Modelling the sharing of resources across collaborative sessions
Spencer, Bruce; Liu, Sandy

Publisher's version / Version de l'éditeur:
https://doi.org/10.1109/APSCC.2008.255
Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE, pp. 813-818, 2009-02-10

Modelling the Sharing of Resources across Collaborative Sessions

Bruce Spencer and Sandy Liu
Institute for Information Technology, National Research Council Canada
46 Dineen Drive. Fredericton, New Brunswick, Canada, E3B 9W4
bruce.spencer@nrc.gc.ca, sandy.liu@nrc.gc.ca

Abstract

Service-oriented architectures can be used to provide multiple simultaneous sessions to users that wish to communicate over a variety of media. This gives rise to rich, highly effective communication sessions that can greatly enhance users' interaction. For example, a health services virtual organization seeks to use such tools for a variety of purposes: virtual patient simulation, anatomical visualization and virtually sharing cadaveric dissections. We propose SAVOIR, Service-oriented Architecture for Virtual Organization Resource and Infrastructure for this task, where tools and applications are resources and they can be accessed and controlled via Web Services. The purpose of this paper is to present a method for modeling sessions in SAVOIR by using Web Ontology Language (OWL). We express in OWL 1.1 constraints on when sessions can and cannot be run or cannot be run concurrently with the sessions now running. There are several types of violations: aggregate bandwidth may exceed capacity, network infrastructure may not be available, too many users may want to access a limited shared resource, etc. The session scheduler depends on the OWL 1.1 description logic reasoner to evaluate the session for violation of these constraints before the session is allowed to be scheduled.

1. Introduction

A growing trend for communications companies is to propose service-oriented architectures for configuring communication tools [2, 4, 7, 14]. Configuation information is dealt with at the services layer, which provides an alternative to the usually closed, proprietary communication protocols that prevent different systems from interacting. Through Service-oriented architecture, which have the advantages of being multi-platform and loosely coupled, a variety of communication resources and tools can be accessed, incorporated and provided simultaneously.

The Eucalyptus system [8, 11, 12, 15] provides industrial designers and architects with the tools to fundamentally change the process of design, by enhancing communication, application sharing, access to batch processing by supercomputers, network bandwidth file sharing, etc. Derived from Eucalyptus, SAVOIR (Service-oriented Architecture for a Virtual Organization's Infrastructure and Resources) is a generic tool for combining any communication resources that can be invoked through web services. Specifically given a WSDL file that describes how to access the tool, and given a widget-based user-interface to allow end users to manipulate the tool, SAVOIR can incorporate the tool and the interface to manipulate the tool throughout a virtual organization. In addition to providing access to the tools, SAVOIR manages the running sessions, which are composed of multiple simultaneous subsessions, each delivering a communication tool to specific users. The sessions are to be modeled internally so that SAVOIR is aware of what tools and resources are currently busy providing what servers to what users. SAVOIR then can determine which additional tools can be run simultaneously and which cannot. Typically a user requests that a session be invoked for communication among a set colleagues, and that a subsession for each of a set tools be started and managed by SAVOIR. Before and tools are started, an analysis must be done to determine that the tools can in fact be run simultaneously, and moreover that the session can be started without interfering with the other sessions that are currently running. This check involves several conditions. Do the participants have the necessary permissions? Is the required aggregate network bandwidth available? Are there any tools that can support only one connection to a user at a time, but more than one of these connections will be needed simultaneously to satisfy the new request?

There are several advantages to conduct this check using a reasoning tool instead of directly implementing a procedural check. There is a combinatorially large number of possible sessions, and there may be many sessions running simultaneously. A procedure that checks for constraint violations cannot effectively be programmed directly because there are too many possibilities to consider. Moreover, we already model the various tools and resources that can be combined in SAVOIR. In this paper we add constraints from description logic to our model of the tools, and show that the new role composition axioms in OWL1.1 [23] provide the expressiveness we need. When combining tools that were never intended to work together and other legacy systems, it is important to have a flexible modeling language to deal with the unintuitive and hidden conflicts that inevitably arise.

