Towards an open collaboration service framework

Liu, Sandy; Spencer, Bruce; Du, Weichang; Chi, Chihung

Publisher's version / Version de l'éditeur:
https://doi.org/10.1109/CTS.2011.5928668
2011 International Conference on Collaboration Technologies and Systems (CTS), pp. 77-85, 2011-06-01

Towards an Open Collaboration Service Framework

Sandy Liu, Bruce Spencer
National Research Council
& University of New Brunswick
Fredericton, Canada
{firstname.lastname}@nrc.gc.ca

Weichang Du
University of
New Brunswick
Fredericton, Canada
wdu@unb.ca

Chihung Chi
Tsinghua University
Beijing, China
& University of New Brunswick
chichihung@mail.tsinghua.edu.cn

ABSTRACT

This paper introduces an open collaboration framework called SAVOIR, which supports collaboration for distributed groups to work together with open sets of heterogeneous resources. This framework is built on the concept of service computing and cloud computing with focus on the openness and accessibility. The set of resources managed in this framework is open; it includes infrastructure services, platform services, and software services. SAVOIR aims to deliver Collaboration as a Service (CaaS). It introduces a generic approach to registering and interacting with distributed resources; it also illustrates a general mechanism in defining and coordinating resources and people into effective collaboration sessions.

KEYWORDS: Open collabortion, Coordination of Collaboration as a Services, SOA, serivce interoperability.

1. INTRODUCTION

Collaboration can be found in almost every context in life today. The prevalence of the Internet, the growing availability and the decreased cost of bandwidth and computational power have made collaborative work dramatically easier, and opened up many new ways to collaborate. Built on concept of Software as a Service (SaaS)[1], Platform as a Service (PaaS)[2], and Infrastructure as a Service (IaaS)[3], the emerging concept of cloud computing is breaking another new ground for collaboration. These services and the ultimate connectiveness allow people to share ideas, knowledge, and skills widely and nearly instantaneously.

Nowadays many people are working with remote collaborators under the notion of Virtual Organization (VO) - the flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources [4]. Forming a VO is considered to be a means to help address critical resource, personnel and logistical issues. There is typically a 30% to 50% increase in productivity as result of implementing VOs [5]. Successful collaboration does not just happen; most of the collaborative work requires much coordination effort from start to end.

As many systems are built independently in silos, data are scattered, and people are located in different places. It is difficult to collaborate effectively under such a situation, where the collaborators and resources are distributed and may not have been designed to work together. The required coordination effort behind the scenes can span from human-to-human, human-to-device, device-to-data, application-to-application, application-to-device, device-to-platform, platform-to-network, and organization-to-organization - let alone other human factors concerns. In general there is a lack of well-defined collaboration models and service models that can support the emerging collaboration practices. This paper outlines an open collaboration framework, where we introduce the concept of Collaboration as a Service (CaaS), specifically identifies what kind of services are required and provides a mechanism to allow these heterogeneous services to work effectively under the notion of cloud computing.

2. GENERAL COLLABORATION DEFINITION

2.1 Concept of Collaboration

Since collaboration is so prevalent in human society, it is receiving an increasing amount of attention from a wide range of disciplines. Domain specific silos often impede the capacity to discover and synthesis research results, which makes it particularly challenging to develop a cross-disciplinary theoretical framework for collaboration that goes beyond a dictionary definition in order to use collaboration as a problem solving strategy in diverse contexts [6].

Collaboration is defined as *the action of working with someone to produce something* or *something produced in collaboration with someone* [7]. Thus, it can refer to the collaboration behavior, the collaborating process, the result of collaboration, and so on. The dictionary definition gives little distinction between collaboration, cooperation, and coordination. Elliott has proposed a generalized framework marked by the distinction between coordination, cooperation, and collaboration, with each activity being dependent upon the previous, while addressing increased levels of complexity [8]. In this paper we consider collaboration more precisely as *the process where two or more collaborators work jointly in coordinated activities toward a set of shared goals.* Furthermore, we consider the collaboration process (P) consists of one or more **collaborative sessions (CSs)** where each session involves one or more collaborators conducting activities collaboratively:

$$P = \bigcup_{i=1}^{n} CS_i$$

Fig.1 below illustrates an example set of collaboration sessions in a specific collaboration; each happy face denotes a collaborator. These sessions can be held synchronously or asynchronously depending on the needs and availability of the collaborators and resources. Multiple sessions can be held at the same time, e.g. CS_1 and CS_4, or with overlapping periods, e.g. CS_5 overlaps with CS_2, CS_3, and CS_6. The same collaborator can participate in multiple sessions, as shown in CS_2 and CS_5, where two collaborators are in both of the sessions. However, these sessions are coordinated, and the aim of these sessions is to work on tasks or activities contributing towards the shared goal(s); coordination is a fundamental enabling requirement for all collaborative activities. It is to be considered that over time, with the aggregated results from all of these sessions, the collaboration goals can be fulfilled.

