web technologies have changed the way teaching and learning is facilitated using digital means. With the inception of the Web now being more than 20 years ago, many web-based learning technology systems (LTS) have had a long life and have undergone many changes, both affecting content and infrastructure technologies (Palmer & Tulloch, 2001). In (Pahl, 2003), a facetted factor model to capture factors that cause LTS to change was introduced and a change-aware development method for LTS was suggested. The first objective of this investigation is to add more detailed empirical results to the change model in order to show the relevance of the model. The second objective is to develop these results further into an impact model for change analysis and impact assessment that can support the LTS software development and maintenance process.

We use as our case study an LTS that has been developed, maintained and extended over a period of 20 years (Smeaton, 1991; Smeaton & Crimmins, 1997; Murray, Ryan & Pahl, 2003; Pahl, 2008). During this period some notable changes have been observed, beyond the expected content updates to reflect changes in the subject domain. These include different instructors, an integration of content and infrastructure with the university's learning technology platform, the export of learning objects to content repositories and the shared delivery of content with several universities. A key observation is an unexpectedly high impact of environmental constraints, i.e. aspects that define how the LTS relates to its technical and organizational environment. In addition, new opportunities emerging and reacted to over time (Devedžić, 2006) have had a positive impact on the student learning experience and the effectiveness of system, but also massively changed the LTS itself, having an impact for those managing and delivering content through it.

Ten years on after the publication of an LTS change and evolution discussion (Pahl, 2003), the proposed solution shall be revisited in the light of collected observations over this period. The facets pertaining to the design and use of an LTS were categorised into content, format, infrastructure and pedagogy aspects. We evaluate this change model and discuss its adequacy. Changes do not only happen as a once-off activity, but are part of a lifecycle and reflect changes in the wider context of an LTS. We discuss the impact of changes in technology or teaching and learning research on LTS in everyday use and the long-term costs of these changes. Using a case study analysis, we integrate the lessons we learned developing, maintaining, and using a concrete LTS into a change impact analysis and assessment model. Our contribution is a discussion of change factors in the context of learning technology and infrastructures and how evolution impacts on LTS. Our aim is to highlight the dangers and pitfalls, despite improvements and advancements facilitated through change and evolution. These are captured in an impact model that can serve as an analytic model for LTS stakeholders.

This study confirms concerns in the software engineering community (Easterbrook, Singer, Storey, & Damian, 2008). Software ageing is a known concern for software development characterized by increasing maintenance costs (Taylor, Medvidovic, & Dashofy, 2009). As we will see, we need to look at software-centric ageing as well as contextual ageing. Two concrete forms are architecture erosion and degradation. Architecture erosion is caused by incremental changes of the system. Degradation is the abrupt consequence of some distinct events in the environment of the LTS. In order to facilitate the applicability of the impact model, we introduce suitable metrics for the architectural change concerns.

We first introduce principles of learning technology systems and the LTS we use as a discussion case study, the IDLE system. We use the change factor model to summarize the evolution of IDLE over its lifetime. The introduction of an impact model based on lessons learned follows, which takes into account observations and analyses of the system successes, but also difficulties and costs resulting from software ageing.

A LEARNING TECHNOLOGY SYSTEMS USE CASE

A learning technology system (LTS) is a software environment that provides learning content through a range of specific (e.g., Web-based) features. Features of some advanced LTSs include high degrees of interactivity and multimedia support, which are not necessarily supported by current commercial and open-source environments. These systems are often distributed and/or shared, which creates significant infrastructure demands. As an example of this category, we look at the evolution of a use case LTS - an interactive database learning and training environment called the Interactive Database Learning Environment IDLE - as our case study (Murray, Ryan, & Pahl, 2003; Kenny & Pahl, 2005); Holohan et al., 2006). Other systems like those described by Barak (2007) or Mitrovic (2007) will also be discussed.

IDLE supports second year undergraduate database courses. A SQL programming training part forms a central part of this course as applied database programming is one of the core learning objectives of the course. Programming (i.e. defining, updating, and querying database tables) is a skill that needs to be trained by the student. Understanding and mastering the overall development process of a database application is equally important. Database programming in the language SQL also requires conceptual understanding of the underlying data model with its structures, operations, and constraints. An instructional model for programming needs to cover a number of different aspects. IDLE provides four different features for the different instructional aspects of SQL programming:

- Conceptual knowledge. Conceptual knowledge is presented in a virtual lecture system based on recorded audio material and synchronized audio/slide presentation that emulate traditional classroom lectures.
- Procedural knowledge. SQL and parts of its underlying data model are about the execution of instructions. Procedural knowledge is presented in an animated tutorial system that allows the student to operate and visualise relational algebra expressions using Flash™ animations.
- Programming skills. SQL programming is the core activity, supported by an interactive and adaptive tutorial that guides the student through exercises to be worked on within the system - IDLE has a semantic analysis component that analyses and corrects student answers and, based on student progress, a recommender component suggests suitable further exercises.
- Development skills. SQL programming is part of the overall database application development process, which is supported by an integrated virtual lab environment with modelling, programming, and analysis features based on graphical editors, automated correction and recommendation features.

