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

Claus Pahl, School of Computing, Dublin City University, Dublin, 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: Change Impact Analysis, Interactive Learning Environments, Learning Technology System (LTS), Software Ageing, Software Development and Maintenance, Software Evolution, Web-based Teaching and Learning

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 faceted 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
Integrating XML Technologies and Open Source Software for Personalization in E-Learning
www.igi-global.com/article/integrating-xml-technologies-open-source/37502?camid=4v1a