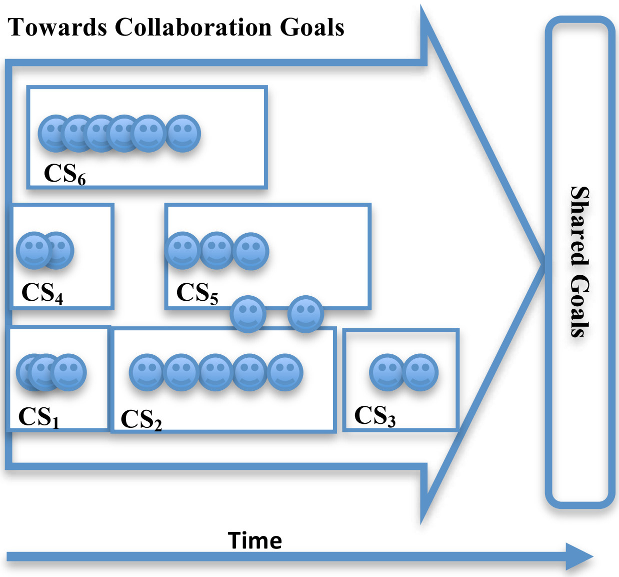


Figure 1. Collaboration Sessions (CSs)

To help understand what needs to be coordinated, we define the key interacting components of collaboration in the following section. With these key components, we can better analyze what qualifies as collaboration and what are the defining principles or elements of this process.

2.2 Components of Collaboration

In this paper, we defined seven basic components for collaboration, listed as follows:

1) *Collaboration Outcome (O)*
The Collaboration Outcome includes the end product and everything created in the process of the collaboration. It can be tangible or intangible such as a report being developed, mutual understanding, consensus on certain issues, meeting minutes, design documents etc.

2) *Collaboration Goals (G)*
The Collaboration Goals are the common goals that are all the collaborators are aiming to achieve. These goals may not be clearly identified in the beginning, but one of the collaboration goals could be to develop the shared understanding and identify emergent goals.

3) *Collaborators (U)*
The Collaborators are the team members who participate in the collaboration activities. At least two collaborators are required in order to collaborate.

4) *Resources (R)*
The resources are anything that supports the collaboration and can be accessed by the collaboration team. A resource may not be globally available for everyone. The resources include data, information, software, hardware, tools, services, networks, and platforms provided by other person(s) or objects. Note that the collaboration outcome of one collaboration session can become resource for another session. Two people can potentially collaborate face-to-face without involving any resource and in this special case, the resource set is empty, however, some additional resources are required for most collaborations.

5) *Activities (A)*
Activities are the actions taken towards completing a task, where a task is a piece of work towards the goals. An activity can be performed by one or more collaborators with or without the support of other resources. The outcome of an activity can become a collaboration outcome. A collaboration activity can be carried out in a number of collaborative sessions.

6) *Communication (E)*

Communication is essential for collaboration. It refers to act of exchanging information and messages among collaborators and various collaboration components. Note that communication refers to the action of information exchange, but it does not include the resources, such as the video conferencing software, that enable the communication.

7) *Coordination (C)*

Based on the coordination theory, coordination is defined as the additional information processing performed when multiple, connected, actors pursue goals that a single actor pursuing the same goals would not perform [9]. Coordination brings different elements of collaboration into a relationship, facilitates communication and interactions among different collaboration components in order to ensure harmony among collaborators, and that the activities to be carried out effectively with the required resources.

To summarize, we define collaboration process ***P*** as a 7-tuple:

$$P = < O,\ G,\ U, R, A, E, C >$$

where $O, G, U, A, E, C \neq \varnothing$, and $|U| \geq 2$.

As we claim that collaboration can be decomposed into a number of CSs, for each CS_i, it can be seen as a mini-collaboration, which can also be defined with this 7-tuple:

$$CS_i = < O_i, G_i, U_i, R_i, A_i, E_i, C_i >$$

where $G_i, U_i, A_i, E_i, C_i \neq \varnothing$, and $|U_i| \geq 2$.

