

Daniel Han

ISE599

Feb 19, 2004

Summary review of Software Architecture for Immersipresence

François, Alexandre R.J.

This paper presents a software framework for the concept of Immersipresence, an application that uses a variety of multimedia data streams that are interactive and immersive. The motivation for creating such a framework is to utilize properties that are found in the field of software engineering (i.e. robustness, scalability, modularity and etc.) and yet not compromise performance which is absolutely vital in the field of multimedia. Furthermore, this model presented stresses the ease of integrating different software modules into one encompassing module. The author describes the different components and functionality that are a part of the Software Architecture for Immersipresence (SAI) as well as a simple notational scheme used in designing the software. MFSM (Modular Flow Scheduling Middleware) is then described as an example that implements the SAI style.

At the core of SAI are two types of components called cells and sources. Cells basically where most of the work and processing of datum is done. Sources on the other hand, are the objects that the cells work on consisting of persistent data. These logically seem to have stemmed from object-oriented programming principles where methods (functions implementing algorithms) and objects (data structures) parallel cells and sources, respectively. In either case, the components can be utilized in a way that can be custom fitted to the design requirements and application. For example, as a small building block to a larger module, the sources can be connected an arbitrary number of cells. Thus the same functionality can be applied on a variety of data sources. If the processed data from one cell needs to be processed by another cell, which may have completely different functionality, then a direct connection between cells can be made as well. Furthermore, all these inter-connections between cells and sources are conveniently shown with a diagrams consisting of simple geometric shapes. Thus, the actual implementation is highly correlated to the diagrams easing in coding of the software, something quite important to the author.

A lot of the concepts presented in this paper are not entirely new. But what the author did well was apply these concepts to a specific area of computing with a high-level of performance and inter-operability as a key requirement. Within research and development, theory and abstract ideas are the focus of interest while actual implementation of ideas are left behind as an afterthought. The SAI is a tool that bridge the gap between theory and practice with a framework that allows the user to see tangible results much faster and the ability to modify the system without significant modification. With several applications that have already used SAI, I feel that the author has gained a certain level of success.