An Autonomic Architecture to Manage Ubiquitous Computing Networks and Applications

John Strassner, Sven van der Meer, Brendan Jennings, Miguel Ponce de Leon
Telecommunications Systems & Software Group, Waterford Institute of Technology, Waterford, Ireland
{jstrassner, vdmeer, bjennings, miguelpdl}@tssg.org

***Abstract*—The current Internet, while successful in many aspects, has a set of associated architectural and business problems that threaten its stability and inhibit new advances, such as seamlessly supporting Ubiquitous Computing applications. This paper proposes a set of autonomic mechanisms that can manage the Future Internet. The context-aware governance features of this approach are particularly suitable for also enabling Ubiquitous Computing applications.**

***Keywords*—Autonomic Systems; Context Awareness; Future Internet; Ubiquitous Computing.**

I. INTRODUCTION AND MOTIVATION

The current success of the Internet architecture has spurred advances in business, social, and technical communications. However, that simplicity is also the source of many of its inherent limitations [1]-[3]. Two of the most important are its architectural limitations and its inability to relate business needs to network services and resources offered. While the former problem is being worked on across the world [4]-[9], the latter problem tends to be ignored, though some theoretical work on using requirements engineering to align business services with technology [23] has been undertaken. Even more importantly, important areas such as management and security are usually listed as important problems; however, the vast majority of these studies do not offer any detailed analysis of the nature of these problems or any concrete ideas on how to solve them.

There are many reasons for this. Technologists tend to work on scientific problems, and forget that the Future Internet will be just as much a social phenomenon as it will be a technological undertaking [24][25]. Hence, economics, sociology, and other important disciplines should also be considered when solutions are created. Second, most of these studies do not have a business problem in mind. For example, the Internet was originally developed independent of any commercial considerations. However, it is becoming increasingly hard for traditional network and service providers to make a profit. While advertising-based and other newer models do enable some revenue to be generated, it is not clear how sustainable such models are in the future. Thus, unless the future network architecture takes competition and economic incentives into account, it is doomed to failure. Any new architecture has to be based on a sound economic model, not just a good technical one.

However, a third problem is that current network management and operational data does not contain business or system information. This means that network management applications must instead infer system and service problems. Since each vendor uses different programming languages and models with different semantics, it is very difficult for network management applications to integrate data from devices from multiple vendors.

Now consider the world of Ubiquitous Computing (UbiComp). In referring to the third wave of computing, one in which many computers serve a single person or group of people, Mark Weiser said "In such a world, we must dwell with computers, not just interact with them….Interacting with something keeps it distant and foreign…Dwelling with computers means that they have our place, we have ours, and we co-exist comfortably." [10] This paper epitomizes many of the fundamental goals of UbiComp systems, but especially the goal that the person now has to think less about his or her tasks and the environment because unnecessary work has already been done or summarized by the computer(s). This requires a "seamless connection" of things in the world with computing devices and mechanisms.

The vast majority of current UbiComp applications are not directly considered by many of the current studies on the Future Internet, as evidenced by the lack of focus on techniques for enabling machines to effectively cooperate with each other, let alone understand things in the world (including human wishes). A machine-understandable representation of business and network concepts, along with an ability to negotiate services and functionality to serve competing interests of users and applications in a domain, is required. Otherwise, machines will remain isolated from each other, and the vision of UbiComp applications will not be realized.

Given the vast diversity in programming models and languages for representing operational and management data, a common knowledge representation is needed that enables the communication of not just static data, such as performance and configuration information, but also semantic information that describes the business objectives that the network services are intended to satisfy. This enables the machine-based translation of these objectives into different vendor-specific languages; it also facilitates the use of machine-based learning and reasoning approaches that can be used to seamlessly connect the world of people with the world of machines.

978-1-4244-4216-4/09/$25.00 © 2009 IEEE

Context-aware services (i.e., services whose functionality changes in accordance with changes in context) exacerbate this problem. This is because the underlying semantics must be able to be context-driven, as the same data may have different meanings for different contexts. In general, each application brings a new concept for managing resources and services, and hence redefines similar concepts. For example, a configuration management application and a billing application can each define the concept of a "user" in different ways, having different attributes and datatypes; this complicates the exchange of management data and, more importantly, understanding the significance of those data.

This paper builds on previous work [11] and proposes extensions to the FOCALE autonomic architecture [12] for managing context-aware services, such as those required by UbiComp applications. We use UbiComp services as a model for services expected in Future Internet scenarios. The organization of the rest of this paper is as follows. Sections 2 and 3 briefly summarize the salient features of our previous work. Section 4 describes how Future Internet applications can be managed by managing the needs of UbiComp applications, a typical Future application. Section 5 summarizes the paper.

II. A Brief Introduction to FOCALE

FOCALE stands for **F**oundation – **O**bserve – **C**ompare – **A**ct – **L**earn – r**E**ason, which describes its novel control loop. FOCALE is a model-driven architecture that can dynamically generate code to (re)configure managed elements using a context-aware policy model [14]. The FOCALE context-aware policy model is ideally suited for UbiComp applications, as it enables FOCALE to use context to select policies to orchestrate behavior. A simplified block diagram of FOCALE is shown in Figure 1.