Note that in each CS_i, we allow outcome O_i to be empty. Also, one may expect collaboration goals G can be decomposed into sub-goals and each CS can achieve some of them, thus the union of G_i denotes the overall collaboration goals G. In our opinion, this is not a realistic way to model goals, as they may not be easily decomposed into discrete sub-goals.

Also note that in this paper, we are only concerned with tangible and quantifiable components; components such as trust are very important but they are not being represented here.

We further define what open collaboration means in this context. First of all, the membership of collaboration team is open, which means a collaborator can come and go as s/he wishes. Second, the set of resources used for collaboration is open, verses being fixed; resources can be added or removed. Third, the collaboration goals may not be clearly defined, and can be created and altered over the course of collaboration. As a consequence, the resulting activities vary depending on what needs to be achieved. Fourth, and most important, the context of collaboration is open, meaning that the domain that the collaboration is related to is also open.

With this openness in mind, we develop a framework that can provide collaboration as a general means of problem solving for various domains. We can then leave the domain experts to decide what is needed, and how collaboration should be conducted by letting them define their own agendas in each collaboration session.

3. COLLABRATION AS A SERVICE

With the growing popularity of service computing and cloud computing, we start to look into the possibility of delivering Collaboration as a Service (CaaS). If that is possible, what are the requirements of such service and how to fulfill the requirements, and evaluate the service? Furthermore, what makes CaaS differ from traditional collaboration systems?

One may argue that collaboration is so complex and that it is impossible to deliver "collaboration" per se as a software service - the interactions and contributions by collaborators and resources cannot be replaced by system services. However, the collaboration service can be seen as a service that facilitates the very process of collaboration. Just like a wedding service, it can only deliver the wedding process but not the wedding itself, which requires the active participation of different actors.

CaaS stems from the ideas in Computer Supported Cooperative Work (CSCW). CSCW's main concern is about "how collaborative activities and their coordination can be supported by means of computer systems"[10], where CaaS deals with collaborative activities and their coordination by means of computer services. When the delivery mechanism changes from system to services, there are at least two main implications to CaaS. First, CaaS is supported by independent services, and each is autonomous on its own. Second, not only does CaaS need to aggregate functionalities of various services that provide direct functions required for collaboration, but it also needs to compose services at the vertical dimension to combine infrastructure services, platform services, and domain function services into a cohesive whole.

There are many aspects to collaboration, as defined in the previous section, it spans from coordination of collaborators, resources, to defining tasks, activities, keeping track of the progress and records of the collaboration. Assuming each of these elements or components is available as a service in the cloud somewhere, we can then bring them together as a collective service under the notion of CaaS. Such a service will take the context information and requirements of collaboration as input, construct the components defined in our collaboration model, and facilitate the collaboration process towards the collaboration goals. Based on the high-level 3C model [12], collaboration consists of three key concepts, cooperation, communication, and coordination, as shown in Fig. 2.

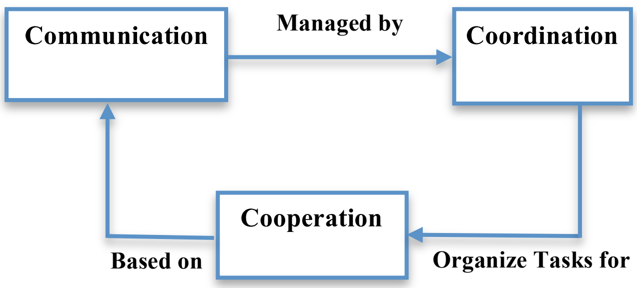


Figure 1. The 3C Model of Collaboration [11]

Since the actual communication actions and cooperation actions are conducted by collaborators within the course of collaboration, the CaaS service will focus on the coordination service, which is the core enabler for collaboration. This paper outlines how CaaS is being designed and realized by the coordination service in an open framework.

When we utilize cloud services to provide network, storage, platform, and software applications for collaboration, there are a number of challenges that need to be addressed. The following are just a few major issues:

1) Registration of different types of services, these services are dynamic and may or may not be available at the time of CS. How such services are being described and retrieved?

2) How a session is being defined? What should be included in this description? And how to select and reserve these heterogeneous services?

3) At runtime, how to coordinate different instances of resources and collaborators into CS in which participants and resources are all distributed.

This paper proposes a generic approach to address these issues and introduce a middleware framework called SAVOIR that implements this idea. The following section will describe the overall design of SAVOIR and methodology used to coordinate these services for collaboration. We then use the component model to evaluate if the resulting CaaS can deliver and manage those key components.