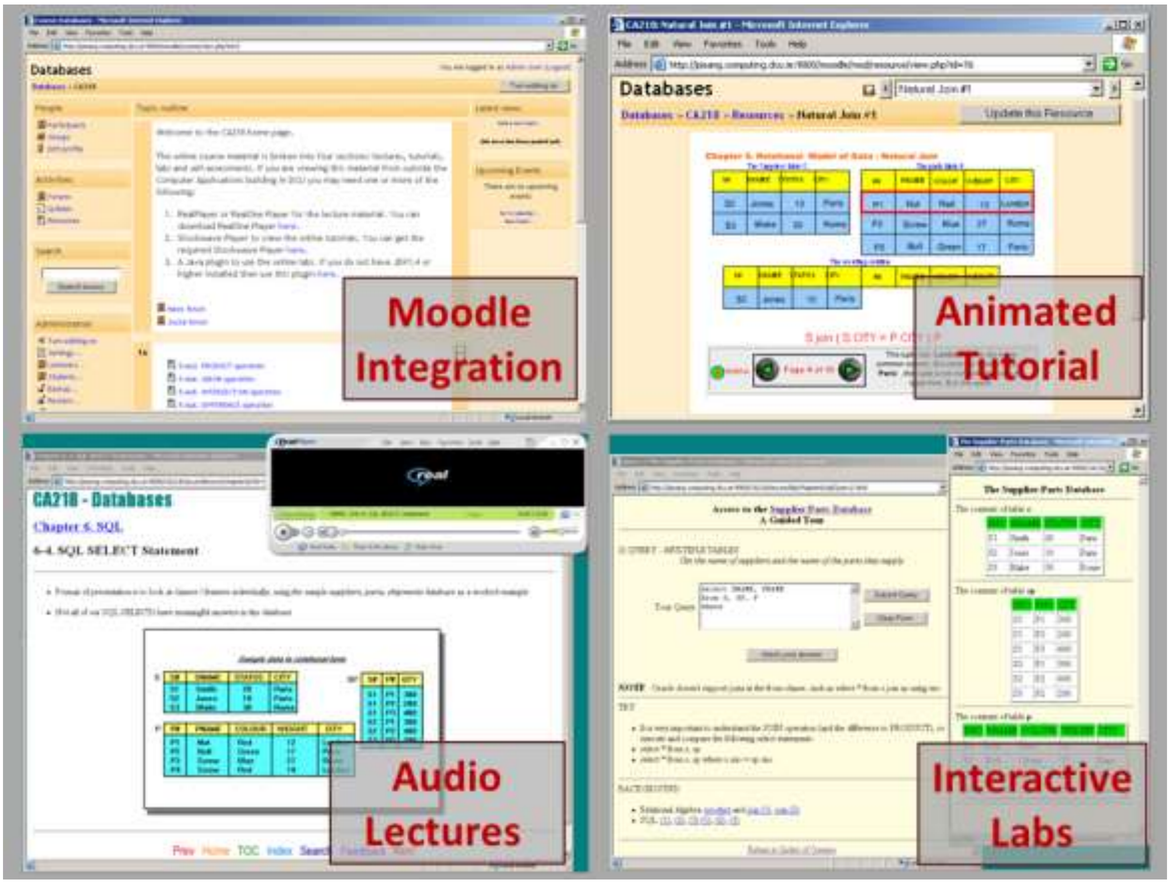

Fig. 1. IDLE Features.

IDLE allows the concurrent combination of lecture, tutorial and lab features (see Fig. 1) - something not possible in a normal classroom setting. The aim is to support a learning-by-discovery style, allowing students to acquire skills, but also to construct and deepen conceptual knowledge through activities in meaningful and realistic problems.

IDLE is a feature-rich Web-based LTS that uses multimedia and interactive technologies for student access, but that also requires support software on the server side like the editors and analysis and recommender components with their background databases. IDLE embodies features of many intelligent, interactive and adaptive LTS through its intelligent tutoring and skills training features, which include an adaptive component that recommends exercises depending on learner progress. This allows us to use IDLE as a template for the evaluation and development of changes factors and their impact of a range of interactive, intelligent and adaptive multimedia LTS. Note that some types of LTS that include for instance context-aware or ubiquitous computing features are not covered here.

In addition to its primary role of an LTS aiming at improving the learning experience, IDLE is also an instrument for research into learning technology, and collaboration and internationalization - which has affected how IDLE has changed. This is a first indication that in LTS development and maintenance, stakeholder requirements are important beyond classical functional and quality requirements. How change has taken place and what the impact was shall now be discussed.

LITERATURE REVIEW – CHANGE AND EVOLUTION OF LTS

The aim of software evolution as part of a development and maintenance process is to implement (and revalidate) the possible major changes to a system. Our change factor model for the capture and representation of change and evolution shall be introduced and applied to specify LTS changes. In addition to an investigation of the IDLE evolution, we also discuss other work on LTS change and evolution.

- Migration. Often, the initial concern is the transition from a traditional setting to a digitally based and delivered one. Barak (2007), Narwanvi & Arif (2008) and Ge, Lubin & Zhang (2010) address this migration from a technical and also a stakeholder perspective. Barak (2007) identifies four transition steps as a technical approach to migration and discusses the benefits, but also the complexities seen by the instructors. Narwani and Arif (2008) focus on adaptation problems in adopting standard LTS, i.e. required change to the LTS and also the environment in which it is running. Ge et al. focus on the transition between LTS platforms. Contributing factors to the success of a transition, such as systems support and support for pedagogical and domain-specific issues, are highlighted.
- Incremental Development. The design of complex, advanced LTS is often a staged, incremental process. Mavrommatis (2008) and Mimirinis & Bhattacharya (2007) report on the development of these systems. Mavrommatis describes a systematic, domain-specific approach based on reusable learning objects development following the SCORM standard, embedded into instructional design principles. Mimirinis and Bhattacharya investigate the relationship between the pedagogical perspective, in particular different approaches to learning and the learning environment as a technical and organisational space. Pahl (2008) suggests a systematic, ontology-based development methodology for LTS that takes into account typical content formats and delivery architectures for Web-based teaching and learning.
- Evolution-driven Development. In (Wu, Chen, Wang & Su, 2010), evolution is addressed as a guiding principle. The authors describe the design of an interactive, Web and service-based learning environment by using an evolutionary approach in an attempt to engineer educational software. Incremental expansions of simulations used in their LTS based on instructor and learner suggestions have been implemented.

Common to all of these is an awareness of the need to address changes as part of a systematic LTS evolution strategy. They have investigated factors that play a role in design and continued delivery and maintenance of LTSs - ranging from user perceptions to organisational and technical (infrastructure) aspects to the pedagogical perspective. These different factors need to be taken into account to address the needs of all stakeholders such as learners, instructors or the organisation that offers a course.

