UML MODELING OF SEMANTIC INFORMATION SYSTEM NETWORK

Evan Tsai, Gabriel Nunéz, Aleksander Milshteyn, Jorge Estrada, Adrienne Lam,
Sergio Mendoza, Neil Arellano, John Paul Adigwu, Sergio Beltran, Helen Boussalis, Charles Liu

Structures, Propulsion and Control Engineering (SPACE) Center
College of Engineering
California State University, Los Angeles
5151 State University Drive
Los Angeles, CA 90032 USA
1-323-343-5445
etsai@calstatela.edu

Abstract - *This paper focuses on the modeling of the Semantic Information System (SIS) Network with Ubiquitous Video Conferencing (UVC) multicasting applications in the Unified Modeling Language (UML) format. The UML diagrams present detailed visualizations and the representation of the SIS Network's architecture, component functionality and interdependency. The development of UVC multicasting applications is geared towards users who are in constant need of real-time interactive collaboration. It is best suited for educators, researchers, and team-project members – all of which can be referred to as SIS Network participants. SIS Network participants will objectize contents of their work to their respective Semantic Networks, which will generate a hierarchical tree structure to interrelate those contents based on their semantic meaning and relationship to each other. These user-generated contents can be accessed, updated, and shared with other Network participants.*

Keywords: Unified Modeling Language (UML), Ubiquitous Video Conferencing (UVC), Semantic Information System network, decentralized server, client-server model

1 Introduction

The Advanced Computation and Communication (ACC) team of the NASA-CSULA SPACE Center is focused on design and development of new tools for information dissemination for collaborative education and research[11]. The SPACE Center consists of faculty-led graduate and undergraduate students which are formed into specific teams based on particular areas of research.

The current project objective is to design and implement an integrated framework for ubiquitous computing based on the Tuple Space paradigm [1]. The framework itself consists of two major components: Ubiquitous Video Conferencing which focuses on seamless communication between users across different platforms, while the SIS Network supports dissemination of information to provide a collaborative work environment.

Both components have a wide range of uses, which vary by its audience. The SIS Network is intended for targeting communities with similar interests, whether that community is in industry, education, or recreation. Combined with Ubiquitous Video Conferencing, the SIS framework is designed to be flexible and powerful for a wide range of uses accommodating a broader range of audiences.

UML (Unified Modeling Language) is geared for object-oriented analysis and design. It is used to model, document, and visualize every individual elements of an object-oriented system by displaying the results in different types of detailed diagrams which are different representations or aspect views of the project. UML is used in the industry by system engineers to visually model software-intensive and/or large and complex systems thus formalizing the organization of the project architecture.

The SIS Network is in its initial UML modeling stage. This visual organization is not only beneficial to the engineers but to the project management and newly hired workers as it accelerates the understanding of the inner workings of the project as well as a roadmap to the desired goal. One of the main features of UML is the ability to tie together and model how different engineering aspects interact with one another within the system such as an electromechanical implementation as well as being able to identify, distribute, and organize the workload between teams by the use and placement of the diagrams.

[1] Acknowledgement to NASA University Research Center Program, Grant # NNX08BA44A

The main tools used for this particular project are described as follows: Qt is an open-source software framework with project vital models of Phonon and QtMultimedia, FFMPEG encoder/decoder and OpenCV library for image processing. Qt is also an Integrated Development Environment similar to Microsoft Visual Studio; it is available as an open-source software and able to generate programs that are cross-platform compatible such as being able to run on different operating systems and different classes of computing hardware such as embedded systems and capable Smartphone devices. Phonon is a Qt module that is used for high-level multimedia manipulation; it is used specifically for playback of media files. QtMultimedia is another Qt module that is used for low-level multimedia manipulation; in our case the capture and streaming of real time audio. FFMPEG is an open-source multimedia software that is mainly used in the project to encode/decode audio and video streams.

2 SIS Network Architecture

A Tuple Space parallel programming paradigm was originally applied to an Aerospace Information Server (AIS), which uses a single-server-multiple-worker model to develop high-performance task management schemes and support parallel processing. This technology is the foundation for our current work and is applied to a distributed server scheme to increase the capabilities of the framework. The role of Tuple Space and other components in the framework is generalized in Figure 1. In the figure below, clients (and their compatible hardware) are situated and interfaced with the various applications on the bottom level. These applications connect and interact in the background with their respective servers that are built upon Tuple Space. Lastly, the servers themselves are synchronized with one another via Active Directory.

