Universidade do Minho
Escola de Engenharia

Semana da Escola de Engenharia
October 24 - 27, 2011

MiNSC: Network Services Configuration Made Easier

Miguel Lopes, Antonio Costa and Bruno Dias
Department of Informatics
E-mail: {miguellopes, costa, bruno.dias}@di.uminho.pt

ABSTRACT

Network management is facing new challenges with the definition of future Internet. Among those challenges the management of network's heterogeneity represents a highly complex and unsolved problem. Most common proposals address the management of network's heterogeneity, following the RFC 3139 guidelines, translating from mid-level independent policies to the network's services and devices heterogeneous management interfaces and data models. However the translation mechanisms implemented, apart from being highly complex, perform inefficient mapping operations and depend on the administrator manual intervention of maintain and validate the mappings. To overcome the use of management translation mechanisms, for high-level network service management, the MiNSC framework is proposed, providing a mid-level management abstraction based on standard information models. This paper beside presenting and motivating MiNSC's framework, evaluates its contributions for the simplification of large scale, heterogeneous network service management. A practical example of MiNSC's service DNS deployment capability is also presented in this paper.

INTRODUCTION

With the future Internet architecture and functionalities definition new perspectives and challenges rise for network management. From the ability to apply new business models, multi-service management, to the highly heterogeneous requirements of user and devices, the future Internet management must extend over several dimensions. At the same time, human interference at the operational and administrative management must be kept at the minimum level, releasing the administrator for the strategic planning and performance tasks. When addressing the management of network's heterogeneity, of legacy service and devices, most solutions rely on intermediary management translation mechanisms, following the model presented in Request for Comments (RFC) 3139. However, the realization of syntactic or semantic translations from high-level management representations to the network services and devices heterogeneous management interfaces is very complex (or even unfeasible) for large scale environments. Implementing such translations creates management solutions with low flexibility, resulting in a high dependence on the administrator intervention to create and maintain the conversions up-to-date. Furthermore the translation mechanisms are commonly built in accordance to the high-level management applications needs, implementing a proprietary solution which leads to a lack of management application interoperability.

In order to overcome the use of management translation mechanisms, to support the network's heterogeneity, a Mid-level Network Services Configuration (MiNSC) framework is proposed. It provides an intermediary (mid-level) service to higher-level network management applications that, by implementing standard-based service management information models, implemented over standard interfaces, provides an infrastructure for universal, secure and efficient service configuration management. The MiNSC's service management architecture is based in two abstraction layers: i) a higher layer where the service behavior is represented following the *Service Management* information model; ii) a lower layer where the heterogeneous service implementations are abstracted by standard-based *Node Management* information models. The management independence provided by the lowest management abstraction layer, associated with the service behavior representation, enables the automation of service management procedures, like the automatic generation of service nodes configurations (also referred as service deployment). To evaluation this capability, for the Domain Name Service (DNS) management, this paper includes a detailed description of the automatic and independent DNS deployment process based on the DNS service meta-configurations. To complement the

presentation of MiNSC's motivations, a succinct evaluation demonstrates MiNSC's most relevant contributions, for large scale heterogeneous service management, compared to other network management frameworks. The remaining paper is organized into six sections: in the next section we present the most relevant network management research projects; MiNSC's framework principles and motivations are described in section three; in section four, the management framework's are shortly evaluated considering the large scale heterogeneous management capabilities; the automatic and independent DNS management is addressed in section five; the last section of this paper is for the conclusions remarks.

RELATED WORK

Autonomic network management [1], [2] deals with the Future Internet heterogeneous environment and inherent management complexity. Implementing the autonomics' control loop, the self-management capability is granted for the network to automatically adjust its behavior in agreement with operational, administrative and strategic business goals (in general, represented as policies [3]). Throughout the management process network resources state information is gathered, business policies are verified and management actions are expedited when needed. This drives the managed resources to the desired state and closes the management control loop. If management actions are needed, resource-specific configurations are generated accordingly. This requires an intermediary entity that support's each network resource management interface and data model. The FOCALE [4] project is based on the implementation of autonomic control loop. There ontologies are used to augment the managed resources heterogeneous data models by mapping their elements into a common vocabulary enabling the development of semantic equivalences between them. This unified way of dealing with network diversity, includes a mid-level configuration conversion tool to apply and retrieve resource-specific management data. It's up to the Model-Based Translation Layer (MBTL) the task of converting higher-level representations into lower-level resources-specific configuration language and commands. However FOCALE does not defines the MBTL practical details, implementation architecture or particular strategy to tackle this complex mapping task. Following the same line of thought [5], [6] implement proxy-based management translations, sharing the same limitations as the previous works.