A CHANGE FACTOR MODEL – IMPACT DETERMINATION AND EVALUATION

At the core of our change management solution is a change factor model for incremental and change-driven LTS development and maintenance. We complement this with a change model-based impact determination scheme. The discussion of the above reports on LTS change and evolution has identified common concerns, but also that the characterisation of IDLE as a multi-feature LTS demonstrates that IDLE can serve as a single evaluation object - a template that captures a range of features common to the evolution and change of a variety of modern

LTS, and, therefore, allows us to focus on IDLE as an empirical evaluation subject and to justify conceptual contributions covering properties of a range of similar LTS.

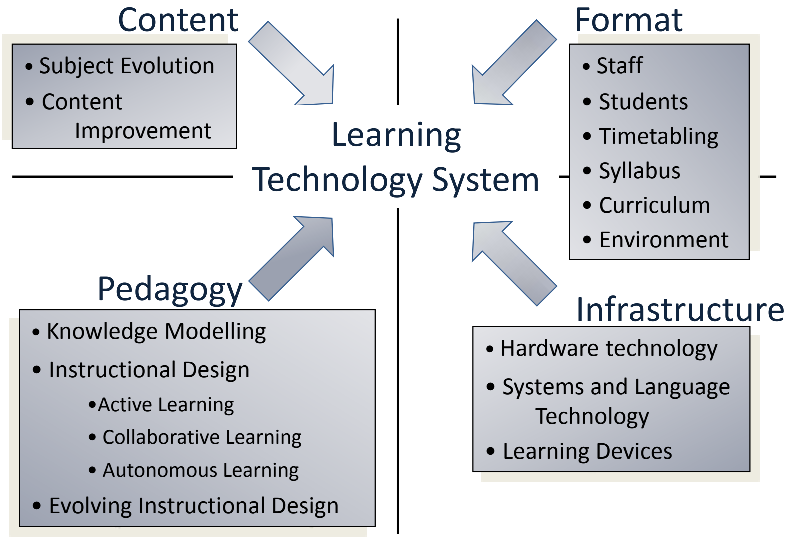

Fig. 2. Change Factors and Facets. (Adapted from (Pahl, 2003))

Change Factor Model

In (Pahl, 2003), we presented a change factor model that categorises change factors, i.e. LTS-external requirements that cause an LTS to change, into a number of facets. This categorization serves not only as an analysis and impact determination technique (post-development), but also as a change-aware design methodology to categorise requirements (pre-development). The model captures the following factors (Fig. 2), with relevant facets for each factor:

- Content - the subject-oriented perspective - refers to the subject taught and the representation of knowledge in the LTS and captures changes relating to the subject, motivated by internal or external change factors.
- Format - the organisational perspective - comprises attributes determined by the institutional context: curriculum, syllabus, staffing, students, organisational aspects like timetabling and other environmental factors like legal and financial issues. This factor captures changes related to people (staff and students) involved, content (curriculum and syllabus) and the organisational environment.
- Infrastructure - the technical perspective - relates to the hardware and software environment in which the LTS is deployed and captures changes due to developments in hardware/software technology or the emergence of new learning devices.
- Pedagogy - the educational perspective - refers to the instructional design or model of the LTS determining in which way the course is taught and captures the evolution of teaching and learning in computer-supported environments. How knowledge is represented in content, how students interact with the content (degree of interactivity, group work and autonomy) and the impact of progress in instructional design (covers knowledge/content representation and learner/content interaction) are concerns.

In (Pahl, 2003), we looked at the LTS-internal perspective, i.e. the effect of changing requirements on an LTS in isolation, which we will exemplify now. We apply the model, but we also look at LTS change and evolution from an external perspective, i.e. we look at drivers of change and how change impact is dealt with as a result of changing requirements, thus providing a further evaluation of the change model.

The change factors are external aspects. Two types of LTS components can be identified: digital content as a representation of the subject content and LTS software infrastructure to deliver content. Generally, both component types are affected by the four change factors. For instance, changes in the subject domain (content) might not only change its digital representation, but also the way it is delivered. An LTS component development and change methodology is beyond our focus on change drivers here.

Criteria for Change Effort Determination

Criteria are needed to classify the consequences of externally initiated change by approximating the required change effort through three broad categories (we use high/medium/low). Change effort includes direct costs in infrastructure (investment) and work (software development and maintenance), but also reflects the complexity of new technologies or methods to be mastered (stakeholder training). Some comments justify the category ratings, see Tables 1a to 1d. The effort categories serve as an impact determination scheme, which serves to

- identify an overall impact by averaging out all individual contributions from the factors and their sub-facets, thus determining the overall cost of envisaged changes,
- identify critically high impacts at an early stage to judge the overall feasibility and to identify maintenance activities that require particular attention.

Table 1a. Criteria for Change Effort Determination for Subject Content Change Aspects.

Subject Content:	**Low**	**Medium**	**High**
Subject evolution	minor additions to content as a response to a changing subject	significant changes to content	complete revision of content
Content improvement	minor corrections as content is changed in order to improve the material in a planned process	more regular / frequent / systematic corrections	substantial and continuous corrections

Table 1b. Criteria for Change Effort Determination for Format Change Aspects.

Format:	**Low**	**Medium**	**High**
Staff	insignificant change relating to educators, course developers, or technical support staff without need for retraining	some staff change, some training involved	significant change with varying staff profiles (research, technology expertise) and training needs
Students	student body changes marginally in terms of numbers, qualifications, or mode of learning	student body changes significantly in some aspects	student body changes significantly in all aspects
Timetabling	minor changes in	some changes in	significant

	relation to where and when a course takes place	relation to part-time or online delivery	changes in relation to part-time or online delivery
Syllabus	minor content / course organisation changes to reflect internal changes	some content / course organisation changes to reflect internal changes	restructuring as a response to internal changes
Curriculum	minor content / course organisation changes as a response to organisational needs that require changes in level, extent, or prerequisites of courses	some content / course organisation changes as a response to course program changes	restructuring as a response to course program changes
Environment	minor content / course organisation changes as a response to legal or financial changes	some content / course organisation changes as a response to legal or financial changes	restructuring as a response to legal or financial changes

Table 1c. Criteria for Change Effort Determination for Infrastructure Change Aspects.