4. SAVOIR OVERALL DESIGN AND IMPLETATION

4.3 Terminlogies

SAVOIR stands for **S**ervice-oriented **A**rchitecture for **V**irtual **O**rganization **I**nfrastructure and **R**esources. In SAVOIR, we use the term ***Edge Device (ED)*** to refer to an entity that can be controlled digitally and is able to provide some functions for collaboration. The word "edge" is used here to indicate the fact that these services are most likely be provided by third-parties in the form of *SaaS*, *PaaS*, or *IaaS*. For example, a visualization software to be used in a collaboration session can be an ED. In order for a heterogeneous ED to communicate with the core CaaS services, we put a service layer called ***Bus Interface (BI)*** in between the ED and the CaaS management services. We use the term ***Edge Services (ESs)*** for EDs equipped with a BI. The word "edge" is used here to indicate the fact that these services are most likely be provided by third-parties in the form of *SaaS*, *PaaS*, or *IaaS*. For EDs powered by SaaS, each of them supports a set of device specific configurations.

4.4 Overall Design

Fig. 3 depicts the high level design in SAVOIR. It includes a set of Edge Services (ESs) that provides functional services, platform services, and infrastructure services to SAVOIR. We use a configurable service bus to handle interactions among various services. Across the bottom, we have the User Interface that generates corresponding service interface to collaborators and administrators. The communication service provides the communication media for CSs.

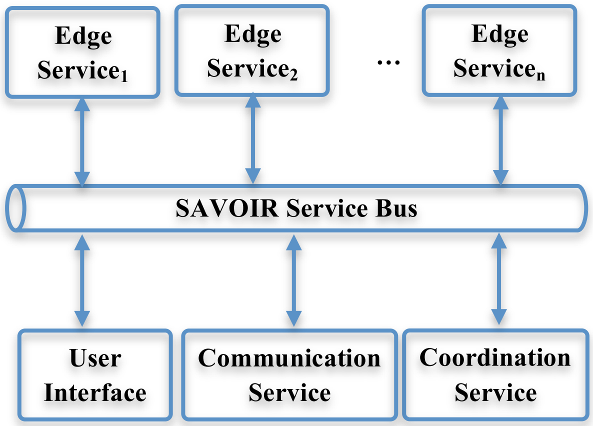


Figure 2. High Level Services in SAVOIR

Fig. 3 depicts the high level design in SAVOIR. It includes a set of Edge Services (ESs) that provides functional services, platform services, and infrastructure services to SAVOIR. We use a configurable service bus to handle interactions among various services. Across the bottom, we have the User Interface that generates corresponding service interface to collaborators and administrators. The communication service provides the communication media for CSs. As shown in Fig. 4, the Coordination Service contains a set of management services acting as the control center for the collaboration service. We will explore the functions of different types of management services that support the coordination service in the following sections.

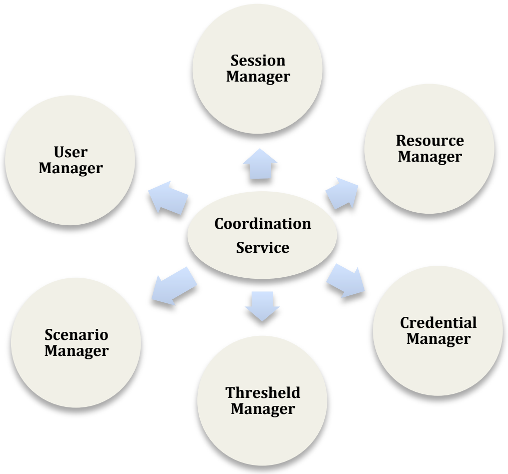


Figure 3. **The Management Services that support the Coordination Service**

4.5 Edge Service Registration

Before any communication can occur between a new device and SAVOIR, the device must be registered as an Edge Service. As a prerequisite the device must be connected to a Bus Interface, which may have to be built or adapted to work with the new device. SAVOIR has provided a set of reference implementation libraries to assist developers in instantiating a new Bus Interface. Once a device-bus interface pair has been set up and is running then the workflow is as follows:
Step 1: the process is initiated using the "add service" tool within SAVOIR. This is a web form that requires the input of a new service name, the selection of the protocol the target bus interface uses (such as HTTP or JMS) and the authentication model the service will use, etc.
Step 2: The "add service" tool generates an XML registration ticket containing the entered information, and submits it to SAVOIR.
Step 3: SAVOIR returns a new unique Resource ID for this edge device, and sends the registration ticket to a system administrator for processing.
Step 4: The SAVOIR administrator performs any required configuration (opening network ports, creating JMS topics, etc.), then sends a completed copy of the registration ticket to the service contact as an attachment on an email sent to their submitted contact address.
Step 5: The bus interface is configured with its required parameters, and sends SAVOIR an initial profile message. The receipt of this profile message completes the edge device registration process.