The FOCALE's Autonomic Element (AE) was also studied in [7] and one important limitation was identified: the MBTL represents a potential management bottleneck for a centralized entity that must *speak* all network resources management interfaces and data models. To solve this and other limitations the authors propose a modified version of the FOCALE's AE architecture. Using an extended version of the AE architecture (based on mobile agents) a simplified MBTL operation is obtained, enforcing vendor-neutral management data to the network resources, overcoming the centralized and inefficient management translations. However such translations are still present performed in a distributed fashion and realized at the network resource level. This represents an obvious performance enhancement, however the realization of management translations at that network resource level constrains the already constrained resources availability.

The configuration management framework proposed by Cfengine [8] uses a declarative language to describe the low-level management policies expressed as *promises* (containing the management domain's intensions, highly dependent on the network resource's implementation details). The *promises* defined are sent to management agents that extract their configuration management operations and ensure their automatic deployment. Some of the management activities realized include package installation verifications, configuration files generation, file protection and consistency checking. Cfengine also uses knowledge representations to relate the *promises* intensions, making easier to reason about and promote the understanding of the management objectives. Cfengine is not considered an integrated network management framework (supporting the management of network's heterogeneity), its low-level *promises* definition language provides a low management abstraction not intended for integrated network management. Another popular configuration management tool is the NETCONF [9], which, in reality, is just a network protocol. It was created with the intension of overcoming the most popular network management protocol limitations: the Simple Network Management Protocol (SNMP) [10]. Due to its alleged simplicity, SNMP presents several limitations especially for the configuration management in large networks, which

seems to be the reason why it is mainly used for the monitoring tasks [11], [12]. NETCONF contains the transport and management facilities to securely manage configurations defined in datastores. To guarantee the configuration management interoperability a data model definition language, named YANG [13], was created providing a standard way to describe NETCONF's configuration management data models. Arguably, NETCONF solves some of the SNMP's alleged configuration management limitations but it does not deal with network's heterogeneity or promotes automation, it's just a configuration transport protocol. Other network services and devices configuration management applications are available like Smartfrog, LCFGng [14] and Bcfg2 [15] however they do not promote the management automation, requiring relevant and specialized human intervention. Besides, they use (or depend) centralized configuration data repositories, maintain a *server-centric* management view with limited high-level functionalities.

MID-LEVEL NETWORK SERVICES CONFIGURATION

Initially presented in [16], the MiNSC framework was formerly designated as Automated and Distributed Network Service Configuration (SCM). It is part of a more general architecture for automated and distributed network service management, where it works in cooperation with an Automated and Distributed Network Service Monitoring (SMON) framework. This later framework dynamically computes the managed services' Quality of Service (QoS) and triggers re-configuration operations to be automatically performed by MiNSC. This paper refers to the MiNSC's proposal and more details about the project can be found in [17]. Most of the integrated network management solutions that aim to support the management of the network's heterogeneity, implement the model presented in [18] and depicted in Figure 1. In the proposed model the Configuration *Management Data* represent the high-level management policies embedding the business goals that the management domain must comply to. Then, combined with the *Network Topological Information* (containing the management domain elements details) and the *Network Status Information* (containing monitoring information), the *Network-Wide Configuration Data* is generated. This data represents mid-level, device-independent policies from which the *Device Local Configurations* are derived. The Network-*Wide Configuration Data maintain* a network-oriented view of the management in opposition to the business-oriented view of the *Configuration Management Data* policies. This makes the mid-level policies much easier to implement and manage since they do not contain device-specific data and do not provide a high-level abstracted view of the management. In order to enforce the Network-*Wide Configuration Data* a *Configuration Data Translator* is used, translating from the mid-level, independent policies to the device/implementation specific configurations (*Device Local Configurations*), following each element management interface and data model. However, the implementation of these intermediary management translation mechanisms has some major limitations:

- When performing a syntactic translation the semantics of management data, on both source and destination is not considered, which might result in management data inconsistencies or collisions [19], [20];
- Due to the lack of semantic content of contemporary management data models, the realization of semantic translation is a complex task, depending on the administrator's manual intervention for the mapping process [19], [20]. This is a very complex task that might not be feasible for highly heterogeneous management environments;
- Management translation mechanisms are highly