Infrastructure:	**Low**	**Medium**	**High**
Hardware technology	minor changes as a result of changing / improving communications and network technology, computing power, and computer platform	changes to a number of system components	change in (almost) all system components
Systems and language technology	minor software maintenance as a response to technology leaps, legacy or pre-eminent technologies are frequent issues	significant software revision as a response	complete redevelopment as a response
Learning devices	minor change in software and hardware such as smart objects, information infrastructures and virtual environments serve as learning devices	noticeable changes in new devices	major migration to new delivery platforms and devices

Table 1d. Criteria for Change Effort Determination for Pedagogy Change Aspects.

Pedagogy:	**Low**	**Medium**	**High**
Knowledge modelling	minor changes regarding the acquisition, modelling of and access to knowledge	noticeable changes	major revision as to how content is accessed
Instructional design - active	minor changes in relation to how learners engage in interaction	noticeable changes	major revision as to how learners interact with the LTS

learning			
Instructional design - collaborative learning	minor changes regarding how learner communication and collaboration is supported through communication systems	noticeable changes	major revision as to how learners communicate and collaborate
Instructional design - autonomous learning	minor changes in relation to how learners personalise and make independent their learning through adaptive technology	noticeable changes	major revision as to how learners work on their own and customise their learning
Evolving instructional design	minor impact of a planned evolution integrated in the design through course evaluation	noticeable changes	major changes as a consequence of unexpected external change factors

This impact determination scheme shall now be applied.

Validation Methodology

The validation reflects a systematic research approach that includes empirical results regarding the IDLE evolution gathered from different sources. The resources used include

- other published papers my the IDLE developers and instructors (Crimmins, Kenny, Murray, Pahl, Ryan and Smeaton, as referenced), specifically on the development of special features and the overall architecture,
- reporting used to determine changes to content and infrastructure and the costs associated to this: project reporting as most of the development has been financed through research and university-internal learning technology development schemes – the respective project reports describe the development and importantly costing of activities – and project software documentation also contains relevant data on changes to the infrastructure. Additionally, effort calculation for student projects has been factored in,
- records of student numbers and activities provided by the instructors based on their own teaching and examination records.

A central role in the change effort determination played the principal investigators and the senior researchers of the research projects and the instructors and coordinators of the course.

Empirical Determination of Change and Evolution Impact

IDLE has changed continuously over the past 20 years - caused by different factors which are summarized using the detailed change model from Fig. 2 and applying the impact determination criteria from Tables 1a to 1d. Each change aspect is categorized in terms of its impact on the IDLE LTS. We provide an impact characterisation for each facet, but also an equally weighted summary impact per aspect.

Subject Content [low - continuous, but moderate]

- Subject evolution: Low - minor additions to content have been made as a response to improved technologies in the database field to be covered by the course.

- Content improvement: Low - minor corrections to address presentation and technical issues have been carried out.

Format [low/medium - the course had to be shortened, which is the most important change in this category]

- Staff: Medium - in total 4 lecturers have been involved - with 3 of them familiar, of which 2 permanent academics and 1 replacement instructor, and 1 unfamiliar with the underlying technology. Staff training has become an issue over time as recent instructors were not familiar with underlying technology. The initial lecturers have seen the system development also as part of their research.
- Students: Low to medium - the student profile changed slightly, but no effect except one significant increase of numbers affecting server support from a technical perspective.
- Timetabling: Medium - part-time evening delivery and support at that time has been a problem for a period when the stability of the system was not as good as desired.
- Syllabus: Low - content has been shortened as part of a degree restructuring, in which some theoretical aspects were removed.
- Curriculum: Medium - change caused a syllabus update (see above) to align to a new course credit system
- Environment: Low - none in our case, but possible issues could be the legality of behaviour tracing of identifiable students (which is possible once a login to the system is required).

Infrastructure [high - significant hardware and software changes]

- Hardware technology: High - enabling and disabling changes have occurred - including a more powerful database server allowing us to serve larger student groups, but also the use of outdated systems without maintenance contract, which forced the withdrawal of services.
- Systems and language technology: High - new technologies (e.g., flash, XML, Semantic Web) emerged and have been incorporated with some considerable costs (but also notable improvements in effectiveness).
- Learning devices: Low - new devices (e.g., mobile) have not been supported, but a national learning object repository can be considered as another content storage and delivery platform.

Pedagogy [medium/high - from online to blended, but also more active forms of learning]

- Knowledge modelling: High - in the context of provisioning LTS components through a national repository, annotation and access infrastructure needed to be provided (packaging).
- Instructional Design - Active learning: High - initially considerable investment into developing active learning support (online tutorials and labs).
- Instructional Design - Collaborative learning: Low - not done.
- Instructional Design - Autonomous learning: High - instruction and support developed with an emphasis on independent learning at later lifecycle stages (including automated correction and feedback mechanisms).
- Evolving instructional design: Medium - often neglected because of a research prototype focus, only partly compensated by the instructors' technical expertise.

Validation of the Change Factor Model and the Impact Determination Scheme

The analysis based on the change factor model and the impact determination organised around the factor structure has been successfully applied for the IDLE system. This empirical investigation, but also more general observations we made on related work earlier confirm that the proposed model is adequate - all aspects are relevant as changes have occurred in all of them and the model is sufficiently complete to capture and categorise all changes to IDLE. In relation to the model presented in (Pahl, 2003), we changed the pedagogy factor to reflect a more generic concern (Instructional Design) on top of individual, possibly changing approaches. The educational approaches included now are sufficient to describe our system and possibly most LTS, but might need to be extended for some others or in the future if new approaches emerge, like learning in mobile or social media contexts. The impact determination allowed us to categorise impacts and identify critical concerns.

An aspect that deserves further attention is learner and instructor perception. Staff and learners have been captured in the original model as a kind of human system component interacting with technical components, as the model's focus was on technical development and maintenance. A solution to the need to reflect the importance of the users would be to include a Stakeholder factor, which we could add to the Pedagogy factor since the concern is the interaction with content and the system that delivers it, i.e. how content is facilitated and consumed (Sommerville et al., 1993; Sharp, deSouza & Dittrich, 2010).