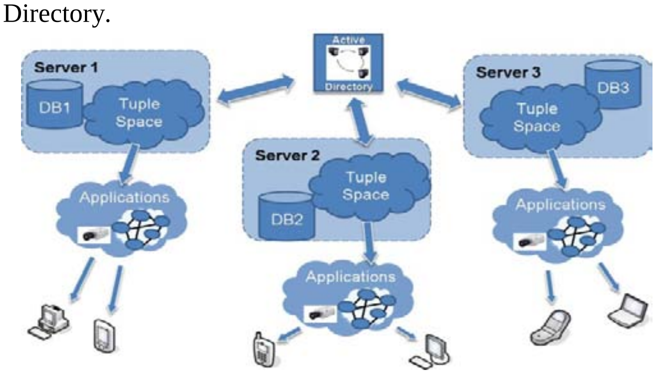

Figure 1: Overview of distributed server scheme and applications

The process model in Figure 2 highlights key areas of work that is necessary to create an integrated and fully functional platform. Only then can real-time network changes and multi-user interactions be supported. The Client-Tuple Space layer handles requests for application events and user authentication protocols. The Tuple Space-Active Directory layer ensures data synchronization across all servers in the distributed network. These include the Tuple Space-Fileserver layer and the Tuple Space-Active Directory Layer.

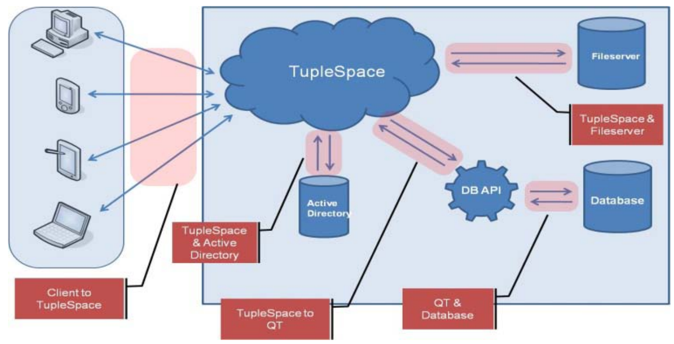

Figure 2: Application process diagram

The diagram shown in Figure 3 is a top-level view of the SIS Network using the Component model of a UML Diagram. The terms <<subsystem>> and <<component>> are parts of the UML standard called stereotypes, which is a variation of an existing system but in a different context. In the figure, each component is a software module for performing the functions of the SIS.

This model highlights the interdependencies between each component by its connections. The combined 'bubble-and-socket' symbol/notation in the figure represents a type of interface in UML. Each 'bubble' connection indicates that the component is providing a service while its associated 'socket' bracket indicates the component requires a service to properly function. In our applications, such interfaces are implemented in software. The small squares in the figure indicate port connections between the client software and server. In our design, such ports indicate physical connections to other physical components.

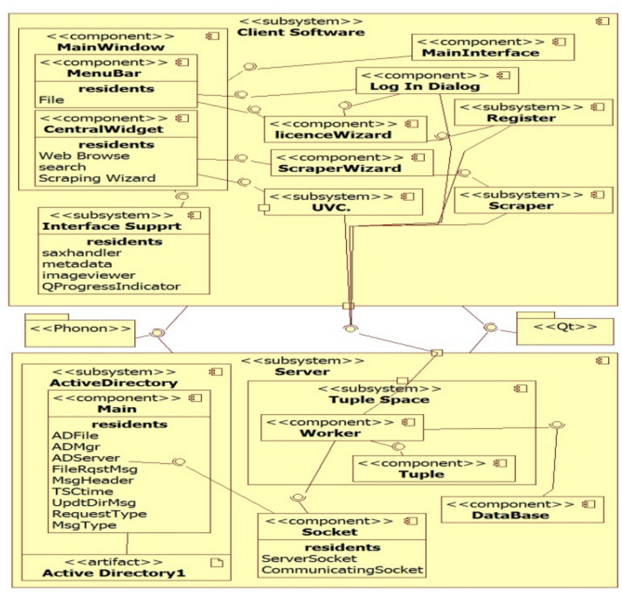

Figure 3: SIS Network Component Diagram

Complimentary to the Figure 3, Figure 4 presents the top-level behavioral view of the system. This model is a Use Case UML diagram which demonstrates the use of the software from the point of view of a user. As seen from in this diagram, the first feature observable to the user is the Main Window interface, which then connects to every other feature in the system such as login and registration, web browsing, the Scraping Tool, and the UVC.

