CHALLENGES OF BUILDING END-TO-END NETWORK TOPOLOGIES FOR MULTI-SERVICE NETWORKS

Anne-Marie Bosneag and David Cleary
Ericsson Ireland Research Centre, Ericsson LMI, Cornamaddy, Athlone, Ireland

Keywords: End-to-end network topology, multi-service networks, inventory solutions, service oriented architecture.

Abstract: Building an end-to-end view of the network in multi-service networks is a very important building block in creating a correct view of the capabilities of the network and is therefore crucial for service deployment, activation and management. At the same time, it is a very difficult task to achieve. Current inventory solutions are static in nature, which introduces problems of data consistency between the managed domain and the view at the management node. An end-to-end view of the network is also hard to achieve because no common data model is in use today, and inventory information between different domains is very hard, if not impossible, to obtain using the current solutions.

This paper discusses the challenges of building an end-to-end view of the network topology, discusses existing solutions, and proposes Stratus, a flexible unified SOA-based architecture that has been implemented and deployed in our lab. Stratus is based on the MTOSI recommendations for inventory retrieval, and comes with a mechanism for data mapping from domain managers for Public Ethernet and Core Wireline Access Networks to the MTOSI model. It also provides dynamic discovery of the different domains in the network, automatic end-to-end topology creation, and notification systems for automatic updates of the topology based on updates in the network. The experiments show that the proposed solution is technically feasible and compatible with existing inventory solutions.

1 INTRODUCTION

An accurate and timely model of the inventory of resources in a mobile telecommunication network provides support for network planning, and correct and effective operation and maintenance. Building an up-to-date, accurate representation of the network inventory in an OSS (Operation Support System) is a challenging problem in the current static inventory solutions. Moreover, solutions for different types of networks have been developed for a long time without relying on a common data model, and usually no inventory information reflects resources that link different domains, such as physical links between two domains.

A common solution in today's commercially deployed systems is to rely on external inventory systems, which get the inventory data from the different domains in the network, interpret it, and provide it to the operator in a common format. However, such a solution is static, being based on transferring the entire inventory data from domains (usually through FTP), followed by intensive processing and aggregation. The resulting inventory will therefore reflect the state of the network at the time of getting the inventory data from the network, and not at the time of presenting it. It also does not reflect data about resources between domains.

In this paper, we describe our practical experience with implementing an inventory service that is based on a *common data model* (TMF MTNM – TeleManagement Forum Multi-Technology Network Management), is able to provide an *end-to-end view* of the network topology, and *reacts in real time to changes* in the network.

The study outlines the different approaches to inventory data modeling that have been proposed to date, and discusses the shortcomings of existing in-

Bosneag A. and Cleary D. (2007).
CHALLENGES OF BUILDING END-TO-END NETWORK TOPOLOGIES FOR MULTI-SERVICE NETWORKS.
In *Proceedings of the Second International Conference on Wireless Information Networks and Systems*, pages 243-250
DOI: 10.5220/0002149902430250

ventory systems. It then introduces Stratus, a flexible architecture for providing an end-to-end view of the network. Stratus is based on a Service Oriented Architecture, in which the different inventory providers (i.e., Domain Managers) are Web Service implementations of the MTNM inventory data model. Dynamic aspects are incorporated into the architecture: the Domain Managers are dynamically discovered and accessed, while changes in the network are detected and clients get alerted through asynchronous notification systems. To provide an end-to-end view of the network, Stratus also allows the operator to define inter-domain links.

For the testing scenario, two Domain Managers were used: Ericsson's Multi-Service Networks Operations & Support System (MN-OSS), providing inventory information for the core wireline access in both next generation networks and circuit switched networks, and Ericsson's Public Ethernet Manager for Multi-service Access Nodes (PEM), offering dedicated management for broadband access networks. Translation of inventory data from the proprietary format used by these two domains to the MTOSI data model is provided, along with the ability to create inter-domain links (also added to the inventory in MTOSI format). The dynamic features mentioned in the previous paragraph have been implemented using the BEA products for Web Services (WebLogic Server and AquaLogic Service Registry). The deployment proves that Stratus is compatible with existing inventory solutions, to which it adds value by providing an accurate end-to-end topological view.