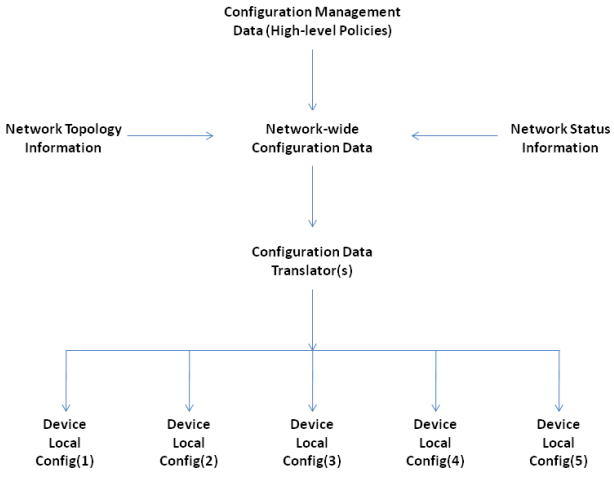

Figure 1 – RFC 3139 heterogeneous network management

dependent on the managed element's features, frequent updates (installing new features or redefining existing ones) en- force evolution of the translation process requiring an administrative effort (mostly manual) in maintaining the translation mechanisms up-to-date;

- Management translation mechanisms do not promote interoperability of the management applications since they do not rely on standardized interfaces. Besides, they are dependent on the management models used;
- Different translation mechanisms must be created for different devices or service implementations, even though they perform similar tasks, there's a low re-use of specifications;
- Translation mechanisms are prone to errors, i.e., eventual changes in the managed elements may require manual adaptation of the translation mechanism, which might result in introduction of errors; and,
- Depending on the location of the translation element, additional limitations might be introduced, for example, if it's placed in a central entity, it will include resilience (single point of failure), scalability (not adequate for large scale domains) and performance (creating a potential management bottleneck for management operations) limitations and if placed on the management element, it will decrease the resources availability and synchronization among all elements will be much more difficult to attain.

Since the network services tend to be well described on international standards, independent service management information models can be derived from those descriptions. This means that all major services implementation can be managed through those models, creating a service management abstraction layer. MiNSC implements such management abstraction layer further divided into two hierarchical sub-layers. The lower abstraction layer unifies the management of heterogeneous network service implementation while the higher abstraction layer uses the independence and management unification provided by the lower layer to automate the service management tasks through the definition of the service behavior. With the implementation of both management abstraction layers, MiNSC overcomes the need for intermediary management translation mechanisms, becoming more suited for large scale heterogeneous network service management. MiNSC's new perspective for the heterogeneous network service management is depicted in Figure 2. As stated before, the Configuration *Management Data represents* the high-level (business oriented) policies while the *Network-Wide Configuration Data* represents the mid-level, independent service management meta-configurations, defining the service behavior. Then it's up to the MiNSC framework the task of automatically deriving and deploying each service node configuration based on the service behavior defined. Both abstraction layers are depicted in Figure 2. The second (and higher) management abstraction layer provides a mid-level, service-oriented management approach based on the service management information models. To deploy the service, at the lower layer, several information is required namely the service nodes management information model, their topological distribution and performance (provided through the monitoring systems). The nodes information model is used to decompose the service-oriented meta-configurations into node-oriented configurations. The nodes topological information is used to find the service nodes available (including their physical location) while their performance information is important to determine the nodes incompliance to the service behavior defined. The first (and lower) management abstraction layer eliminates the service nodes heterogeneity through the implementation of standard-based node management information models.