AN IMPACT MODEL

While the application of the change model confirms that change is a ubiquitous and multi-facetted problem in LTS development and management, we need to go further by looking at the benefits and costs of change as forms of impact (Abgaz et al., 2012). We look at the sustainability of change and factors that influence this sustainability by analysing the changes we recorded. We summarise observations regarding the actual impact of change in an LTS lifecycle. The results of observations and analyses are captured and formalised in a conceptual impact model. Impact refers here to the consequences of change regarding the effectiveness of LTSs in functional terms and the costs incurred as the result of required or desired change. We analyse the relationship between change on one side and success and limitations as effectiveness concerns on the other side. The impact of change in terms of costs arising is another issue. Finally, we look at implications for the future. This discussion provides us with a conceptual model, see Fig. 3,

- (factors and impact) that links characteristics of the LTS, factors that drive change and the impact on the system - helping stakeholders like instructors and support staff to understand why and how LTS change and to predict and manage change,
- (impact analysis and assessment) helps stakeholders in judging and justifying the benefits and benefitting from the opportunities arising in terms of teaching and learning and educational technology research, but also drawbacks of change in terms of costs.

We complement the model by some methodological aspects (impact reduction) addressing how to overcome increasing change costs and side-effects by managing, but also developing and sharing LTS effectively. We introduce the model in three parts (and motivate them later using IDLE):

- change factors as causes and change impact determinants as consequences,
- change categorisation and metrics for impact analysis and assessment,
- proposals for impact reduction.

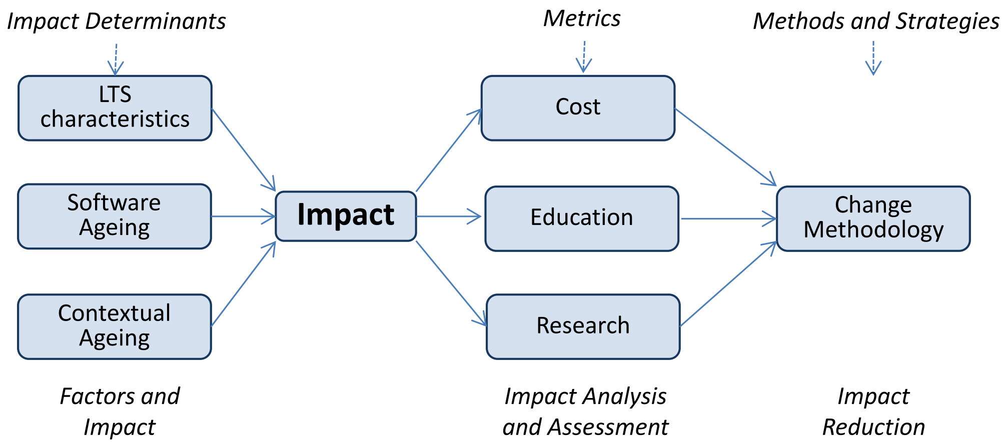

Fig. 3. Outline of a Conceptual Impact Model.

Factors and Change Impact Determinants

The change model from Fig. 2 has proven itself to be useful. All actual changes that have occurred do fit into the categorisation scheme. We can consider the factor taxonomy stable, apart from the pedagogy factor which acts as a container to capture important learning technology trends. We now look at three impact determinants:

- characteristics of the LTS and how their change impact affects the LTS and users,
- the software ageing process of the LTS as a software system consisting of the software architecture, code and content, and
- specific contextual ageing issues in relation to external factors like learning and software research and technology development.

These are presented in the form of qualitative aspects, whereas other impact model elements will be quantitative in nature. An important first observation to be captured in the model is that the impact of change factors varies depending on characteristics of the LTS:

- The more research-driven the LTS is, the more important are the infrastructure and pedagogy factors (and consequently the more cost/effort is required) to keep the system up-to-date and at the leading edge of research. This becomes apparent when comparing our IDLE system with the platform used in our university to deliver education electronically. While the platform has evolved, it did so in a controlled way, allowing the educators to focus on content and format issues.
- Equally, the longer the life of the LTS, the more important these two factors become as both are subject to change. We have experienced this with IDLE, resulting in even the unavoidable costs for maintenance as part of a normal software and infrastructure ageing process to increase in a non-linear fashion.

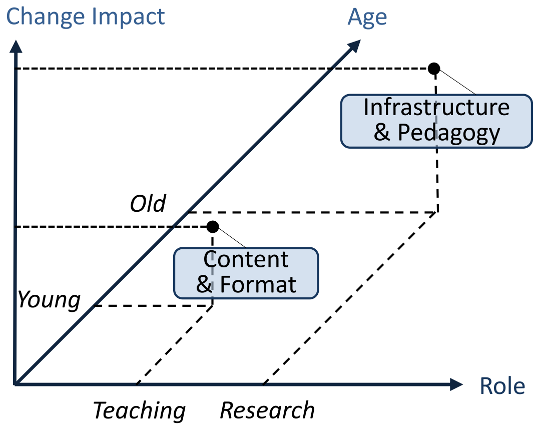

Fig. 4. Impact Model - LTS-internal Characteristics and their Impact on Factors during Change.

In terms of impact analysis, it is important to distinguish the staff categories impacted by change. Two characteristics of LTS determine the impact significantly: the role and age of the LTS, i.e. external factors need to be correlated with internal change impact, see Fig. 4 where the characteristics role and age are presented. Other LTS characteristics with an impact include media-richness or distribution. All enhance the importance of the infrastructure and pedagogy perspectives. Content and format are mainly influenced by the day-to-day environment (the institution that provides the learning experience).