The paper is organized as follows: Section 2 describes the different approaches to data modeling that have been in use or proposed by different standardization bodies in telecom, while Section 3 brings forth issues related to existing architectures for inventory systems. Section 4 introduces the architecture used in Stratus, and explains how this architecture addresses the problems that exist in current inventory systems. Section 5 presents the scenario and the deployment used for testing the architecture proposed, and finally, Section 6 presents conclusions and future work directions.

2 CURRENT APPROACHES FOR INVENTORY DATA MODELING

2.1 Taxonomy of Approaches

The importance of inventory information prompted a lot of activity in the area of inventory systems. Typically, the inventory information in a telecom network is classified into three groups defining inventory functions; these are product, services and resources. Each of these functions have their own set of entities and relationships specific to the business logic, and interactions with other OSS functions. However, all inventory applications share a common set of abstractions. If we analyze the current approaches we can see a simple taxonomy emerging. At the heart of this taxonomy stay information representation and access.

- **Deep Modeling** or detailed modeling of the underling network with typed finely-grained access to the information model: This category is interesting as it places most of the intelligence in the structure of the information model. As the model is strongly typed, access to it normally results in a specific application programming interface with strong data typing. The key advantage of this approach is that the model is explicitly interoperable. However, strongly typed systems are hard to agree on and standardize, and are normally very specific to the network infrastructure (such as ATM/IP). Such an approach can be seen in DMTF CIM (Tosic and Dordevic-Kajan, 1999), TMF MTNM (TeleManagementForum, b), TMF MTOSI (TeleManagementForum, c).

- **Shallow Modeling** or abstract modeling of network resources with generic access mechanisms: This category strives to remove the concept of strong typing. The information model introduces the concept of Managed Object as a generic container for inventory information. This approach stems from the seminal work on network management put forward by ITU-T (the Telecommunication Standardization Sector of the International Telecommunications Union) (3GPP, 2000). This approach allows for syntactical interoperability, but requires the semantics to be captured in the application logic. The current third generation 3GPP IRP (TS32.695, 2006) follows this approach.

- **Meta Modeling** is a mixture of the first two models, normally modeling the underlying capability of the various networks. In this approach a trade-off is reached, between the difficulty of agreeing on a strongly typed model and capturing the semantics in the application logic, by a model layer that captures some meta-modeling making the data easier to process. A good example is the JSR 142 (TeleManagementForum, a).

In the next subsection, we take a look at a current proposal regarding inventory systems for wireless networking.

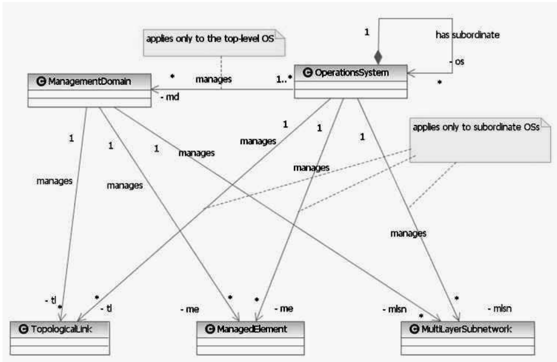

Figure 1: MTOSI model for Managed Domain.

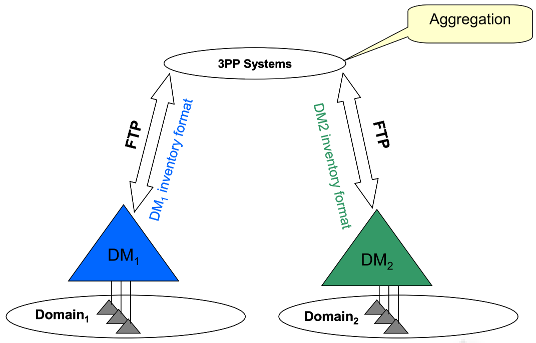

Figure 2: Current inventory in systems.