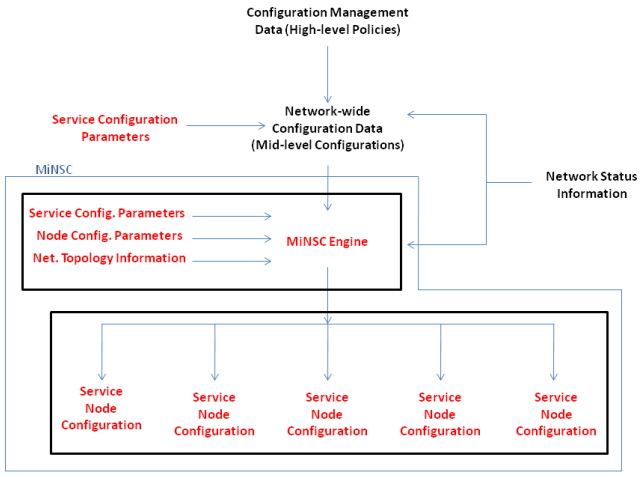

Figure 2 – MiNSC's integrated network Service Management

Semana da Escola de Engenharia
October 24 - 27, 2011

Architectiral Considerations

MiNSC proposes a distributed architecture for automatic and independent network services configuration management. High scalability and resilience is achieved through the distributed implementation of both management abstraction layers, organized as depicted in Figure 3. Following a bottom-up approach, the lowest abstraction layer (referred as Network Service Node Management) is composed by the Active Service Nodes (ASNs) and Candidate Service Nodes (CSNs) [16]. The ASN nodes perform the productive operations of the network service to be managed at a server or device level, while the CSNs are not actively performing a service but might (are prepared to) be in the future. At this level the nodes implement a management agent supporting a MIB for the service node's configurations management (based on the service node management information model). This eliminates the management heterogeneity and the need to implement management translation mechanisms. The ASN nodes configurations are automatically calculated by the layer above. The Active Configuration Servers (ACSs) and Candidate Configuration Servers (CCSs) comprehend the highest abstraction layer, maintaining the service management meta-configurations.

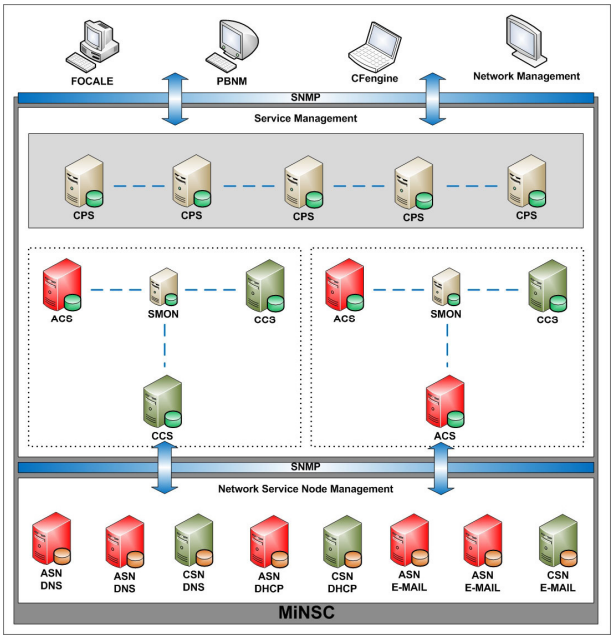

Figure 3 – MiNSC's network service management architecture

The ACS nodes actively manage the lower layer nodes (ASN and CSN), automatically calculating their configurations and distribution, based on the service behavior defined, while CCS nodes improve the system resilience and scalability maintaining up-to-date replications of ACS nodes configurations. This enables the realization of a quick transfer of management tasks (in case of ACS failures) or performs load-balancing operations, to improve the management framework's scalability, dividing the management tasks between ACS servers. The management servers also implement a management agent supporting a MIB providing an interoperable interface for high-level management applications. Since the ACS server classification can change along the time, the CPS servers are used to maintain an updated reference of the services being management by the ACS servers available. The CPS server is also used to assist the ACS server's configuration replication and load-balancing procedures, maintaining control information defined by the domain's administrator.

Configuration Management Protocol

To effectively manage the network service node's configurations, a configuration management protocols must be used. However, to be used in MiNSC's framework the configuration management protocol must be in agreement with the following requirements:

- Implement a standard network management interface to promote the management interoperability;
- Provide an efficient, reliable and secure configuration management operation;
- Facilitate the integration of MiNSC with contemporary network management applications.

Considering the previous requirement the two most relevant alternatives are SNMP and NETCONF. The usage of SNMP for configuration management tasks has been widely discussed, however the recall of its monitoring limitations into the configuration management context leads to some misinterpretations. When applied to MiNSC's distributed architecture, the SNMP's scalability limitations are mitigated, being the configuration management process distributed by a group of ACS servers, besides, the implementation of the SNMPv3 over TCP provides a secure and reliable operation. Another advantage of using TCP is the

capability to send in a single transaction, between manager and agent, an unlimited number of GET/SET operations (regardless of the number of managed objects). Traditionally, SNMP requires a large number of transactions, when the number of managed objects increases, with direct repercussions to the protocols scalability. Beside from an empirical point of view, even though a large number of configuration management objects exist, not all objects are managed all the time. Nevertheless SNMP provides a very limited set of configuration management operations, but includes a well defined network management interface (MIB) and well defined data models definition language hose expressiveness is considerably limited. NETCONF was recently created to solve the SNMP's alleged configuration management limitations. With configurations encoded in XML, NETCONF provides a secure and reliable operation with an advanced set of configuration management functionalities. NETCONF's data model definition language (YANG) was also recently standardized providing a rich set of modeling functionalities for definition of complex management models.