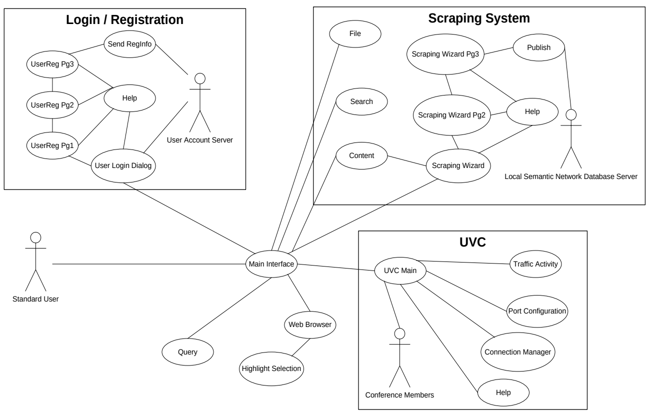

Figure 4: SIS Network Use Case Diagram

2.1 Model of SIS Scraping Tool

The Semantic Network development has focused on the incorporation of the Scraping Tool. The Scraping Tool provides users of the Semantic Network the ability to add new objects into the hierarchical tree structure by featuring drag and drop capability of object content, as shown below in Figure 5. Within this context, object contents can include text, image, video and/or audio files.

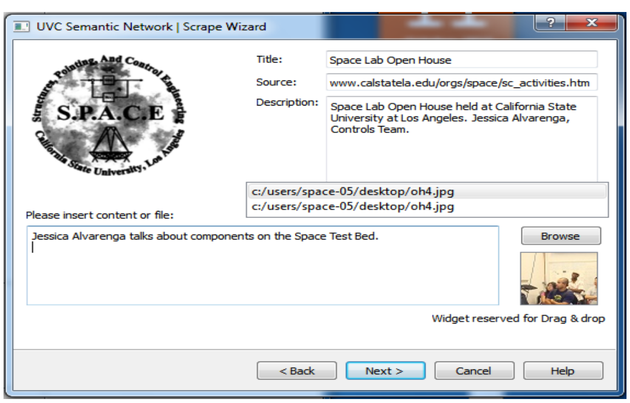

Figure 5: Scraping Wizard, Content Input

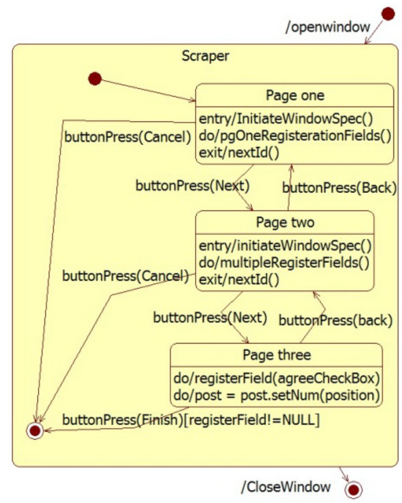

Figure 6: State Chart Diagram of the Scraping Tool (Top Level)

The Activity chart diagram in Figure 6 is a slightly more detailed behavioral model of a system, which describes the state of the system at any point and the transitions made to change its current state. This diagram is powered by 'states', which are tasks that are done sequentially. For example, the solid circle on the outside of the box denotes the start of a specific sub-system. The smaller solid circle with multiple inputs in the middle of the diagram represents a junction node where it can fork out to single connection based on input conditions, which are known in UML as 'guards' or denoted as square brackets.

Each state can contain action primitives inside such as: 'entry/', 'do/', and 'exit/' in an Activity Chart UML diagram. An 'entry/' action implies that a section of code will only be executed once when it enters the state (e.g. such as initialization routines). A 'do/' action shows that this section of code is to be looped continuously. Lastly, the 'exit/' action is another section of code that will only execute once just before it leaves the state. For our example of the Scraping Tool, the system revolves around four basic states. Each state is dictated by such actions as the click of a button to continue or cancel the publishing process, as well as requesting help for the current task.

For example, the 'Page 2' state in Figure 6 contains the bulk of the information necessary to publish new information into the SIS Network. An important feature to note at this point is the use of sub-state systems and the advanced features of drag and drop, which facilitates the process.