2.2 Inventory for Wireless Networks

A very important point to be made when talking about wireless network inventories is that the inventory is an **end-2-end** concept from the operator's perspective. The goal is to create an all encompassing view of network resources and services. This becomes even more important in the context of multi-service networks, where there is a heterogeneous mixture of access technologies that facilitate wireless edge connectivity. Typically, the termination points are represented by a home/office environment. More and more often, this termination is no longer a termination in the classic sense, but a gateway or bridge to a wireless network. If we have an objective to allow global services to roam into these new networks, it is imperative that we capture the global, end-to-end view in our inventory data.

MTOSI (Multi-Technology Operations Systems Interface) is the basis for a full-featured, carrier-grade, scalable solution that provides a unified, open interface between Operations Support Systems (OSS) for the purpose of network and service management. MTOSI is a single standard (TMF 854) whose goal is to decrease the complexity and avoid un-necessary repetition of work when integrating a multitude of OSS systems together. In the context of the taxonomy presented above, it is an XML binding of the MTMN modeling standard that focuses on the areas of inventory and alarms:

- Inventory retrieval: through this functionality, systems can retrieve inventory data from OSs
- Inventory notifications: notifications of changes in the inventory data are reported to interested parties
- Alarm reporting: this functionality allows an OS to send requests to a set of interested OSs
- Active alarm retrieval: through this functionality, an OS is able to retrieve the set of active alarms known to another OS.

Figure 1 shows the top level of the MTOSI model, where the concepts such as Management Domain, Managed Element, and Topological Link exist. We highlight these concepts here, as they were used in our implementation, as explained in the next sections.

3 THE END-TO-END NETWORK TOPOLOGY ISSUE

Apart from the problem of aligning all Domain Managers to provide inventory data using a unified data model, as discussed in the previous section, an architecture for an end-to-end topology system presents additional challenges. We focus here on the following problems:

1. lack of an end-to-end topological view of the network
2. inconsistencies between the live state of the network and the perceived state at the management node
3. static vs. dynamic approach
4. level of aggregation of inventory data
5. compatibility with existing inventory providers
6. platform independence

Current inventory systems rely on getting information about parts of the network and then aggregating this information to create an end-to-end view of the network. With the capabilities offered by current inventory systems, which are mostly FTP-based (Figure 2), creating this end-to-end view requires a lot of processing and aggregation at high levels in the architecture, even at the level of an external inventory system (such as Cramer, Telcordia, MetaSolv). Moreover, inter-domain inventory data is usually not available and very hard, if not impossible, to infer.

The goal of our work is to provide an extension to current inventory systems, that is compatible with current inventory providers in the network, and that is able to offer an end-to-end view of the topology. Compatibility with existing inventory providers (or domain managers) in the network is obviously essential, as the cost of introducing the new solution is much lower if current deployments are upgraded than if a completely new system must be put in place. Current inventory data will still be used, and nonexistent data related to resources between domains will be added to the current view, the result being an end-to-end view of the network topology.

Platform independence is also a requirement for a system that spans domains. A good solution should work equally well on different platforms. Since Web Services provide a platform-independent way of leveraging a specific functionality, they are a good choice to match the goal. In Section 4 we present an architecture based on Web Services.

Another major issue with current inventory systems is that they are usually static. This leads to data inconsistencies, increased overhead and delay in reflecting changes in the network, and also presents a problem when aggregating data over several domains if the domains themselves have changed. Since static approaches capture the state of the network at the time of the inventory request, and no automatic updates notify the inventory system of changes taking place in the meantime, inconsistencies between the state of the network and the view of the inventory system are introduced (Brennan et al., 2006). At the same time, changes affecting the availability of domain managers, or their properties, can have an impact on the final end-to-end topological view, and without automatic notifications regarding the changes, an incorrect network view will be created at the management node. Therefore, one of the requirements for our system is to incorporate automatic notification services that allow the update of the topological view as changes in the network take place.