The paper is organized as follows: We present some related work in the area of communication technology based on web services. The next section presents some previous work on which SAVOIR technology is based. We then discuss the design of SAVOIR and how it can be applied in a Health Services Virtual Organization (HSVO) setting. The next section describes how SAVOIR sessions are modeled using OWL 1.1. Finally we present our conclusions and tasks left for future work.

2. Related Work

Previous work by a team at Avaya seeks to deliver voice and multimedia services through Web services, opening the usually proprietary systems to open standards [4]. For example the services provided by the SIP protocol, which is based on neither XML nor Web Services, can also be provided by WIP, as described by this team. SAVOIR differs in that it does not attempt to provide a standard protocol; instead SAVOIR can incorporate applications and allows them to use their own protocols.

Other work on unified communications and communication enabled applications is proposed by a consortium from Nortel, IBM and partners [2, 7]. This team proposed to set up a sandbox for partner companies to trial new services, testing for interactions and compatibility.

These efforts illustrate that there is much interest in delivering suites of communication products that interact with the existing telephone systems, and apply the mature telecommunications architecture to the problems that will be encountered. In this paper we focus on just the high level modeling of the service suites, which is a technique that can be applied to all these efforts, but is somewhat orthogonal to the work on the protocols themselves.

SAVOIR is built on our previous work Eucalyptus [11, 13, 12, 15]. It was specifically designed and built for architects and industrial designers. Architecture and industrial design are examples of advanced professions requiring collaboration of a diverse team around powerful visualization and modeling tools. Our team was responsible for the web services infrastructure that brought the tools together. The tools include lag-free and jitter-free UltraGrid uncompressed high definition videoconferencing [21], delivered over a high speed research network using User Controlled Lightpath Provisioning (UCLP) [20, 24]. It also included desktop applications like Maya from AutoDesk and OpenSceneGraph for sharing access to high-fidelity 3D models of a city street-scape, replicating ten city blocks of Montreal's Boulevard St. Laurent. The architectural team reported that working with these tools greatly improved their ability to collaboratively design, share insights, and work productively [8].

3. Design of SAVOIR

With the growing accessibility of the Internet and the sophistication of the middleware tools, many people are working with remote collaborators under the notion of Virtual Organizations (VOs) [19]. A VO usually consists of a group of geographically distributed members and resources. Using the state-of-the-art cyberinfrastructure services, members in a VO can work coherently as a whole to conduct scientific research, industrial design, or solve business problems. Typically each VO shares a pool of service-enabled resources and communication-enabled tools that may include software applications, hardware, data collections, computational power, storage, and even specialized applications for configuring optical private networks [18, 20]. SAVOIR is designed to be a generic service-oriented framework that can be used by different virtual organizations using different sets of resources.

Naturally, in this type of environment, there will be complex requirements for access control and authorization between the separate entities, with the possible inclusion of multiple unchangeable legacy systems.

In SAVOIR, the pool of shared resources are made available to participants in a VO in the form of Web Services. As shown in Figure 1, the SAVOIR dashboard acts as an integrated service client for accessing resources in a VO. The set of resources that appear on the dashboard is customizable. Each resource is represented as a widget that can be added or removed on demand. Thus the user can choose what widgets to appear in the dashboard. The core part of SAVOIR includes a set of management services that manages users, resources, sessions, and session workflows with the assistance of a set of utility services. A SAVOIR session may involve the usage of multiple resources provided by different organizations in the VO.

In this paper, we illustrate how SAVOIR models and Manages collaborative sessions in the context of a HSVO. Let's consider this conversation among health professionals:

Dr. S: The patient has evidence of a pulmonary embolism here in the X-ray and the MRI scan shows plaque in the carotid artery.
Dr. T: According to the Canadian Medical Associations Clinical Practice Guidelines, here, this is not uncommon for patients from his demographic.