The Scraping Tool is the means to populate the Semantic Network [3]. This component facilitates scraping of information, objectized tagging, and the generation of metadata. The latter two are important for increased relevancy when searching for information within the Network.

The hierarchical tree structure is comprised of the QTreeWidget class, which provides a predefined tree view model [4]. Object IDs are assigned from the PostgreSQL Database upon execution of a SQL insert statement. This returned object ID is used to name the object directory of the newly published content. Similarly, the metadata file created is also saved as 'objectID.xml'[5].

The Scraping Tool is implemented using several classes within the Qt Nokia cross-platform environment. QWizard provides standard core functions associated with the development of 'form' design. These include field, pages, error-checking subclass functions, enabling the creation of a fully functioning registration tutorial/wizard.

The 'Scraping Wizard' sequence is displayed (Figure 7) when the 'add new object' button is triggered in the main application window. The sequence is such that a series of pages are displayed in specific order to provide proper navigation through the object creation process. Unlike many other classes within Qt, QWizard manages the allocation and temporary storage of user input within the forms. After the Wizard process is completed, the registered QWizard fields are available outside of the QWizard class for further information processing.

The first layer of the Scraping Wizard gathers object positioning information for new data to be placed in the tree structure. The relative positioning tracking is made available via QTreeWidget, which inherits several core classes from QTreeView. To keep track of object positioning, QTreeWidget::currentItem() is selected as a pointer to QTreeWidgetItem, an item within the tree structure. Depending on the position selected, a function call to QWizard::ItemAbove, QWizard::currentItem, QWizard::ItemBelow, and QWizard::insertTopLevelItem allow for the insertion of new objects into the desired position within the tree structure.

Figure 7a: Scraping Wizard Interface – Object Placement

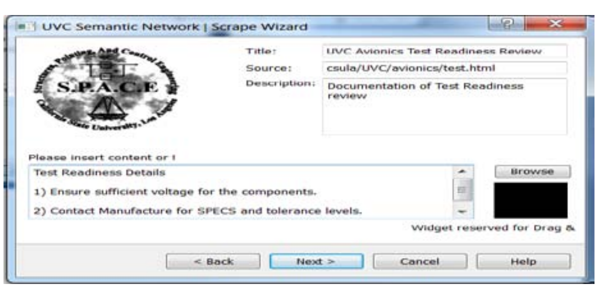

Figure 7b: Scraping Wizard Interface – Content Insertion

Figure 7c: Scraping Wizard Interface – Publishing Stage

The Scraping Wizard can detect if the network has an Internet connection. In the event of connection error, the addition of new information is still possible locally. Moreover, the newly created information is uploaded to the network when an Internet connection becomes available. This scheme is accomplished through a simple timestamp naming convention to object directory folders.

2.2 Model of SIS Ubiquitous Video Conferencing

The Ubiquitous Video Conferencing application is utilized over the SIS Network. The primary responsibility of the UVC is to provide SIS participants with flexible forms of communication such as visual, audio, text-messaging, and data transmission. Video and Audio transmissions are handled through UDP channels, while text and data are transmitted through TCP/IP.

Figure 8 displays a State Chart Diagram for one of the many features of the Ubiquitous Video Conferencing. When the SIS network participant clicks on the UVC tab in the Main User Interface, it goes into an idle state, doing its first initialization while continuously refreshing the GUI and checking for hardware changes in the user's audio and/or video input sources. At that idle state, the user can either start to transmit its audio/video and broadcast its existence within the SIS network or it can begin receiving other users' audio/video from a specified port.

