

“Software Architecture for Immersipresence”
by François

SAI is a software architecture model that enables the creation of applications performing distributed, asynchronous parallel processing of data to be easily designed, implemented, and then analyzed. It allows diverse software systems each with their own requirements to work together. This is important for interactive applications such as the processing and manipulation of video, audio, and data from interactive sources. SAI allows research to accomplish its goals of creating system to demonstrate proof of concept. It also follows good software design principles such as efficiency, scalability, extensibility, reusability, and interoperability. SAI brings the software models of Pipes and Filters and data centric models together. Pipes & Filters is used in multimedia streaming applications in which the storage, retrieval, transmission, and presentation of data is important. The downside to Pipes & Filters is that the system is limited by the slowest filter thus limiting its efficiency and it does not have any mechanism for communication between other streams. Data centric models focus on interactive data driven applications. Time and synchronization is important to applications so a structure called a pulse allows synchronous data to flow in the system. SAI also emphasizes parallelism of operations and thus it is multithreaded. Different types of data can be classified as volatile or persistent. Volatile data does not need to be kept in the system. Cells do the processing and sources contain the data for the cells. Cells can do filtering on the pulses collect only the relevant data required to do its task. SAI has the property that the system will scale linearly depending on the available computing power. MFSM, Modular Flow Scheduling Middleware, the open source implementation of SAI, allows for easy implementation of SAI by users. It consists of

implementation level classes, a set of software modules, example applications, and documentation. Various applications have already been implemented in this system. Examples include: Real time video segmentation and tracking, handheld mirror simulation, Music on the Spiral Array. Realtime, IMSC communicator, and a distributed collaborative game. All these applications demonstrate that SAI does indeed allow efficient, extensible, and modular applications to be created. This is a useful software architecture that is helpful for research applications and eliminates the need to recreate an underlying modular components of software.