4 STRATUS: A UNIFIED ARCHITECTURE FOR AN END-TO-END SOLUTION

In this section we describe a flexible, SOA-based architecure for inventory solutions, that addresses the issues presented in the previous section. This architecture provides:

- a unified model for presenting inventory data to outside systems – all inventory data is presented in an XML-based format that conforms to the MTOSI specification
- end-to-end topology creation – in addition to presenting all inventory data from individual domains, an end-to-end view of the network is created by facilitating creation/ destruction of inter-domain links. The new links are presented according to the same MTOSI specification.
- automatic discovery of network domains and of management nodes – the network domains are discovered through the UDDI, and clients can therefore dynamically bind to them to retrieve inventory data for that particular domain. Moreover, all changes regarding additions/ deletions/ changes in properties of the domains are automatically sent to clients from the UDDI.
- automatic updates of inventory data within the domains – asynchronous notifications are sent via a JMS queue from each of the domain managers to the Link Manager, whenever changes in the inventory data within the domain took place.

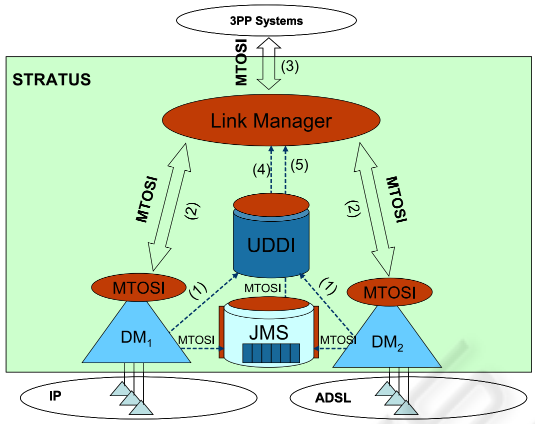

Figure 3: Stratus architecture.

The architecture is presented in Figure 3. Two domain managers DM_1 and DM_2 are represented in the figure: these managers are providers of inventory data for the two domains, for example IP and ADSL as in the figure. The two domain managers are Web Services, registered with the UDDI ((1) in Figure 3). They offer inventory information for the domain in a common data model (the MTOSI format).

The Link Manager is the module responsible for the creation of the end-to-end network topology view. This module will discover the available network domains by querying the UDDI for available Domain Managers. Once these are discovered, each domain can be accessed and inventory data retrieved (2). To create the end-to-end view, the Link Manager needs information about links between the domains, in addition to the inventory data from each domain. This information can be created/ modified/ deleted through a functionality offered by the Link Manager, and further presented to external systems, such as Cramer, using the same data model (MTOSI) (3).

The UDDI also sends automatic notifications to the Link Manager when changes pertaining to the registered Domain Managers are detected (4). The changes detected are: new Domain Manager registered with the UDDI, existing Domain Manager deleted from the UDDI, type of Domain Manager changed in the UDDI. The first two types of changes refer to changes in the number and/or the identity of the Domain Managers, while the last type of change is very useful in upgrade situations, when for example a newer version of the Domain Manager is available for providing inventory information.

The architecture also includes a JMS queue, used for notifications of inventory data changes. Each Domain Manager is responsible for monitoring its own domain and whenever changes are taking place, a message is sent to the JMS queue, from where they are asynchronously delivered to the Link Manager (5).

The next paragraphs expand on the architectural concepts outlined at the beginning of this section, by presenting how the architectural choices enable the implementation of these concepts, and also how these concepts address the problems outlined in Section 3.

The ***unified data model*** follows the MTOSI inventory data model. The focus is on information associated with physical links, respectively on the resource model associated with Equipment, Physical Termination Point and Topological Link, as presented in Section 2.

Compatibility with existing inventory providers is ensured by the fact that the current Domain Managers are based on the existing inventory providers, which are extended to include a *translator* from the internal, domain-specific inventory format to the MTOSI format, and a *Web Service interface*. We are currently using domain-specific translators for the conversion, and plan to go towards more automatic methods in the future. Each Domain Manager becomes a Web Service that registers with a central UDDI, to enable dynamic discovery and binding. The choice of the Web Service technology also ensures platform independence. All messages sent through the system between the different elements (Figure 3) are SOAP messages, and the messages carrying inventory or notification information also conform with the MTOSI model.