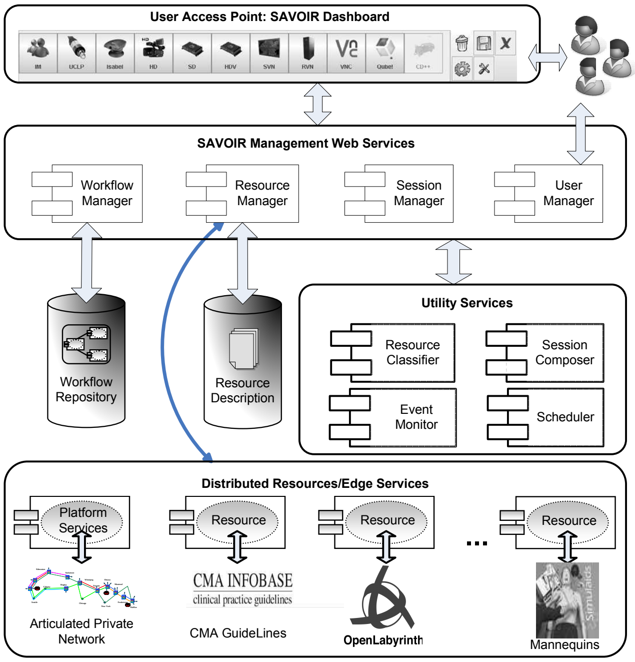

Figure 1. An Overview of SAVOIR

Dr. S: Lets look at the 3-dimensional fly-through of the normal tissue from the Digital Human for comparison.
Dr. T: OK. Now watch as I simulate the recommended procedure on the mannequin with the same condition as the patient.
Dr. S: Oh, the mannequins blood pressure is 180/120. While you were doing that, I also ran a virtual patient simulation and found a similar BP and also shallow, rapid breathing. Lets put this case in the database for further study.
Instructor: OK, Doctors, good work. Can anyone in the class tell me what they missed?

Dr. S and Dr. T are doctors in training and are discussing a case under the observation of the instructor and the rest of the class. They are using a number of tools that allow them to investigate the patient's symptoms, find the root causes, look up the recommended procedure, test that procedure on both a mannequin and a virtual patient simulator and to observe that the procedure's effect put the patient in a state of high blood pressure, which means that something may not have been done correctly.

While this conversation may be the state of the art in training physicians in the classroom settings, we seek to create a system in which the participants are in different locations. The instructor is in one city, the trainees S and T are in another city and the rest of the class are perhaps in a different location. Furthermore some of the tools that are being used are from the internet, such as the Visible Human [22] and the online searchable Canadian Medical Association Handbook on Clinical Practice Guidelines [3]. In order that all of the interactions in the conversation can be done, we need to create the connections between servers and clients shown in Figure 2. That is we need to incorporate access by

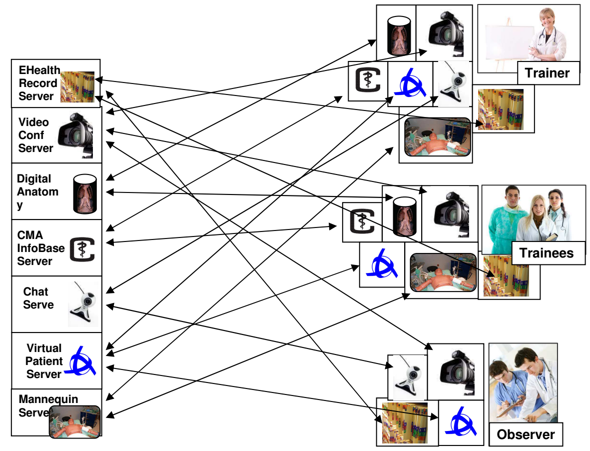

Figure 2. HSVO Session

each of the three teams to an EHealth record server, some videoconference equipment, a digital anatomy server, the CMA Infobase, a server for the virtual patient [6, 10] and a server for the physical mannequin simulated patient, such as the SimMan by Laerdal [9]. Notice that the mannequin can be controlled by the instructor while it is being manipulated by the trainees.