Content and format are the primary concerns of the instructor, whereas infrastructure and pedagogy are the responsibilities of technical support staff. While this is in general obvious for infrastructure aspects (unless an instructor develops her/his own advanced software/content features), the pedagogy perspective is often difficult to influence or change for an instructor. Learning environments often follow certain philosophies and support a specific form of learner-content interaction. The open-source environment Moodle (Moodle, 2012) for example encourages a social constructionist pedagogy. Whether collaborative or active forms of learning can be facilitated might depend on the strictness of the platform in enforcing a particular model (Moodle is open in this regard).

A second concern is that the LTS lifecycle reflects a software ageing process affecting the infrastructure perspective (Grottke, Matias and Trivedi, 2008; Magalhães and Silva, 2010). Software ageing is a known phenomenon in the software industry characterized by increasing maintenance costs. Maintenance and evolution cause ageing, often called erosion when changes violate requirements or principle design decisions (Taylor, Medvidovic, & Dashofy, 2009). This process is negatively impacted through research prototyping before proper systematic software production methodologies are applied. Ageing is caused for instance by platform technology becoming out-dated, software system maintenance becoming more expensive, increased complexity of managing a system of different components developed at different stages. These are common factors for all software, but can increase if feature-rich systems in rapidly evolving environments like LTS are in question.

Thirdly, in addition to this internal software-centric ageing, we can also look at contextual ageing in the context of developments in two domains: learning technology research (LT) and Web and Internet software technology development (ST), see Fig. 5, i.e. context factors impacting on infrastructure and pedagogy. The LTS reflects the time it lives in. The technical infrastructure revolutions in the computer technology domain that led to multimedia, Web and

Internet technologies are obvious. This evolution phenomenon can also be observed for the pedagogy factor where new approaches are developed, explored and experimented with and eventually become mainstream. Significant trends have been incorporated in IDLE - some adopted early (even before they became mainstream), indicated through shorter arrows, some adopted later, indicated through longer arrows in Fig. 5. Research, however, causes a form of opportunistic evolution (exploiting a technology opportunity to follow educational trends). This is in principle costly, but can to some extent be alleviated through reuse and knowledge transfer, e.g., by incorporating externally developed sharable content. In some cases, there are obvious links between the LT and ST aspects where LT takes new technologies into account and facilitates new forms of learning - adaptivity needs semantic technology support; equally, collaborative approaches need adequate technical solutions such as shared workspaces.

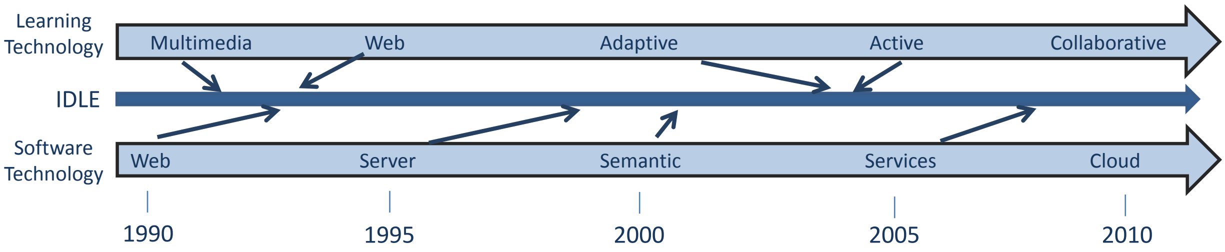

Fig. 5. IDLE Evolution in the Context of Learning Technology and Software Technology Evolution - Selected Trends.

So far, we have not clearly distinguished change and evolution. Fig. 5 indicates the major evolution steps that IDLE has undergone. Change, in contrast to macro-level evolution, is a small-scale, micro-level process. The observations above have primarily addressed evolutionary aspects, i.e. the effect of long-term change.

Categorisation and Metrics for Impact Analysis and Assessment

We have looked into what kind of an impact change has. While change has always has an impact, this impact is sometimes beneficial and desired, but sometimes has disadvantages in terms of costs without being beneficial. We can categorise changes that happened by their actual benefit (e.g., if effectiveness improvements are the result of a new technology adaptation (i.e. a change):

- beneficial and justifiable changes that improve for example the effectiveness of learning (through improved technology or pedagogy),
- non-beneficial changes, but that leave effectiveness and the overall quality of the LTS intact, like migrating away from unsupported platforms,
- detrimental changes with negative effect or quality impact in addition to costs, like the discontinuation of crucial platform technologies.

While impact of change is unavoidable - the result might often be desired, but sometimes also costly. Thus, the impact needs to be looked at and justified in terms benefits and negative implications. We add a quantitative part based on a number of assessment metrics to the impact model.

Overall, IDLE has been a success as a learner support system and as a research vehicle. In educational terms, student attainment in written examination and practical work has improved continuously while we extended IDLE (Kenny & Pahl, 2005). The IDLE system and approach to active learning have also received high appreciation by the students (more than 1000 students used the system). Another benefit is the fact that IDLE has introduced many of our students to online and blended life-long learning technologies, which many will encounter again during further education throughout their careers. In terms of research, IDLE as a research prototype has led to a significant number of publications, established the reputation of the research group and led to various contacts and collaborations with other researchers, groups and institutions.

For our impact model, the positive impact of change can be measured in terms of metrics that capture the educational and research improvements resulting from change. Here are suggestions for metrics addressing the two main concerns:

- educational perspective: satisfaction of users; usability, number of users; percentage of performance increase per person month (PM) of development
- research perspective: amount of research funding; published papers per person month (PM) of development; citations per PM of development

Cost is a potentially negative impact. With about 50 person months development time (one M.Sc. research project, paid software development carried out by research assistants and contributions by students through projects and internships), the project has resulted in eight journal, six book-chapter and eight conference publications. In the four major evolution steps, we achieved an improvement of around two per cent each time in terms of student attainment.

However, high-end features in LTS are showpieces that raise the developer's academic profile. While initially cost-effective to develop, IDLE is a system that has turned out to be costly and difficult to maintain. New investments and new developments are required if new concepts are to be implemented. For many years, we have attempted to include new features and update the infrastructure, but eventually the costs of maintaining the system infrastructure and training have caused us recently to discontinue the support of the system. However, if a system cannot be developed further as in our case, sometimes a community might be willing to continue development and support (for instance in an open-source setting). We will discuss this later on.