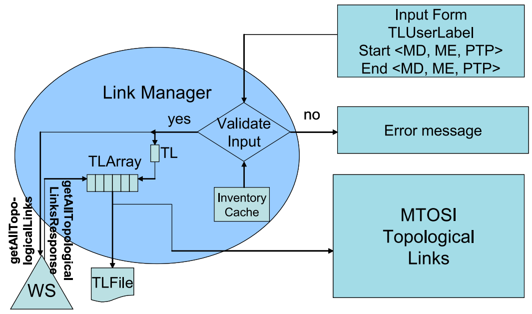

Figure 4: End-to-end topological view enabled by creation of links between domains.

An ***end-to-end topological view*** is created at the level of the Link Manager. This is achieved by allowing users to input information about physical links between managed domains, as well as to modify and delete this type of information. Figure 4 presents the method for creating a new inter-domain link. The start and end points for the topological link are provided by the user in the form $<$Managed Domain, Managed Element, Physical Termination Point$>$. Once these entries are validated against the inventory data provided by the relevant Domain Managers, a new object of type TopologicalLink is created, and added to the inventory data. MTOSI-compliant messages getAllTopologicalLinks/ getAllTopologicalLinksResponse are used for retrieving information about existing topological links.

By providing this functionality, Stratus covers a gap that exists in current inventory systems, in which information about links between domains is not stored anywhere. The architecture also allows aggregation of inventory data closer to the managed domains, as opposed to aggregation at the level of third party systems, such as an external inventory system.

Dynamic discovery of the managed domains is achieved through the UDDI. The UDDI performs two functions in the context of the Stratus architecture:

- *discovery of existing Domain Managers* – at the time of creation/ activation, each Domain Manager registers with the UDDI as provider of inventory data for a particular type of domain. This registration is done manually in Stratus, an automated method being forseen for future work.

  The Domain Managers register in the UDDI as inventory providers, following a taxonomy that includes type (e.g., Domain Manager for the Public Ethernet Access Network) and version (e.g., v 2.3.). Therefore, when a Domain Manager registers with the UDDI, it will also be classified according to this taxonomy. Discovery of domains is performed by querying the UDDI for registered services of type inventory provider (with certain

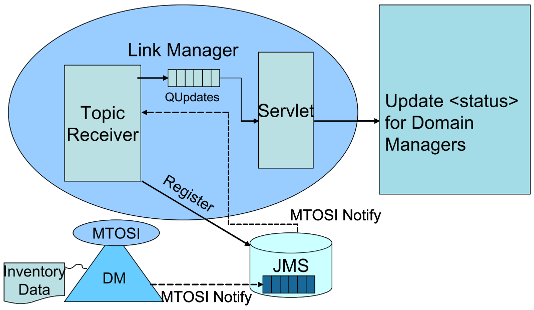

Figure 5: Asynchronous notification system for changes in the inventory data.

characteristics, according to the described taxonomy).

- *notifications of changes with the registered Domain Managers* – whenever changes are registered with the UDDI, such as new Domain Manager registers, or existing Domain Manager is deleted, or the characteristics for an existing Domain Manager change (e.g., it is upgraded to a higher version), the UDDI sends notifications to the client Link Manager (who registered for receiving notifications for events related to services of type inventory provider). For the implementation, we used the BEA AquaLogic Service Registry 2.1, which is fully compliant with UDDI v 3, which provides support for user-defined taxonomies and notification services regarding changes in the registered Web Services.

Such a dynamic and flexible architecture is compliant with the MTOSI recommendations, and brings advantages over the currently used static architectures in terms of effort and response time to changes in the system.