The profile message is in XML form. It provides all the information required to interact with a particular edge service. It has two parts, one is used for SAVOIR to create a corresponding widget for users to get direct access to such a service; the second part of this message is to include the operational information of such a service, for example, the service name, ID, supported activities, and corresponding parameters. It is the responsibility of the edge service to keep this information up to date by sending a <profileMessage> message. This must be done when a new service comes online and every time its profile changes. SAVOIR can also force an update. This is handled through a simple request/response message dialog. SAVOIR will request the edge service send a profile for an activity by sending a message with a "getProfile" action. The edge service will respond by reporting that profile. SAVOIR also uses the information in the profile for creating the network connection to the Edge Service.

4.6 Defining Collaboration Sessions

A Collaboration Session denotes a group of collaborators working on coordinated activities with the support of some resources. Thus it consists of a set of Edge Services and collaborators with transitional rules specifying conditions to move from one activity to another. Each Edge Service is mapped to one to many collaborators. A session contains a temporal attribute that indicates the start time and end time, the required network connectivity such as bandwidth, and the supported platform, as well as the collaboration workflow.

The collaboration workflow can be defined by a scenario, which is acting as the template for a session, only that it will not have the temporal factor, and the collaborators information. A scenario has to be instantiated with the additional information to make it a concrete runnable collaboration session.

SAVOIR provides a graphical authoring interface for users to define scenarios and sessions. Imagine two architects are working on a design, the collaboration flow can be set as using a 3D design Edge Service first, when the design is "saved", the work will be sent to a rendering Edge Service for processing, and the processed result will then be displayed to the collaborators using a virtualization Edge Service.
There are two separate and distinct steps in the message flow between SAVOIR and the edge services in a session. Before a session can be run, all the edge services used in that session must be initialized by authenticating the users and loading with the right set of parameters. Once all the services have successfully initialized, the session will run based on the description.

The following table lists each message flow and the supported actions in a message. Note that the Edge Service is required to send an "acknowledge" message back to SAVOIR for every message it receives.

Message Flow	Message Action(s)
Initializing a Session	acknowledge authenticate load
Running a Session	acknowledge endSession getStatus pause reportStatus resume setParameter stop start startResponse

Table 1. Edge Services Communication Messages and Actions

The transition rules are defined by a set of production rules. Each edge service can be seen as a node, and the link between two nodes is defined by a set of embedded rules. These rules are then translated into JBoss Drools rules [13] to govern the progress of a collaboration session at Runtime.

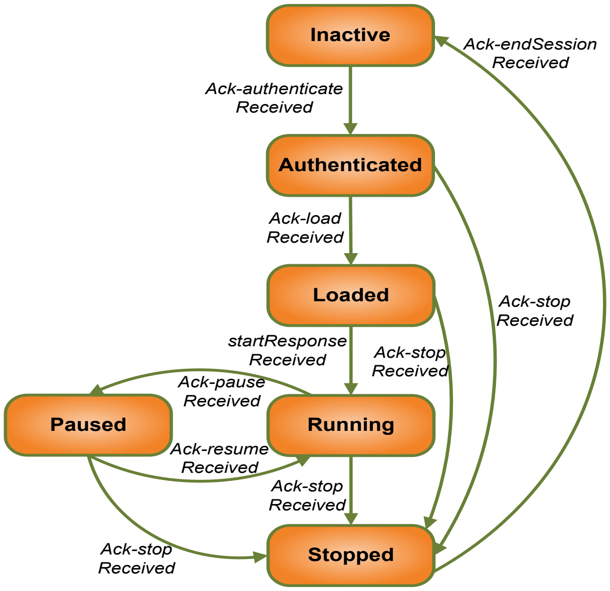
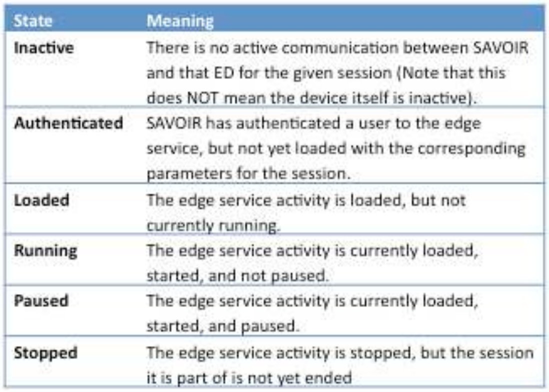