From MiNSC's perspective both configuration management protocols are supported and compliant with the identified requirements. SNMP is a more mature protocol, it possesses a data model definition language with limited expressiveness, but enables an easier integration with contemporary network management applications (mostly based in SNMP). NETCONF provides an advanced set of configuration management operations and an improved data model definition language. Given the functionalities provided by both protocols the authors decided to follow a traditional management approach, supported by well known and deployed management interface (MIB) and use SNMP to perform the basic configuration management operations. Beside facilitating the integration of MiNSC with contemporary management solutions, SNMP takes advantage of the existing implementation knowledge to aim a wider acceptance. However the authors are also aware about the evolution of network management models and increased requirements for suited configuration management operations. The evolution towards the use o NETCONF will be a natural step and this evolution can only be realized because of MiNSC's independence regarding the configuration management protocol used.

Standard-based Management Information Models

The implementation of standard-based service management information models has important implications inherent to the standardization process. These implications are referred in [21] and include the following:

- The standardization process takes normally a long time to be completed, this means that the standards may not be available when they are required, promoting the proliferation of proprietary management solutions;
- Standard network management information models include, sometimes, low-quality proposals due to the following reasons: i) the high commercial pressure to create management models in time; ii) the standardization process does not attracts the technological experts to the work groups due to the commercial motivation; iii) reaching an agreement among all vendor's requirements creates models either too generic, that are complex to implement (and can incur into interoperability problems) or too low-level loosing the integration perspective of the models.

From the previous reasons the usage of standard-based information models is questionable, requiring a more expedite standardization process. This issue is explored in [21] where the authors propose a multi-tier, interactive modeling process that aims to improve the development of network management standards.

LARGE SCALE HETEROGENEOUS NETWORK MANAGEMENT FRAMEWORKS EVALUATION

In this section we evaluate the framework's capabilities to address the management of large scale heterogeneous environments. To perform the evaluation a group of criteria were compiled and their compliance evaluated for MiNSC, Cfengine [8], FOCALE's [4] and WBEM [21]. The evaluation results are depicted in Table 1. To classify the criteria compliance, a "++" indicates a complete agreement with the criteria, a "+" indicates a partial agreement and the "-" indicates the criteria incompliance.

Heterogeneity

Networks keep increasing their size not only in the number of services and devices but also in the number

Semana da Escola de Engenharia
October 24 - 27, 2011

Table 1 – Management framework's evaluation

Criteria	*MiNSC*	*Cfengine*	*FOCALE*	*WBEM*
Heterogeneity	+ +	-	+	+
Resilience	+ +	-	*	-
Scalability	+ +	-	+	-
Interoperability	+ +	-	*	+
Translation	-	-	+ +	+

of functionalities and vendor diversity. This highly heterogeneous environment must be efficiently managed through the implementation of high-level management applications that focus on the integration of management tasks, while deploying management procedures throughout the network's heterogeneity. Cfengine fails in providing support for high-level network management since it enables only the definition of low-level management policies to work in a stand-alone mode. On the other hand FOCALE and WBEM support the management of network's heterogeneity, translating from high-level independent representations into heterogeneous management interfaces and data models. WBEM creates an integrated network management framework using a group of standard network management information models, called Common Information Models (CIM), to create a uniform data representation to be shared among heterogeneous managed elements. Then it's up to the Providers the task of mapping the CIM-based management representations into each device specific management interface and data model, implementing, most commonly, a syntactical translation.

FOCALE also creates an integrated network management framework based on the MBTL operation. The MBTL translates the DEN-ng independent management representations to device-specific management interface and data model. In order to create a semantic integration between the heterogeneous management data models, ontologies are applied to enhance the management elements represented, mapping them into a common vocabulary. Then ontologies are also used to semantically translate the DEN-ng representations into device-specific configurations. MiNSC employs a simpler alternative to support the heterogeneous network service management. Instead of defining management information models supporting all vendor-specific management functionalities, it focus on defining minimalist management information models based on the service standard definitions, ignoring all non-standard functionalities. These models enable the management of heterogeneous network service implementations that associated with a standard network management interface overcomes the need for management translation mechanisms. The most obvious consequence is all the non-standard functionalities that become un-managed since they are not contemplated in the service's standards.