Another dynamic feature offered by the proposed architecture is an ***asynchronous notification system*** that detects and notifies clients of ***changes in the inventory data within domains***. Figure 5 presents the implementation of this notification system. Each Domain Manager monitors its own domain and whenever changes in the inventory data are detected, a MTOSINotify message is sent to the JMS queue in the system, under the topic InventoryDataChanged. Clients such as the Link Manager register a message listener with the JMS queue, for notifications on the same topic. Whenever a message arrives at the destination, the JMS provider delivers the message by calling the listener's onMessage method, which acts on the contents of the message (it queues updates in Link Manager and makes them available for displaying).

This notification system addresses data inconsistency issues that exist in current systems between the

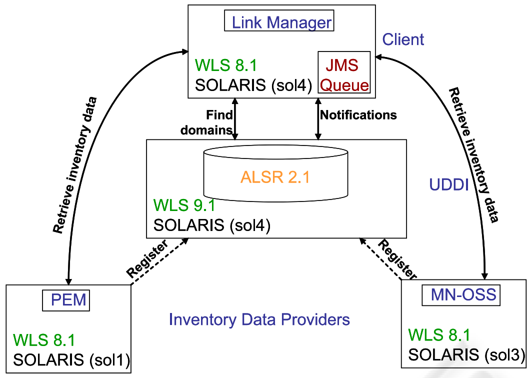

Figure 6: Deployment of Stratus prototype.

live state of the network and the perceived state at the management node, in that it ensures that the most current data is always seen by the management node.

5 EVALUATION OF THE APPROACH

The Stratus prototype was implemented and deployed in our lab, as depicted in Figure 6. Real data from two Domain Managers (Ericsson's Multi-Service Networks Operations & Support System – MN-OSS, and Ericsson's Public Ethernet Manager for Multi-service Access Nodes – PEM) was used for testing purposes. The MN-OSS product manages the core wireline access in both next generation networks and circuit switched networks, while PEM offers dedicated management for broadband access networks. Both of them offer information about physical links between the nodes in the managed domain, in their own proprietary format. To fully prove backward compatibility with existing Domain Managers, the prototype was implemented on the Solaris platform normally used to run MN-OSS and PEM.

For the development of Web Services, UDDI, and JMS queue, we used the BEA products. The WebLogic Server WLS 8.1 (highest fully developed version at the time of prototype implementation) was used for developing the Web Services, while for the UDDI we used the AquaLogic Service Registry ALSR 2.1, which is fully compliant with UDDI v3. For the JMS queue, we used the one included in the distribution of WLS 8.1.

The scenario considered is depicted in Figure 7. The network includes Core Wireline and Public Ethernet domains, managed respectively by MN-OSS and PEM, as well as wireless networks such as 3G, managed by OSS-RC. The inventory system gathers information from all these Domain Managers, and

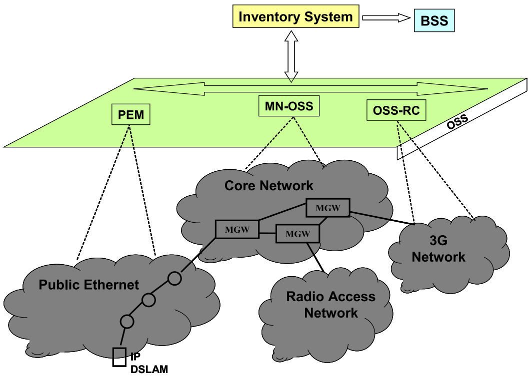

Figure 7: Deployment scenario.

feeds relevant data (the end-to-end view, capabilities, etc.) into BSS (Business Support System).

The cases studied in our tests are:

1. discovery of and dynamic binding to existing domain managers – the Link Manager enquires the UDDI for a list of available Domain Managers that offer inventory data. Once these are returned, they can be accessed directly by the Link Manager. The inventory information from each of the domains is displayed in MTOSI-compliant format (Figure 8).
2. creation/deletion of inter-domain physical links – the input data (<Domain, Network Element, Port> for start and end points in the case of creation, and linkUserLabel in the case of deletion) is verified against the existing inventory data. If the information is correct, the link is created and displayed in MTOSI format, respectively deleted from the end-to-end inventory.
3. changes in inventory data within a domain – both the MN-OSS and the PEM managers monitored their own inventory file. Whenever a change was detected, a notification was sent to the Link Manager through the JMS Queue. The BEA visualization tool enabled tracking of these messages to prove correctness and timeliness of the mechanism.
4. new domain added / existing domain deleted / type of domain changed – all these changes were simulated by registering, un-registering, or changing the version for the PEM domain. All changes were correctly propagated to the Link Manager and further displayed onto the screen.