4. Modeling SAVOIR Sessions in OWL 1.1

A SAVOIR session is composed of a number of subsessions, each of which is responsible for one or several connections between users' client software with the servers. A SAVOIR session is composed of a number of subsessions, each of which is responsible for one or several of these connections. For instance, there is one session for the connection between one user and the CMA Infobase server. So there are two CMA Infobase sessions. On the other hand the connections with the Open Labyrinth Virtual Patient server may be a single session because all three teams may be sharing the same use case; Dr. S, one of the trainees is running the use case while the trainees and the observers are only observing. Likewise use of videoconference between teams typically group several teams into a single session.

Consider the following SAVOIR session, which will be discussed through the rest of this paper. It is a simplification of the session shown in Figure 2. Two users, Dr. S and Dr. T, each connecting to the CMA Infobase server are communicating with each other over a videoconference server based on equipment from Pleora [16]. The Pleora system is available only as point-to-point. We chose Pleora because it is atypical and thus challenged our modeling technology.

In the description that follows, words in *italics* are terms used in the description logic model. Of those, words starting in upper case will be classes and those starting with lower

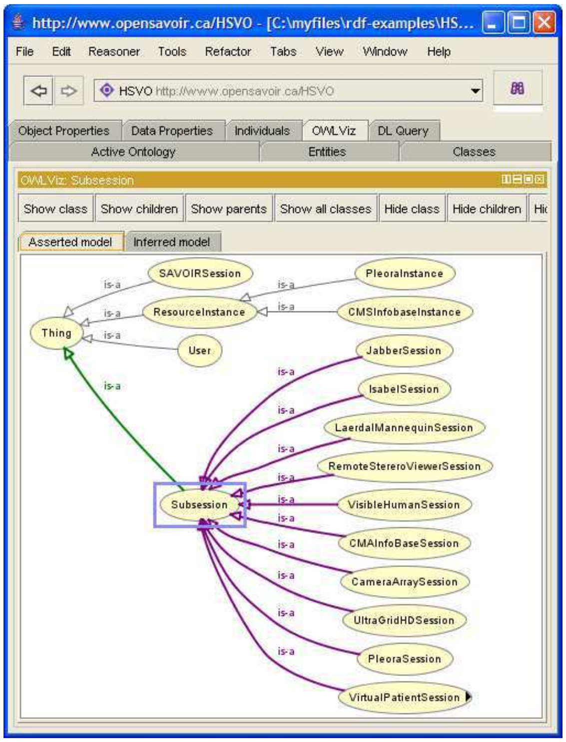

Figure 3. HSVO Class Hierarchy in Protege

case are instances, or individual members of those classes. For instance, the class *User* has two instances: *userS* for Dr. S and *userT* for Dr. T.

A *SAVOIRSession* is composed of a number of *Subsessions*, each of which has a number of running instances of some resource. The set of running instances is *ResourceInstance*. In our example, the instance of the *SAVOIRSession* is *consultationSession01*. This is composed of three *Subsessions*. The first *pleoraSubsession01* is connecting *userS* and *userT* over Pleora technology. The second *cmaInfobaseSessionForUserS* connects *userS* to the CMA Infobase Server, and the third *cmaInfobaseSessionForUserT* connects *userT*. The *pleoraSubsession01* has two running instances: *pleoraInstanceForUserS* and *pleoraInstanceForUserT*.

The class hierarchy is shown in Figure 3, as drawn by Protege 4.0 [17].