Figure 4. States of Edge Services

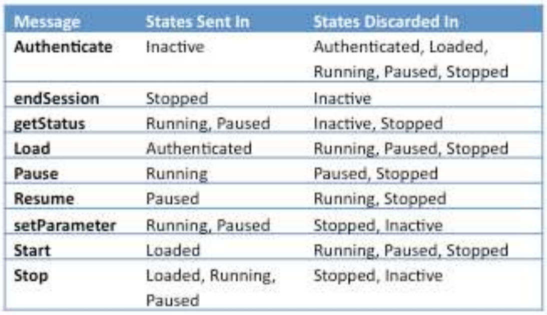
Runtime coordination for collaboration session is not a straightforward task. SAVOIR uses a set of management services to coordinate the runtime sessions. These Managers are implemented as Web Services to interact with an Enterprise Service Bus (ESB). In general, messages are sent to SAVOIR over the ESB and the management services determine what action, if any, is required from the message and performs the appropriate action, sending messages over the ESB to appropriate edge devices. Note that all communication in the system is through the ESB and controlled by the managers. The only exception is communication between the managers, which uses inter-process communication techniques to improve the efficiency of the system. The following outlines the responsibilities of the key management services in the core of SAVOIR.

Resource Manager (RM): The resource manager is responsible for all direction and long-term state of resources (i.e. edge services). Each instance of a resource is represented by a state machine. There is a set of Message Sequencing Rules to ensure proper ordering of messages; the RM maintains a general set of rules governing the states in which it is proper to send certain kinds of messages. Fig. 2, Table II and Table III illustrate the message sequencing state machine and the corresponding rules. These rules are scoped to a single session. That is, each and every session gets an instance of this rulebase to manage its message sequencing. A reference to this session's ID is provided as a global for this knowledge base. Within a session, there are two types of facts asserted into this knowledge base. The first is the running state of a single instance of an edge service. The second type of fact is a message that is queued to send.



State	Meaning
Inactive	There is no active communication between SAVOIR and that ED for the given session (Note that this does NOT mean the device itself is inactive).
Authenticated	SAVOIR has authenticated a user to the edge service, but not yet loaded with the corresponding parameters for the session.
Loaded	The edge service activity is loaded, but not currently running.
Running	The edge service activity is currently loaded, started, and not paused.
Paused	The edge service activity is currently loaded, started, and paused.
Stopped	The edge service activity is stopped, but the session it is part of is not yet ended

Table 2. Description of states



Message	States Sent In	States Discarded In
Authenticate	Inactive	Authenticated, Loaded, Running, Paused, Stopped
endSession	Stopped	Inactive
getStatus	Running, Paused	Inactive, Stopped
Load	Authenticated	Running, Paused, Stopped
Pause	Running	Paused, Stopped
Resume	Paused	Running, Stopped
setParameter	Running, Paused	Stopped, Inactive
Start	Loaded	Running, Paused, Stopped
Stop	Loaded, Running, Paused	Stopped, Inactive

Table 3. Message handling Rules

Session Manager (SM): The SM Manages SAVOIR sessions: authored and unauthored. Authored sessions include a choreographed interaction of resources (a scenario) - managing this scenario is delegated to the Scenario Manager that will be described below.
The SM also manages Infrastructure and platform services reservations, and delegates that specific tasks to the scheduling services. The SM also contacts RM for the loading and unloading of resources included in a session.

User Manager (UM): The UM is responsible for user authentication, authorization, and user information management for all SAVOIR users and groups.

Credential Manager (CM): The CM is analogous to the password store in a Web browser, or the OS X keychain. It implements a repository of stored authentication credentials for edge services, as well as managing which users have authorization on which resources, providing management for both of these facilities.

Scenario Manager (ScM): The ScM creates, stores, and runs scenarios. It is responsible for directly handling scenarios. It includes code for parsing of scenario XML, generating rulebases, and loading and running those rulebases in the context of an authored session. Again, a scenario is the template for an authored session - it includes a set of resources, and rules to orchestrate their interactions.

Threshold Manager (TM): the TM handles all the incoming and outgoing messages. It routes incoming messages to the appropriate services and sends outgoing messages from SAVOIR to the ESB, while handling message sequence, logging, and waiting for acknowledgements.