Resilience

The network management activity provides a vital task in keeping the managed resources state under control, so it's essential to improve their availability. In this sense it's important to evaluate the management framework's capability to operate in the presence of management server's failures. Cfengine is a low-resilient framework, not implementing any type of protection to failures. However due to its management by delegation, in case of management framework failures, the last promises received (by the end host's agents) are continuously enforced. In the same way WBEM also does not implements any type of management resilience insurance, using traditional centralized client-server management architecture. Considerations about FOCALE's resilience can not be easily made since it's mainly composed by high-level descriptions. However, in case of failures external to the AE, the management policies can continue to be enforced but not updated. When failures happen internally to the AE reactive measures can be taken, using the AE's flexibility to increase its resilience dimension. However if the MBTL is implemented in a central entity, the AE resilience is reduced through the creation of a single point of failure. MiNSCs management resilience is achieved through the realization of indirect service configuration replication procedures. During the ACS operation the service management meta-configurations are periodically replicated to CCS servers. In case of ACS server failure a CCS server can quickly continue the management tasks increasing the management framework resilience. The monitoring system is used to inform about the ACS failure, starting the service execution transfer.

Scalability

To the management systems is required the capability to manage large scale environments. Here Cfengine

and WBEM proposals fail since they are depend on a centralized entity which makes them prone to scalability problems. Like the previous criteria FOCALE's scalability is also complex to evaluate, however, since it's based on the autonomous enforcement of policies at the AE level, negotiated through an orchestration plane (not directly enforced through a central entity), and represents a scalable management operation. However if its MBTL is implemented in a central entity, some scalability (and performance) limitation are introduced to the AE. In MiNSC, the provisioning of management scalability is achieved through the realization of indirect service configuration replication procedures, periodically replicating the ACS servers' configurations to CCS servers. When the performance of the ACS server decreases, do to scalability problems (detected by the monitoring system), a load-balancing procedure is executed, dividing the management tasks between ACS and CCS servers.

Interoperability

While supporting the management of network's heterogeneity, the high-level management applications interoperability must also be ensured to aim a wide usage. In this sense, CFengine fails in guaranteeing the management interoperability since it uses a proprietary language to define the management promises. Considerations regarding FOCALE's interoperability (at the AE and MBTL level) can not be done since it lacks implementation details. MiNSC and WBEM address the management interoperability using standard protocols to enforce the configuration management tasks. However, since WBEM's transport protocol and data model definition language were not specially developed for network management, it might incur in some interoperability limitation due to XML's excessive flexibility.

Conclusion

The most important conclusion taken from this study are summarized by:

- Most network management framework aiming to build integrated management solutions commonly use management translation mechanism. This introduces important limitations at the levels of scalability, resilience and interoperability;
- MiNSC overcomes the need to use management translation, based on the implementation of standard information models and standard interfaces. Besides, its distributed architecture provides a scalable and resilient operation.

AUTOMATIC AND INDEPENDENT DNS MANAGEMENT

A MiNSC-based prototype was created for independent DNS management. To effectively implement both management abstraction levels both DNS *Service* and *Node* management information models were created. This section describes the use o MiNSC for automatic setup of a DNS domain regardless of the underlying nodes heterogeneity.

DNS Management Information Models

The representation, in a class diagram, of the DNS management information models can be found in [17]. To represent the service behavior the DNS Service information model is composed by three classes:

1. **Domain**, that represent the DNS service authority domain;
2. **Element**, that contains information about other network elements' present on the network requiring name to address translation;
3. **Operation**, that describes the DNS general functionality like the service resilience level, the volatility of DNS database, service persistence, etc;

To abstract the heterogeneous DNS implementations a standard-based DNS Node management information model was defined. The model was built around the DNS Zone definition and includes the following classes:

1. **Zone**, to aggregate the DNS domain information of other classes;
2. **Directives**, to include the DNS standard directives;
3. **Behavior**, to represent the DNS node behavior;
4. **Records**, to aggregate the information of the DNS standard resource records, **A**, **MX**, **NS**, **SOA**, etc.