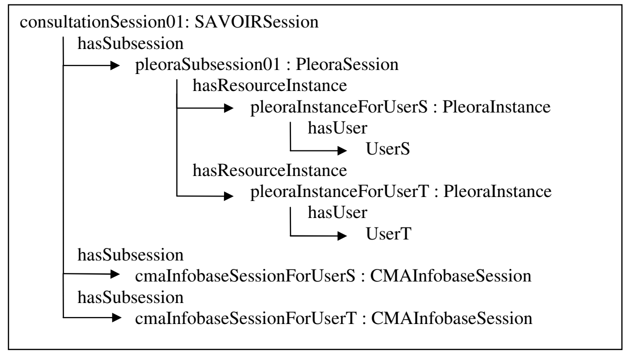

Figure 4. Savoir Session Instance with all of its parts

In addition to the class hierarchy, there is a classification based on the part hierarchy, a so-called partonomy, shown in Figure 4. In this view we can see that the *SAVOIRSession* is related to various subtypes of *Subsession* by the relation, or role, *hasSubsession*. The inverse relation of *hasSubsession* is *isSubsessionOf*. This figure shows the actual values of the instances within the partononmy that relate to this example. Thus there are in fact three instances of *Subsession* discussed already: *pleoraSubsession01, cmaInfobaseSessionForUserS*, and *cmaInfobaseSessionForUserT*. The figure also shows that the role *hasResourceInstance* relates *Subsession* to *ResourceInstance*. In this case *hasResourceInstance* relates a *PleoraSession* to a *PleoraInstance*, which are subclasses of *Session* and *ResourceInstance*, respectively. Note that the inverse role name for *hasResourceInstance* is *isResourceInstanceOf*. Also, *hasUser* and *isUserOf* have their obvious meanings.

We use the specific language OWL 1.1 [23] to represent the HSVO sessions. Features in OWL 1.1 that are not found in earlier versions have made the modeling easier. We take advantage of OWL's role composition feature. This is a rule-like feature that allows us to state easily that a certain property has an additional definition as the composition of two properties. This is often used, for instance, to state that the property like *owns* should be applied to all the parts of the thing that someone owns: if you own the thing then you own all of its parts, and even the parts of those parts.

$$a\ owns\ b \wedge b\ hasPart\ c \rightarrow a\ owns\ c.$$

In Protege, this is written succinctly as

$$owns \circ hasPart \rightarrow owns$$

where the $\circ$ means relational composition.

We also make use of the standard constraints available in OWL 1.0, such as existence requirements (at least one must exist) and numeric constraints (at least N or at most N can exist). This is explored more in the next section.

5. Managing Multiple Simultaneous Sessions

In general there are a number of constraints to check when a new session is requested by a user. What permissions do the users have to start and access the *ResourceInstance*'s? What are network requirements of these instances? What other physical constraints arise from the technology itself that prevent the new session from being realizable? All of these constraints can be checked with special purpose code that verifies the various conditions in a procedural way, and constructs the answer. The problem with the procedural approach is that it does not scale: one would need to write very general purpose procedures that can apply in all situations that might arise, and it is very easy to miss conditions. Since we have a model of the SAVOIR resources, our strategy is to also express constraints on this model, giving us a general purpose tool that can be read declaratively and is therefore easier to understand and verify.

For instance we want to be able to check that the two instances of the class *SAVOIRSession*, namely *consultationSesson01* and a new *consultationSession02*, can be managed simultaneously, given that *consultatoinSession01* is already running.

A problem with using the description logic OWL 1.0 for this task in the past was that the knowledge representation system was not very flexible. Properties of objects would need to be repeated if they also applied related objects. For the ownership example of the previous section, if one wanted to declare that a owns b and also owns a part of b called c, then one would have to repeat the declaration that a also owns c, for all such parts c. This gives rise to a need to manually repeat these values, which is neither a general nor sustainable solution and gives opportunities for making the set of instances inconsistent. The corresponding issue in our situation is that a *ResourceInstance*, say *r* that *isResourceInstanceof* some *Subsession*, say *s* should also be considered to be *isResourceInstanceOf* the overriding *SAVOIRSession*, say t that *subS isSubsessionOf*. The general rule is

$$r\ isResourceInstanceOf\ s \wedge s\ isSubsessionOf\ t \rightarrow r\ isResourceInstanceOft,$$

