Chapter 11
Intertwining E-Learning Technologies and Pedagogies at the System Design Stage: To Support Personalized Learning

Georg Weichhart
Johannes Kepler University, Austria

Chris Stary
Johannes Kepler University, Austria

ABSTRACT
Although a large number of e-learning systems for individual learning support exist today, many of them still deal with pedagogical issues in an isolated way. In contrast, intertwining interactive system features with educational concepts allows pedagogical designs that may be considered according to their educational rationale. However, pedagogical approaches also do not provide requirements for technologies; they rather consider tools and features as predefined design parameters. Taking an interoperability point of view allows focus on the interaction between the pedagogical and the technological systems. By interpreting technology and didactic approaches as systems and ensuring their interoperability, educators are able to adapt learning experiences and technological features in a way that the overall learning system becomes personalized. A key element of the described work is an architecture that captures the design elements from both progressive education focusing on individual learning support, and the enabling web-based e-learning technologies.

INTRODUCTION

Today, e-learning systems are developed to support the independent and individual acquisition of knowledge (Friedman & Deek, 2003). Individual and group learning processes may be facilitated through dedicated e-learning designs and implementations (Aqda, Hamidi & Ghorbandordinejad, 2011, Auinger & Stary, 2005). This includes didactic designs according to pedagogical objectives, as well as technological designs. However, recent research reveals a gap between the feature-oriented approach to e-learning software development, and education viewing technology as a black box (cf. Casanova, Moreira & Costa, 2011).

Designing and developing an e-learning system requires a team of developers that is not only capable of producing software, but needs to “be aware of process of curriculum development, technical pedagogical knowledge, (TPCK), content knowledge, the place for designing and delivering the course, the learners.” (Assareh & Hosseini Bidokht 2011, p.792). Although a large number of different pedagogical approaches have been developed so far, educators still need to acquire know how on software development to design a learning support system that grounds technological features on reflected didactic instruments (cf. Anderson, 2004, Pan et al., 2010). In the following an approach to bridge conceptual gaps between technology developers and educators is developed for designing individual learning support systems in a mutually informed way. It is based on identifying interoperable components that may be arranged according to certain situational contexts and designer needs.

Interoperability focuses on the interface and interaction between two and more independent systems that together form a larger system (Naudet et al., 2010; Guédria & Naudet, 2014). Although these system elements are not dependent on each other, all elements contribute to a greater common functionality of the larger system. Integrated systems, in contrast to interoperable systems, have a functional dependence, where the entire system fails as soon as a single element fails. Hence, in an integrated system the sub-systems are not independent. Interoperable systems are contrary to compatible systems, as these latter types of systems do not interfere with each other and therefore do not contribute to some higher larger system functionality or goal (Panetto, 2007).

In order to support the structured implementation of pedagogical approaches in e-learning, explicit representations of didactically valid designs and software features allow for transparent mutual relationships. While multiple approaches exist for the representation of such systems, to reach our objectives an approach providing an overview of modules and interfaces on both levels, the e-learning/pedagogy level and the software technology level, is required. We make use of an Enterprise Architecture language, as it promotes a layered and contextual perspective on usage- and stakeholder-relevant processes supported by software technologies.
Robotics Community Experiences: Leveraging Informal Design and Learning Experiences to Motivate Urban Youth in STEM
Kimberley Gomez, Debra Bernstein, Jolene Zywica, Emily Hamner, Ung-Sang Lee and Jahneille Cunningham (2016). Human-Computer Interaction: Concepts, Methodologies, Tools, and Applications (pp. 120-147).
www.igi-global.com/chapter/robotics-community-experiences/139033?camid=4v1a

A Critical Review of Social Screen Time Management by Youngsters in Formal Educational Contexts
www.igi-global.com/chapter/a-critical-review-of-social-screen-time-management-by-youngsters-in-formal-educational-contexts/223058?camid=4v1a