DNS Deployment

With the support of both management abstraction levels the prototype is able to perform automatic DNS deployment procedures. To demonstrate such capability a group of automatic deployment procedures were performed over heterogeneous DNS implementations: *Linux Bind* (A), *MS Windows Bind* (B)

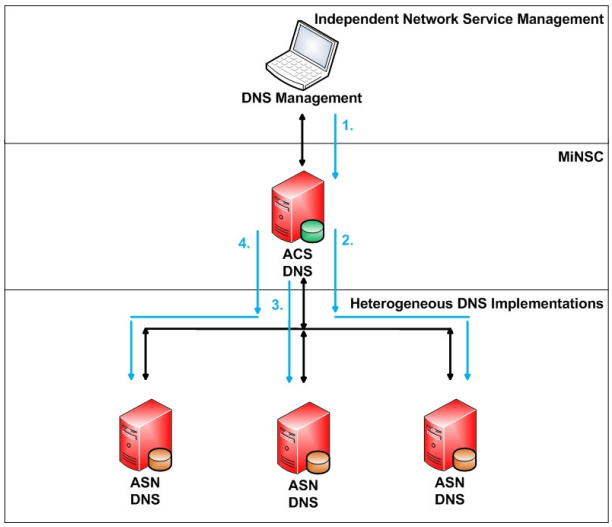

Figure 4 – Automatic and Independent DNS deployment

and *MSWindows Posadis* (C). In accordance to the DNS *Service* management information model, the following meta-configurations were used:

- Domain(parent:**di.uminho.pt**, authority:**scm**);
- Operation(resilience:**low**, recursion:**no**, notification:**no**, caching:**no**, volatibility:**low**, persistence:**low**, validity: **low**, duration: **low**).

This creates a DNS domain called **scm.di.uminho.pt** without recursion, notification or caching. Besides the **low** resilience means all three nodes are defined as ASN (with one **master** and two **slaves**). The deployment results are shown in Table 2 and the procedure, depicted in Figure 4, include:

1. Definition of the DNS behavior pretended;
2. Automatic (and independent) enforcement of the first DNS node's configuration (master);
3. Enforcement of the second DNS node configurations (first slave);
4. Enforcement of the third DNS node configurations (second slave).

Conclusion

The most important conclusion taken from this experiment are summarized by the following items:

- The deployment procedure is performed almost independently of the DNS implementations heterogeneity, which demonstrates the effectiveness of MiNSC's lower management abstraction layer;
- The effectiveness of the second management abstraction layer is demonstrated by automatically deriving each node configuration based on the defined meta-configurations (including the server's dependencies);
- The service deployment procedure was performed automatically (and safely) in a few seconds, which commonly takes dozens of minutes and intensive manual work.

Table 2 – Automatic and independent DNS deployment

Deployment Pattern	*Duration(seconds)*
A-A-A	4.035
A-A-B	3.919
A-B-B	3.469
A-B-C	3.558
A-A-C	3.908
A-C-C	3.471
B-B-B	3.017
B-B-C	3.005
B-C-C	3.340
C-C-C	3.154

CONCLUSION

Until now, high-level network management applications had the responsibility of unifying the management of heterogeneous software products and hardware devices. They maintain high-level management representations and define the intermediary translation mechanisms required to effectively perform their enforcement into each network element management interface and data model. The implementation of translation mechanisms is the most popular way to create integrated management solution that aims to support the heterogeneity of network's elements, following RFC 3139 guidelines. However, such translations introduce an identifiable group of limitations, namely: a high dependency on the administrator's manual intervention to create, maintain and validate the mappings; complexity to create integration models in highly heterogeneous environments; functional limitations when considering the management of large scale heterogeneous environments (incurring into scalability, resilience, interoperability and efficiency limitations).

To overcome the use of management translation mechanisms, MiNSC proposes a distributed architecture supported by the implementation of standard-based service management information models (for the management independence) and standard network

management interface (to improve the interoperability) that by elimitating and unifying the management heterogeneity does not incurs into the previous limitations. Besides, through the over-provisioning of network nodes (at both layers), scalability, resilience and interoperability is improved for the network service management.

REFERENCES

[1] J. W.-K. H. John Strassner, Sung-Su Kim, *The Design of an Autonomic Communication Element to Manage Future Internet Services*. Springer Berlin / Heidelberg, 2009, pp. 122–132.

[2] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. O. Foghlu, W. Donnelly, and J. Strassner, "Towards autonomic manage- ment of communications networks," *Communications Magazine, IEEE*, vol. 45, no. 10, pp. 112–121, 2007.

[3] J. Strassner, *Policy-Based Network Management.* Morgan Kaufmann, 2004.

[4] J. C. Strassner, N. Agoulmine, and E. Lehtihet, "Focale - a novel autonomic networking architecture," *ITSSA Journal 3(1)*, pp. 64–79, 2007.