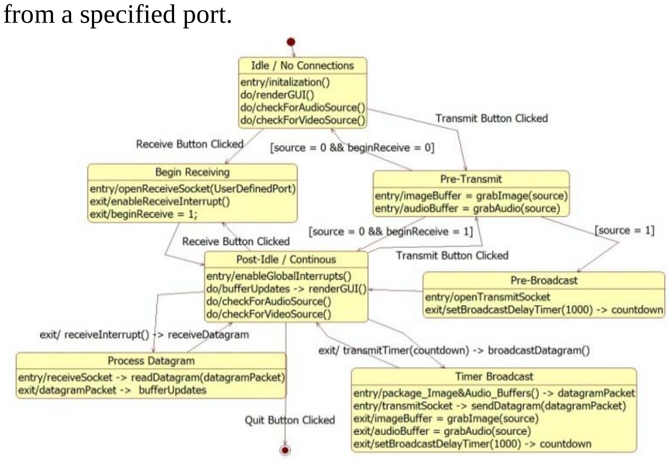

Figure 8: UVC State Chart Diagram

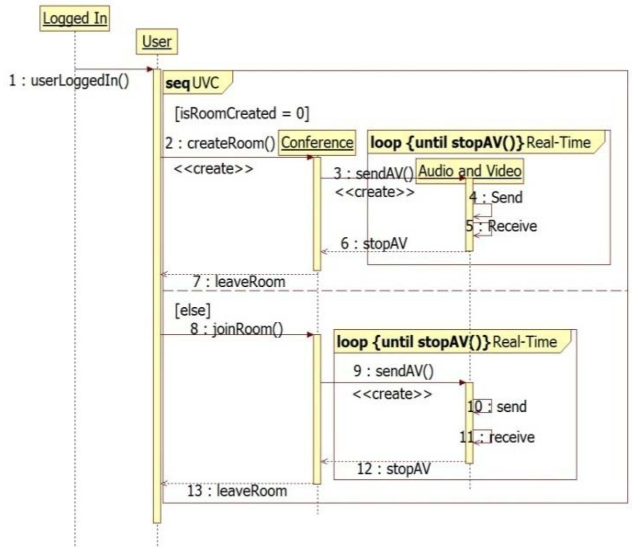

Figure 9: UVC Sequence Diagram

Figure 9 above is a Sequence Diagram that shows another different perspective of a top-level view of the Ubiquitous Video Conferencing, which can be interpreted as the 'layers' the program has to go through in order to perform the desired function. In this case, we show the steps the program goes through to create or join an existing Conference Room and chooses to participate by sending real-time audio and video data. The user logs in through the Main Interface, goes to the UVC tab, then if the user wants to create a room, the block statement creates a 'Conference' layer; then the user can decide whether to send audio and/or video or just idle along the 'Conference' layer. In this diagram, the 'seq UVC' frame is called a combined fragment in UML, which is a conditional statement that divides up the program path. For example, [isRoomCreated = 0] and [else] are fragments within 'seq UVC' box which perform different functions when being passed into the 'Conference' lifeline; either createRoom() or joinRoom().

3 SIS NETWORK DATABASE

PostgreSQL database was selected to perform metadata information storage [6]. Due to the hierarchical structure of objects on the network, a database capable of facilitating keyword searching across "parent" and "child" nodes is the dominant determining factor in appropriate database selection. These features include the ability of PostgreSQL to define data types and fully describe relationships and features, which are not available on traditional databases such as mySQL. In addition, there is a capability for "child" nodes to access the network to inherit "parent" attributes (data types which compose the parent object).

4 Conclusion

Further efforts in regards to Semantic Information System Network implementation will deal with integrating the Database and Tuple Space. Specifically, the addition of hash ID in Tuple Space provides recognition of a SQL defined request from network users and inter-server query tasks. The performance tests in real-time simulation will be conducted in order to provide accurate results on bandwidth and memory consumption. Different scenarios will be simulated in order to optimize SIS Network system.

5 References

[1] S. Balle, D. Palermo, "Enhancing an Open Source Resource Manager with Multi-Core/Multi-threaded Support," *Hewlett-Packard Company*, 2007.

[2] A. Alegre. "Aerospace Information Server." California State University Competition, 2008. Structures, Propulsions, And Control Engineering (SPACE) Center – Research Topic: Ubiquitous Computing and Embedded Architectures URL:http://www.calstatela.edu/orgs/space/sc_UbiquitousComputing.htm

[3] J. Alvarenga, H. Boussalis, "A Semantic Aerospace Network using Objectized Tags Exchanged through Decentralized Information Servers." HBCU/OMSI \ Aerospace Collaboration Conference 2009 Cleveland, Ohio, USA, July 2009.

[4] "QT Cross-Platform Application and UI Framework" URL: http://qt.nokia.com/

[5] R. Tolksdorf, F. Liebsch, and D. Minh Nguyen, "XMLSpaces.NET: An Extensible Tuplespace as XML Middleware."

[6] The PostgreSQL Global Development Group PostgreSQL 9.0.3 Documentation." URL: http://www.PostgreSQL.org/files/documentation/pdf/9.0/PostgreSQL-9.0-US.pdf