Within a collaboration session, each user may be interacting with a different subset of services. SAVOIR employs a 3-section Session ID in the form of "xxx-yyy-zzz" to help in coordination:

- xxx - represents the master SAVOIR session ID
- yyy - represents the unique ID to identify the current instance of the Edge Service (ES)
- zzz - represents the unique ID to identify the USER in the session

With these three pieces of information, the edge service can easily map a particular instance of his/her device to a particular user.

Here's an example of an in-session message SAVOIR sends to an ES - the "authenticate" message:

```
<message ID="327" action="authenticate" sessionID="2594-
2001-3001">
   <service ID="133" activityID="1"
   activityName="Healthy patient 3G"
   servicePassword="nosm" serviceUserID="nosm_user"/>
</message>
```

In this case, sessionID "2594-2001-3001" uniquely identifies user, service instance within the SAVOIR session "2594".

To further illustrates how this mechanism works, let's look at a few more examples.

Example 1: Single user logs on to a service with session ID 101, instance ID 202, and user ID 303. This session is entirely its own and no other session or instance has access to it.

Example 2: Several users are in the same session working their way through their own version of an ES. The mapping would be as follows:
User 1: session ID 101 instance ID 202 user ID 315
User 2: session ID 101 instance ID 203 user ID 316
User 3: session ID 101 instance ID 204 user ID 317

Example 3: In a session with multiple users, one of the users, for some reason, has two instances of an ES running:
User 1: session ID 101 instance ID 202 user ID 315
User 2: session ID 101 instance ID 203 user ID 316
User 2: session ID 101 instance ID 204 user ID 316
User 3: session ID 101 instance ID 205 user ID 317

Thus, this method allows for a very flexible means of combining sessions, users, and instances and preserves the mapping between SAVOIR and edge devices since each fully qualified SessionID is unique. This mechanism also allows SAVOIR to deliver the right message to the right user. As each user subscribes to system wide messages, session-specific messages, and user-specific messages.

5. APPLICATIONS OF THE FRAMEWORK

We aim to build SAVOIR as a generic framework to allow users to use collaboration as a general means to work on domain specific projects and problems. One way to test it is to actually use the framework for various projects with different virtual organizations. Our current framework has been tested in two use cases with very different usage domains and contexts.

The first use case of SAVOIR supports architects, urban designers, and engineers to work on modeling, designing, and developing ideas for urban renovation plans, building designs [14][15]. The collaboration service enables multiple users from distributed geographic areas to work in synchronous collaborative sessions. For each collaborative session, SAVOIR invokes a set of distributed services to support design, rendering, virtualization, and communication. Additionally SAVOIR can also issue calls to lightpath network infrastructure services when the bandwidth requirement for application level services exceeds what a commodity network can offer.

The second use case of SAVOIR supports a Health Services Virtual Organization (HSVO). One of the main functions of HSVO is to provide training for medical school students and allowing them to work with students in different schools as teams under certain medical scenarios. SAVOIR enables the sharing of simulation equipment across a high speed network to enable learners to benefit from equipment not at their site. SAVOIR allows trainers to create advanced learning scenarios with multiple simulation tools based on the real-time performance of the learner. For example, students can work on a text-based medical scenario, and put the conditions to a medical mannequin to validate the actions chosen. If the condition improves, then work on a different problem, or bring up a 3D body simulator as a reference to help further understand the problem. SAVOIR also enables training to be given when the trainer is in a different location to the learner, an example being a dissection occuring in McGill University, a trainer in Sudbury and learners logging in a number of other locations all seeing the same or different resources depending on the activity design [16].

6. RELATED WORKS

Many systems and frameworks claim to be collaborative systems, yet most of them only support certain aspects of collaboration. For example, Cisco's unified communication systems[17], Tandberg[18], and many chat room or IM systems are mostly focusing on providing the communication media for distributed collaborators. Another category of collaborative system is focusing on providing concurrent editing or manipulation to shared objects, wikis and most of the online gaming systems are examples. CSCW systems, often being considered as groupware systems[19], provide a more integrated environment for collaboration, but most of them exist as monolith systems, and users often have steep learning curves to get familiar with the environment to become productive. Furthermore, these systems are mostly closed systems, meaning that the set of tools and functionality provided by the system are fixed, making it non-trivial to add new components to it. A typical example in this category is Lotus Notes[20]. These kinds of CSCW system sometimes developed into Collaboration Project Management Tools (CPMT). CPMT are very similar to collaborative management tools (CMT) except that CMT may only facilitate and manage a certain group activities for a part of a bigger project or task, while CPMT covers all detailed aspects of collaboration activities and management of the overall project and its related knowledge areas[19]. Since the main function of these systems are focusing on project management, key features of these systems are the document repository, task allocation and scheduling (often using a GANTT chart), the actual coordination of activities are outside the scope of these systems. As collaborative tools evolve to embrace the web as the access portal. This gives rise to Web 2.0 collaboration platforms, such as many of the social network systems, Twitter, Facebook, GoogleWave, and so on. These tools provide open APIs for developing new tools, making them very flexible and adaptable for collaboration in social context, however, the stigmergic aspect of such system implies that the collaboration sessions are mostly self-organized with fairly straightforward activities, take Wikipedia as an example, co-editing is the primarily action being supported. Scripted sessions that require communications among tools and collaborators are not typical and largely not being supported.