Responsible for the eventual winding down of the system were the increasing maintenance costs. The development costs have been largely covered through funded research projects, but the development has also significantly relied on the unpaid contributions of the faculty members and project students involved. However, maintenance and other activities interoperability have not been as adequately covered:

- Maintenance of hardware and software - both have incurred costs, particularly more than 10 years in operation have resulted in, first, the removal of some software features and then retirement of the whole system. Training as a maintenance concern on a human level has also been a problem.
- Exchange and collaboration have partly been supported through grants allowing us to make some components available as reusable learning objects for a national learning object repository, which in turn allowed us to deliver the course in collaboration with universities in two other countries.

Without an institutional strategy, such systems are difficult to maintain over longer periods.

Methods and Strategies - Change-aware Development and Management of LTS

Development methods and strategies can complement the conceptual model with its qualitative and quantitative aspects. Software and content can be developed with change in mind. Change-aware engineering methods exist - generics ones (Taylor, Medvidovic, & Dashofy, 2009; Magalhães & Silva, 2010) and domain-specific ones (Mimirinis & Bhattacharya, 2007; Pahl, 2008). In our own work, we have pointed out aspects that are especially crucial if dealing with change. Simple efforts like proper documentation or the use of standards do help, but are often neglected in research-oriented systems in order to facilitate a deeper scientific investigation.

Dealing with long-term change is costly and can result in systems becoming unmaintainable, in particular for once-off developments that are not within the platform technologies supported by the organisation. A way out of this dilemma beyond standard engineering methods is the componentisation of LTS content and infrastructure, which would allow more reuse and sharing to take place. While IDLE as a system is not supported any further, at least parts of it will remain as reusable learning objects in a national learning object repository. For this purpose, content (including software support) has been annotated and deposited in a repository. This makes content and infrastructure components available to other instructors (subject to some IP agreements). Some IDLE features, like animations lend themselves for this purpose; others, like service-based interactive features require prior installation of software, requiring some technical expertise for reuse. User-generated content in addition to instructor-generated content can contribute to this effort of reducing costs.

Another avenue that might be open to similar systems is commercialization. IDLE is not a general-purpose LTS, but a very subject-specific system. However, as the example of a similar system, the SQL-tutor (Mitrovic, 2003), by the University of Canterbury in New Zealand, shows (see (Addision Wesley, 2012) where the system is made available by a publisher as the Database Place), the commercialization of such a system as textbook companion available through a Web tool, is also an option.

Limitations and Threads to Validity

The application of the impact model focusses on the determination of impact in qualitative terms. Determining whether change should be carried out is not an aim. Whether change is beneficial needs to be decided by stakeholders. A quantification of the categories beyond characteristics such as predictability is part of our future work.

The reliability of the models for impact determination depends on the IDLE system, which we have primarily used to evaluate the models. We have, however, argued in the literature review that IDLE is a sufficiently rich LTS and is therefore representative of a broad range of LTS, particularly high-end research-driven systems.

CONCLUSIONS

While change and evolution in learning technology systems are, not unexpectedly, common, the problem is often aggravated by a number of factors. In academic and research environments, research-oriented prototypes often have a longer-than-expected life span as the discussion of related work on change and evolution of LTS earlier on shows. While this is not the rule, the IDLE and Database Place examples, and also long-lasting activities based on the Knowledge Tree platform that has been used for database courses (Brusilovsky, 2004), are some examples for a specific subject. The rapid evolution of particularly the Web and Internet platform and multimedia technologies requires constant updating and migrating. Learning technology research takes new platform technology on board to facilitate new forms of learning. These factors together cause a rapid ageing process in an LTS lifecycle process to take place. The trend towards mobile and informal learning, often embedded in new types of environments (e.g., mobile devices or social platforms), is another development that LTS might have to be adapted to.

We have used a conceptual model to categorize changes to an LTS. The model is neutral and non-judgmental by identifying change factors only. However, we need to analyse whether change is beneficial and what the impact is. Primary change factors such as changes in the subject domain or changes in the organizational context of a course will always have a certain, but usually predictable impact. However, our observations have demonstrated that factors such as platform and pedagogy have a higher, often detrimental impact the longer the system runs (de Silva and Balasubramaniam, 2012). Two clusters of change factors emerge and cause an ageing process:

- Erosion - Content and Format: These change factors cause LTS and (digital) content erosion - a gradual, evolutionary process based on incremental changes. These changes are part of the usual remit of an instructor.
- Degradation - Infrastructure and Pedagogy: These factors cause more abrupt degradation - caused by distinct, often discontinuing events. These changes have implications beyond the usual instructor's remit.

As technical contributions, our analysis and impact analysis technique contains a change factor model, an impact determination scheme and an impact model for further analysis. The change factor model helps stakeholders (decision makers and software developers) in organising change aspects into factors. The impact determination scheme can then be used to qualify impact based on the factors. The impact model with its categorisations and metrics acts as a tool to assess, understand and predict impact.

Without additional support, resulting costs can made an LTS unsustainable. Often costs are only justifiable if the LTS serves different roles, e.g., as a delivery system, but also as a research vehicle. These potentially high-impact factors all relate to an LTS and its links and connectivity to its wider context in terms research and technology and communities involved. While some developments have a negative impact and are prohibitive in terms of sustainability, even complex systems have a change of survival (at least in parts) based of the same connectedness with the environment. With a culture of sharing and community-based development (cf. learning object repositories in different countries or open-source software development), efforts made by individual researchers or groups can benefit larger communities and can be utilized and developed further beyond their abilities.

While we ended with notes on negative aspects of change and its management, the research-oriented LTS prototypes we looked at do have strong benefits for learners and instructors alike that should not be forgotten. Both learn about or learn in novel environments where quality improvements can be qualitatively and quantitatively evidenced. A driver behind change in these settings is the aim to further improve the learning experience and the facilitation of teaching guided by research in learning and software technologies. The mutual interaction between infrastructure and pedagogy research and development is important for an LTS to be successful from a research and a learning perspective.