[5] B. S. Karthik, M. Jaiswal, V. Menon, V. Kannan, S. Venkobarao, M. Pande, A. Talukder, and D. Das, "Seamless network management in presence of heterogeneous management protocols for next generation networks," *Information Technology, 2006. ICIT '06. 9th International Conference on*, pp. 68–71, 2006.

[6] S. S. Chavan and R. Madanagopal, "Generic snmp proxy agent framework for management of heterogeneous network elements," *Communi- cation Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, pp. 1–6, 2009.

[7] C. Hong, Z. Wenan, and L. Lu, "An approach of agent-based architec ture for autonomic network management," *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*, 2009.

[8] M. Burgess, *Cfengine 3 Concept Guide*, 2008.

[9] R. Enns, "Netconf configuration protocol," *RFC 4741*, 2006.

[10] D. Harrington, R. Presuhn, Wijnen, and B., "An architecture for describing simple network management protocol (snmp) management frameworks," *RFC3411*, 2002.

[11] C. Mi-Jung, C. Hyoun-Mi, J. W. Hong, and J. Hong-Taek, "Xml-based configuration management for ip network devices," *Communications Magazine, IEEE*, vol. 42, no. 7, pp. 84–91, 2004.

[12] G. Pavlou, P. Flegkas, S. Gouveris, and A. Liotta, "On management tech nologies and the potential of web services," *Communications Magazine, IEEE*, vol. 42, no. 7, pp. 58–66, 2004.

[13] E. Bjorklund, "Yang - a data modeling language for the network configuration protocol (netconf)," *RFC6020*, 2010.

[14] P. Anderson and A. Scobie, "Lcfg: The next generation," 2002.

[15] R. B. Narayan Desai and J. Hagedorn, "System management method- ologies with bcfg2." *login: Magazine*, vol. 31, pp. 11–18, 2006.

[16] M. Lopes, A. Costa, and B. Dias, "Automated network services configu- ration management," *Integrated Network Management-Workshops, 2009. IM '09. IFIP/IEEE International Symposium on*, 2009.

[17] B. Dias, M. Lopes, and A. Costa. (2011) Mid-level network services configuration (minsc). [Online]. Available: http://www.facebook.com/profile.php?id=100002078211438

[18] L. Sanchez, Megisto, K. McCloghrie, and J. Saperia, "Requirements for configuration management of ip-based networks," *RFC3139 (Informa- tional)*, 2001.

[19] J. E. L. D. Vergara, V. A. Villagr, and J. Berrocal, "Semantic management: advantages of using an ontology-based management information meta-model," in *Proceedings of the HP Openview University Association Ninth Plenary Workshop (HP-OVUA'2002), distributed videoconference*, 2002.

[20] J. Lpez de Vergara, A. Guerrero, V. Villagra, and J. Berrocal, "Ontology- based network management: Study cases and lessons learned," *Journal of Network and Systems Management*, vol. 17, pp. 234–254, 2009-09-01. [Online]. Available: http://dx.doi.org/10.1007/s10922-009-9129-1

[21] J.-P. Martin-Flatin, D. Srivastava, and A. Westerinen, "Iterative multi-tier management information modeling," *Communications Magazine, IEEE DOI - 10.1109/MCOM.2003.1252804*, vol. 41, no. 12, pp. 92–99, 2003.

AUTHORS' BIOGRAPHIES

MIGUEL LOPES is a PhD student at the University of Minho in Portugal. He completed its bachelor degree in Electronics Engineering at the University of Minho in 2004, worked at INESC Porto and PT Inovação before enrolling in the MAP-TELE doctoral program. His research interests include Computer Communication and Management. His e-mail address is: miguellopes@di.uminho.pt.

ANTÓNIO D. COSTA is Assistant Professor in the Department of Informatics at University of Minho in Portugal. He completed his PhD thesis in Informatics/Computer Communications at University of Minho in 2006. During the last years he has been working in several projects on Computer Communications, unicast/multicast routing and Quality of Service for IP networks. His e-mail address is: costa@di.uminho.pt and his Web-page can be found at http://marco.uminho.pt/~costa.

BRUNO A. F. DIAS is Assistant Professor in the Department of Informatics at University of Minho in Portugal. He completed his PhD thesis in Informatics/Computer Communications at University of Minho in 2005. During the last years he has been working in several projects on Computer Communications, Network& Services Management and Multimedia Technologies. His e-mail address is: bruno.dias@di.uminho.pt.