7. CONCLUSION

This paper introduced the concept of open collaboration, and a framework built on such a concept. We also introduced the concept of Collaboration as a Service (CaaS) and demonstrated how such service differs from the services provided by traditional collaboration environments such as CSCW systems. SAVOIR builds on the context of cloud services, allowing "open" resource registration, user-defined collaboration sessions with scripted workflow that utilizes various levels of internal and external services. We also illustrated how vertical and horizontal services can be composed together to support effective collaboration for distributed expertise and distributed resources in this open framework.

ACKNOWLEDGMENT

This work is partially derived from the Health Service Visual Organization project funded by CANARIE. Thanks for every member in the team who contributed ideas and works towards this research.

REFERENCES

[1] Software & Information Industry Association, *Software as a Service: Strategic Backgrounder*, Software & Information Industry Association, 2001.

[2] Vishwas Lele, "Platform as a Service – When it comes to the cloud, PaaS is the point," *Fleeting Thoughts*, Dec. 2010.

[3] Darryl K. Taft, "Survey: Infrastructure-as-a-Service Adoption on the Rise - IT Infrastructure - News &

Reviews - eWeek.com," Aug. 2010.

[4] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid - Enabling Scalable Virtual Organizations," *International Journal of Supercomputer Applications*, vol. 15, 2001.

[5] L. Pang, "Understanding virtual organizations," *Information System Control Journal*, vol. 6, 2001.

[6] Mark Elliott, "Stigmergic Collaboration: A Theoretical Framework for Mass Collaboration," University of Melbourne, 2007.

[7] Oxford University Press, "Definition of Collaboration," *Oxford Dictionaries Online*. Accessed Dec. 2010.

[8] Wikia Contributors, "Collaboration theory," *Meta Collab*. Accessed Dec, 2010.

[9] T.W. Malone, M.I.O. Technology, and S.S.O.M.C.F.I.S. Research, *What is coordination theory?*, Citeseer, 1989.

[10] P.H. Carstensen and K. Schmidt, "Computer supported cooperative work: New challenges to systems design," *In K. Itoh (Ed.), Handbook of Human Factors*, 1999.

[11] F.F. Oliveira, J.C. Antunes, and R.S. Guizzardi, "Towards a collaboration ontology," *Proc. of the Snd Brazilian Workshop on Ontologies and Metamodels for Software and Data Engineering*, 2007.

[12] C.A. Ellis, S. Gibbs, and G. Rein, "Groupware: some issues and experiences," *Communications of the ACM*, vol. 34, Jan. 1991.

[13] JBoss Community, "Drools Expert." Accessed Dec, 2010.

[14] M. Jemtrud, P. Nguyen, B. Spencer, M. Brooks, S. Liu, N.Y. Liang, N.B. Xu, and N.L. Zhang, "Eucalyptus: Intelligent Infrastructure Enabled Participatory Design Studio," *Winter Simulation Conference*, Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 2047-2054.

[15] S. Liu, Y. Liang, B. Xu, L. Zhang, B. Spencer, and M. Brooks, "On Demand Network and Application Provisioning ThroughWeb Services," *Web Services, IEEE International Conference on*, Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 1120-1127.

[16] HSVO team members., "HSVO: Healthcare Services Virtual Organization." http://hsvo.ca. Accessed Dec. 2010.

[17] Cisco Systems, Inc., "Unified Communications - Main Page - Cisco Systems." Accessed Dec. 2010.

[18] Tandberg Co., "Video conferencing solutions and Telepresence systems - Tandberg Canada." Accessed Dec. 2010.

[19] Wikipedia, "Collaborative software." Accessed Dec. 2010.

[20] Charles Arthur, "Survival of the unfittest," *Guardian Unlimited*, Feb. 2006.