The tests prove the compatibility with existing Domain Managers (MN-OSS, PEM), the correctness of our solution, and the technical feasibility of the implementation on the existing platform.

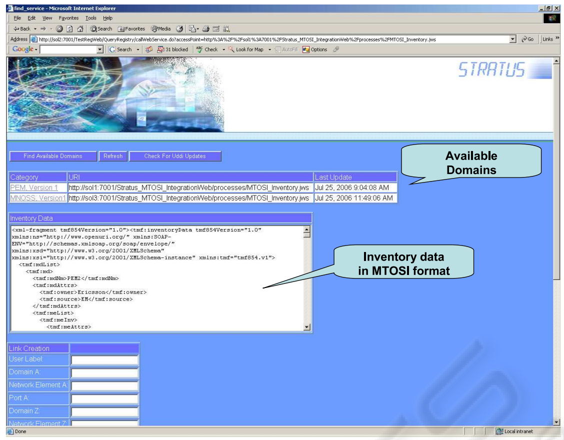

Figure 8: Available Domain Managers and Inventory Data presented according to the MTOSI model.

6 CONCLUSIONS AND FUTURE WORK

The main contributions of this paper are presenting an overview of current inventory solutions, with the different approaches and problems of existing inventory systems, as well as presenting our practical implementation of a flexible inventory system that provides an accurate end-to-end topological view of the network following the MTOSI standard format. The system monitors changes in the network and reacts in a timely fashion. Our study shows the feasibility of the approach and compatibility with current inventory solutions.

For future work, we plan to extend the use of SOA to include sematic data representation, to facilitate automatic methods for mapping the domain-specific inventory data to the standard MTOSI model, as well as to use semantic Web Services for discovery of Domain Managers as opposed to the central UDDI based discovery.

ACKNOWLEDGEMENTS

The authors would like to thank to their Ericsson colleagues for their comments and contributions, including Tony Thisted Jakobsen, Jan Groenendijk, Alan Babington, James Sweeney, Epifanio Salamanca Cuadrado, Javier Baliosian, Boris Danev, and Francoise Sailhan. We would also like to thank to BEA for their support throughout the duration of the project. The research of Anne-Marie Bosneag is funded by the European Community's Marie Curie Fellowships for the Transfer of Knowledge action.

REFERENCES

3GPP (2000). 3GPP TS 23.002 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Network Architecture. In *http://www.itu.int/rec/T-REC-M.3010/en*. ITU-T.

Brennan, R., O'Gorman, G., Doherty, C., Hurley, N., and Olsson, S. (2006). Autonomic Replication of Management Data. In *In Proc. of the IEEE/IFIP Network Operations And Management Symposium (NOMS 06)*.

TeleManagementForum. OSS/J, JSR 142: OSS Inventory API, Release 2.0. TMF.

TeleManagementForum. TMF608 Version 3.3 Information Agreement for MTOSI Release 1.1. In *http://www.tmforum.org/browse.aspx?catID=4133 &linkID=32482*. TMF.

TeleManagementForum. TMF854 Systems Interface: XML Solution Set for MTOSI Release 1.1. In *http://www.tmforum.org/browse.aspx?catID=4133 &linkID=32480*. TMF.

Tosic, V. and Dordevic-Kajan, S. (1999). The common information model (CIM) standard - an analysis of features and open issues. In *Proc. of the 4th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, page 677.

TS32.695 (2006). 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Inventory Management (IM) Network Resource Model (NRM) Integration Reference Point (IRP): Bulk CM eXtensible Markup Language (XML) file format definition (Release 6). In *http://www.3gpp.org/ftp/Specs/html-info/32695.htm*. 3GPP.