We have looked at the lifecycle of an LTS - the life and times of IDLE - as an investigation into how LTS change and how change takes places as an interaction between LTS and its environment consisting of stakeholders, technologies, research and organisational concerns. The IDLE life as a system has, as a research-oriented system, always been a reflection of the time it lives in and in which it is developed and used, materialised through current research context at a moment of time. IDLE has, however, to some extent also influenced and driven this context further. To achieve this, constant change was necessary.

Acknowledgements. The author is greatly indebted to Alan Smeaton, without whose efforts the project would not have started and IDLE would not have seen the light of the world.

REFERENCES

Abgaz, Y.M., Javed, M., & Pahl, C. (2012). Analyzing Impacts of Change Operations in Evolving Ontologies Composite Ontology Change Operators and their Customizable Evolution Strategies. 2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics EvoDyn 2012, Boston, US.

Addison Wesley (2013). Database Place - Online Database Course Resources. Retrieved November, 14, 2013, from: http://www.aw-bc.com/databaseplace/

Barak, M. (2007). Transition from traditional to ICT-enhanced learning environments in undergraduate chemistry courses. Computers & Education, 48(1): 30-43.

Brusilovsky, P. (2004). KnowledgeTree: A distributed architecture for adaptive e-learning. Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters). ACM Press. Pages 104-113.

de Silva, L., & Balasubramaniam, D. (2012). Controlling software architecture erosion: A survey. J. Syst. Softw. 85(1), 132-151.

Devedžić, V. (2006). Semantic Web and Education. Berlin: Springer-Verlag.

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. Guide to advanced empirical software engineering (pp. 285-311).

Ge, X., Lubin, I.A., & Zhang, K. (2010). An Investigation of Faculty's Perceptions and Experiences when Transitioning to a New Learning Management System. Knowledge Management & E-Learning: An International Journal. 2(4):433-447.

Grottke, M., Matias Jr., R., & Trivedi, K.S. (2008). The Fundamentals of Software Aging. Proc. 1st International Workshop on Software Aging and Rejuvenation.

Holohan, E., Melia, M., McMullen, D., & Pahl, C. (2006). The generation of e-learning exercise problems from subject ontologies. Sixth International Conference on Advanced Learning Technologies ICALT'2006 (pp. 967-969).

Kenny, C. and Pahl, C. (2005). Automated tutoring for a database skills training environment. ACM SIGCSE Symposium 2005 (pp. 58-64). ACM Press.

Lei, X., Pahl, C., & Donnellan, D. (2003). An evaluation technique for content interaction in web-based teaching and learning environments. Proceedings 3rd IEEE International Conference on Advanced Learning Technologies (pp. 294-295).

Magalhães, J.P., & Silva, L.M. (2010). Prediction of performance anomalies in web-applications based-on software aging scenarios. IEEE Second International Workshop on Software Aging and Rejuvenation (WoSAR).

Mavrommatis, G. (2006). Learning objects and objectives towards automatic learning construction. European Journal of Operational Research, 187(3):1449-1458.

Melia, M., & Pahl, C. (2009). Constraint-based validation of adaptive e-learning courseware. IEEE Transactions on Learning Technology, 2(1):37-49.

Mimirinis, M., & Bhattacharya, M. (2007). Design of Virtual Learning Environments for Deep Learning. Journal of Interactive Learning Research, 18(1): 55-64.

Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. Int. J. Artif. Intell. Ed. 13(2-4):173-197.

Moodle (2013). The Moodle Community Web Site. Retrieved November, 14, 2013, from: http://moodle.org

Murray, S., Ryan, J., & Pahl, C. (2003). A Tool-mediated Cognitive Apprenticeship Approach for a Computer Engineering Course. International Conference on Advanced Learning Technologies ICALT'2003 (pp. 2-6). IEEE Press.

Narwani, A., & Arif, M. (2008). Blackboard Adoption and Adaptation Approaches, in M. Iskander (ed.), Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education (pp. 59-63).

Pahl, C. (2003). Evolution and Change in Web-based Teaching and Learning Environments. Computers & Education. 40(1):99-114.

Pahl, C. (2008). Content-driven Architecting of E-Learning Applications. Advanced Technology for Learning. 4(1).

Palmer, S., & Tulloch, W. (2001). The Evolution of Online Teaching and Learning in Engineering at Deakin University. Journal of Computing in Higher Education, 14 (1).

Reynard, R. (2008). Changing Designs of Online Learning: The Evolution of Digital Learning Systems Through Customization. THE Journal. Retrieved November, 14, 2013, from http://thejournal.com/Articles/2008/07/16/Changing-Designs-of-Online-Learning-The-Evolution-of-Digital-Learning-Systems-Through-Customization.aspx?Page=1

Robbins, S.R. (2002). The evolution of the learning content management system. Retrieved November, 14, 2013, from: http://www.learningcircuits.org/2002/apr2002/robbins.html

Sharp, H., deSouza, C. and Dittrich, Y. (2010). Using ethnographic methods in software engineering research. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (pp. 491-492). New York, USA: ACM.

Smeaton, A.F. (1991). Using Hypertext for Computer-Based Learning, Computers and Education, 17(3): 173-179.

Smeaton, A.F., & Crimmins, F. (1997). Virtual Lectures for Undergraduate Teaching. Proc. ED-MEDIA'97 World Conference on Educational Multimedia and Hypermedia.

Sommerville, I., Rodden, T., Sawyer, P., Bentley, R., & Twidale, M. (1993). Integrating ethnography into the requirements engineering process. Proceedings of IEEE International Symposium on Requirements Engineering 1993 (pp. 165-173).

Taylor, R.N., Medvidovic, N., & Dashofy, E.N. (2009). Software Architecture: Foundations, Theory, and Practice. Chichester, UK: John Wiley and Sons.

Wu, W.-H., Chen, W.-F., Wang, T.-L., & Su, T.-J. (2010). A pedagogical Web service-based interactive learning environment for a digital filter design course: An evolutionary approach. Computer Applications in Engineering Education, 18:423-433.