or more succinctly as the composition rule

$$isResourceInstanceOf \circ isSubsessionOf \rightarrow isResourceInstanceOf$$

By pushing the property *isResourceInstanceOf* from the *Subsession* instances to the *SAVOIRSession* instances, we can do checking for constraints at the *SAVOIRSession* level. Recall that, one of the consequences of Pleora's point-to-point technology is that a user cannot be connected to two different communicating partners at the same time, since that would be a three-way conversation. Now by using this standard cardinality constraint, that a *PleoraSession* must be part of at most one *SAVOIRSession*, we can express the condition:

$$PleoraSession \sqsubseteq isResourceInstanceOf \leq 1.SAVOIRSession$$

This constraints states that *PleoraSession* is a subset of the (anonymous) class of items which are *ResourceInstance*'s of **at most one** SAVOIRSession. In other words, if any Pleora session is in two different SAVOIR sessions, there is an inconsistency. In our example, *pleoraInstanceForUserT* is known to be a resource instance for *consultationSession01*, by virtue of the composition rule.

If the second SAVOIR session, *consultationSession02* were to be requested, the Session Manager would need to ensure that this new session would not interfere with any currently running session, in particular *consultationSession01*. Suppose *consultationSession02* also contains a *PleoraSession* in which *userT* and some other *userU* are to connect, The Session manager adds this proposed session to the knowledge base and checks for any inconsistency. On finding a violation of the cardinality constraint, the session manager will disallow the proposed new session.

The reasoning engine Pellet version 1.5 [5] implements these features of OWL 1.1, and was used in conjunction with Protege 4.0. The particular error message generated by Pellet is very easy to understand:

```
Consistent: No
Reason: Individual \#pleoraInstanceForUserT has more than
1 values for property \#isResourceInstanceOf violating the
cardinality restriction
```

SAVOIR employs a Jena interface to the check consistency of any session request.

6. Conclusion and Future Work

In this paper we have described a generic approach to modeling the requirements and capabilities of a set of legacy tools that are intended to be run simultaneously to connect a set of users. As far as we know this in one of the first attempts to apply Semantic Web technology to the emerging fields of Unified Communications and Communication-Enabled Applications. The main contribution of this paper is the validation step applied to any newly requested session against our OWL constraints. We exemplify these constraints with a full example showing too many users of a shared limited resource.

We base our approach to communication upon Service-oriented Architectures, because of their open standards, loose coupling and applicability on multiple platforms. This allows us to incorporate a highly diverse set of tools with very different resource requirements and potentially many unanticipated interactions. We model the running sessions as a session composed of subsessions each with a number or running instances of some application, thus generalizing the usual notion of a session from TCP or SIP. We use description logic as the modeling language, specifically OWL 1.1 so that resource instances from separate sessions can be checked for consistency, and we work through a detailed example, showing that the message for inconsistency from Pellet 1.5 is quite informative about the source of the inconsistency. This inconsistency indicates that the requested sessions will not be able to be realized, or in description logic terms, does not have a consistent model.

In future work we see that while the system as proposed does answer the question "Can my session scheduled now?", it does not answer the question "How soon can my session be scheduled?" i.e. incorporate the consistency check within a session reservation service. This adds a temporal dimension to the verification; this session must not violate any constraints with the set of sessions scheduled to be run concurrently with it, or at overlapping times. To do this properly will require us to use an interval temporal description logic [1].

Since we are talking about health care services, the natural question of priority arises. Suppose one session is being used for training, and meanwhile another session is requested that is a matter of clinical treatment, that is of higher priority. It would be useful to determine a minimal set of currently running sessions that can be shut down to accommodate that higher priority session.

References

[1] A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Ann. Math. Artif. Intell.*, 30(1-4):171–210, 2000.

[2] J. Bednarek. Bring the World of IT Applications & Communications Together with SOA. `http://soa.sys-con.com/read/523433.htm`, 2008.

[3] Canadian Medical Association. Canadian Medical Association Handbook on Clinical Practice Guidelines. `http://mdm.ca/CPGSNEW/CPGS/`, 2008.

[4] W. Chou, L. Li, and F. Liu. Web Services Methods for Communication over IP. In Zhang et al. [25], pages 372–379.

[5] K. Clark and B. Parsia. Pellet: The Open Source OWL DL Reasoner. `http://pellet.owldl.com/`, 2008.

