# Chapter 9
# Content Design Patterns for Game-Based Learning

**Dennis Maciuszek**
*University of Rostock, Germany*

**Sebastian Ladhoff**
*University of Rostock, Germany*

**Alke Martens**
*University of Education Schwäbisch Gmünd, Germany*

## ABSTRACT

*To address the lack of documented best practices in the development of digital educational games, the authors have previously proposed a reference software architecture. One of its components is the rule system specifying learning and gameplay content. It contains quest, player character, non-player character, environment, and item rules. Documented content design patterns can assist in the authoring of such rules. This paper reports on four studies that have collected quest, character, environment, and item design patterns by analysing a variety of media. A selection of the results is presented, as well as a discussion of how the patterns can be used in designing educational game content.*

## INTRODUCTION

When computer scientists look at the field of game-based learning, they investigate existing systems, their implementations and software architectures. One goal of system comparison and analysis is to find design patterns, which can be described as typical best-practice solutions (Alexander et al., 1977; Gamma, Helm, Johnson, & Vlissides, 1994). These are useful for system design because they are application-independent, or as the basis for interdisciplinary project work (Martens, Hambach, & Lucke, 2009) because they are easy to communicate.

Regarding the system design aspect, we have previously demonstrated that the fundamental software design pattern behind user-adaptive eLearning software also known as Intelligent Tutoring Systems (Harrer & Martens, 2007) can be transferred to adaptive educational games,

DOI: 10.4018/978-1-4666-1864-0.ch009

in particular the genre of computer role-playing games (RPGs). Links between eLearning conventions and game design conventions are important to implement a seamless integration of learning and gameplay (or to reduce discrepancies between these). Our observations led to a software architecture we have proposed (Maciuszek & Martens, 2011) for the development of adaptive educational games. Its design is described in Figure 1.

The main idea is that the in-game character sheet describing the player's avatar (or player character, short: PC) can act as a learner model, which is the basis for adaptation in Intelligent Tutoring. In a narrative game, the user interface part becomes a virtual world, and the pedagogical process steering component of an Intelligent Tutoring System can be mapped to a drama manager, i.e.: the interactive storytelling component (e.g. Crawford, 2005). Expert knowledge is represented in the rule system of the game.

On the implementation level, the four semantic components (grey boxes in Figure 1) are divided into communicating modules (rectangles in Figure 1). Rule system modules provide interfaces to the content data. For example, content data can consist of authored specifications of interactive adventures and more general rules detailing what a PC can be, do, and learn. Similar to Howard's (2008) analysis of narrative games, we divided adventure-related rules into four modules relating to quests, non-player characters (NPCs), environments, and items.

Content data is the place where software engineers and instructional and game designers meet and do interdisciplinary project work. A special type of patterns – content design patterns – promises to be of great help here. Semi-formal documentation of successful interactive, educational stories and game mechanisms can inspire the content designer to create meaningful learning experiences, while the software designer obtains specifications that are straightforward to implement.

This article provides an overview of four studies in which we collected and documented quest design patterns, character design patterns, environment design patterns, and item design patterns. The following four sections detail objectives, method, and results of each respective study, and point out how to use the results for the design of educational game content. We conclude by identifying future work and giving an example on how to combine patterns of all four kinds.

*Figure 1. Educational RPG architecture*