[6] R. Ellaway. OpenLabyrinth. `http://sourceforge.net/projects/openlabyrinth`, 2008.

[7] IBM. IBM and Partners Utilizing SOA Strategy to Help Healthcare Providers Improve Efficiency and Patient Care. `http://www-03.ibm.com/press/us/en/pressrelease/23582.wss`, 2008.

[8] M. Jemtrud, P. Nguyen, B. Spencer, M. Brooks, S. Liu, Y. Liang, B. Xu, and L. Zhang. Eucalyptus: Intelligent Infrastructure Enabled Participatory Design Studio. In *WSC '06: Proceedings of the 37th conference on Winter simulation*, pages 2047–2054. Winter Simulation Conference, 2006.

[9] Laerdal. SimMan. `http://www.laerdal.com/simman`, 2008.

[10] Learning Technology Section, College of Medicine and Veterinary Medicine. Labyrinth. `http://mdm.ca/CPGSNEW/CPGS/`, 2008.

[11] S. Liu, Y. Liang, and M. Brooks. Eucalyptus: A Web Service-enabled e-Infrastructure. In B. Spencer, M.-A. Storey, and D. Stewart, editors, *Proceedings of CASCON 2007*, pages 1–11, 10 2007.

[12] S. Liu, Y. Liang, B. Xu, L. Zhang, B. Spencer, and M. Brooks. On demand network and application provisioning throughweb services. In *IEEE International Conference on Web Services (ICWS)*, pages 1120 – 1127, July 2007.

[13] S. Liu, B. Spencer, Y. Liang, B. Xu, L. Zhang, and M. Brooks. Towards an Agile Infrastructure to Provision Devices, Applications, and Networks: A Service-oriented Approach. In *31st Annual International Computer Software and Applications Conference. COMPSAC 2007*, pages 473 – 478, July 2007.

[14] Nortel. Nortel Agile Communication Environment Delivering Speed, Accuracy and Agility. `http://www.nortel.com/solutions/soa/collateral/nn123284.pdf`, 2008.

[15] ORION Discovery Award. The ORION Discovery Award 2007 Winner: Eucalyptus. `http://www.orion.on.ca/2007orionawards/pdf/Backgrounder\%202007\%20Winners.pdf`, 2007.

[16] Pleora Technologies. Pleora Technologies News Releases. `http://www.pleora.com/news/pr_070416_pleora_cims.php`, 2007.

[17] Stanford Medical Informatics. The Protege Ontology Editor and Knowledge Acquisition System. `http://protege.stanford.edu/`, 2008.

[18] B. St.Arnaud. CA*net4 research program update - UCLP roadmap: Web Services workflow for connecting research instruments and sensors to networks. `http://www.canarie.ca`, December 2004.

[19] The National Science Foundation Cyberinfrastructure Council. Cyberinfrastructure vision for 21st century discovery. `http://www.nsf.gov/pubs/2007/nsf0728/nsf0728.pdf`, March 2007.

[20] The UCLP Development Team. User Controlled Lightpaths. `http://www.uclp.ca`, 2006.

[21] the UltraGrid Project team. UltraGrid: A High Definition Collaboratory. `http://ultragrid.east.isi.edu/`.

[22] United States National Library of Medicine. The Visible Human Project. `http://www.nlm.nih.gov/research/visible/visible_human.html`, 2008.

[23] W3C OWL Working Group. OWL 1.1 Web Ontology Language. `http://www.webont.org/owl/1.1/`, 2008.

[24] H. Zhang, M. Savoie, S. Campbell, S. Figuerola, G. von Bochmann, and B. S. Arnaud. Service-oriented virtual private networks for grid applications. In Zhang et al. [25], pages 944–951.

[25] L.-J. Zhang, K. P. Birman, and J. Zhang, editors. *2007 IEEE International Conference on Web Services (ICWS 2007), July 9-13, 2007, Salt Lake City, Utah, USA*. IEEE Computer Society, 2007.