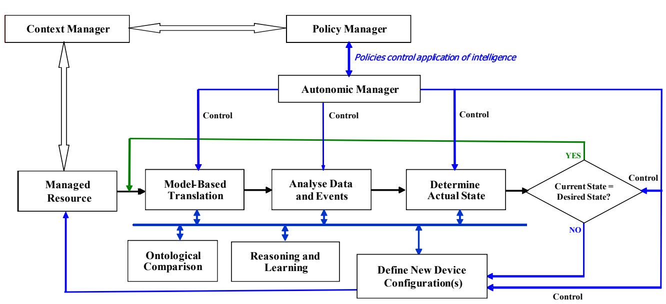

Figure 1. Simplified FOCALE Autonomic Architecture

Sensor data is retrieved from the managed resource (e.g., a router) and fed to a model-based translation process, which translates vendor- and device-specific data into a normalized form in XML using the DEN-ng information model and ontologies as reference data [17]. This is then analyzed to determine the current state of the managed entity. The current state is compared to the desired state from the appropriate Finite State Machines (FSMs). Nodes in a FOCALE FSM represent a configuration state; each state has an associated set of one or more configuration actions that define the configuration of an entity. Edges represent state transitions, and connote permission to change the configuration of a managed resource. Static behavior is thus "programmed" into FOCALE by designing a set of FSMs; dynamic behavior is defined by altering one or more FSMs. Context-aware policy management [16] governs the autonomic control loop. This enables context to select the set of policies that are applicable; policies are used to then define the functionality allowed. As context changes, policies change, and system functionality is adjusted accordingly. This is shown in Figure 2.

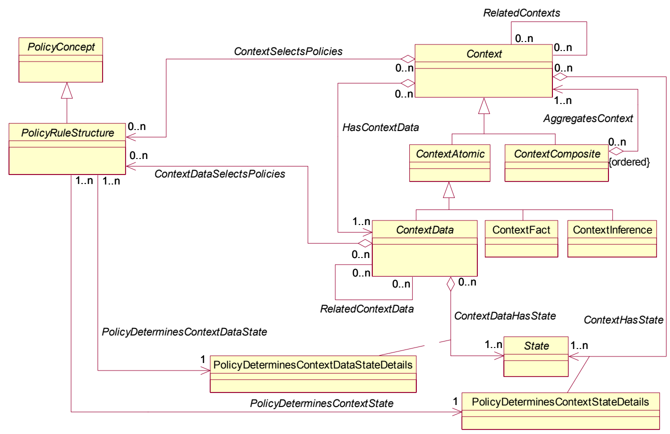

Figure 2. Context-Aware Policy Model

Context is defined as an aggregation of different aspects that collectively define an overall context. Each aspect of context, such as time and location, is modeled by a ContextData instance, whereas the Context instance represents the final assembled context with all of its different aspects. Both ContextData as well as Context can affect the set of policy rules that are currently used to govern behavior; this enables granular decisions that are dependent on one or more aspects of Context to change policy rules as well as a higher level change in the entire context to change policy rules. These two changes can also be linked to the changing of state in one or more FOCALE FSMs; this is provided in a two step process. The first step defines the set of policy rules that are used to determine how context affects a State, and the second defines the set of actions for either maintaining that State or transitioning to a new State. These are defined by the two association classes PolicyDeterminesContextDataStateDetails and PolicyDeterminesContextStateDetails, and by the two associations ContextDataHasState and ContextHasState.

FOCALE defines a normalized network management lingua franca by mapping vendor-specific data and commands to a vendor-neutral form based on a novel combination of information and data models augmented by ontologies [15]. Information and data models represent facts; these are augmented with semantics to enable machine-based learning and reasoning. It then uses a *model-based translation function* to interact with vendor-specific languages and programming models.

FOCALE supports a *dynamically updateable* knowledge base – one that can reflect new knowledge at runtime as new knowledge is discovered. Our approach supports this requirement by using semantic reasoning to examine sensor data to see if it is new as well as to determine if it is different (and especially, if it leads to different conclusions) than that stored in the knowledge base. In either case, the semantic reasoning uses first order logic to reason about the validity of the new or changed information with respect to the rest of the knowledge base. In other words, given new or changed information, the system must determine (1) if the new or changed information is valid, and (2) if it is valid, then how much of the existing knowledge base needs to be updated?

The axioms and theories present in the existing knowledge base are used to validate if the new or changed data makes logical sense. This makes use of existing data and relationships in the knowledge base to build assertions and other types of queries to test the implications of the new or changed data. Once the new or changed data are determined to be valid, then additional logic checks the relationships of the changed data to see if those data also need to be changed. Similarly, existing axioms and theories are applied to the new data to hypothesize new relationships.

In general, the new or changed data will either be able to be immediately verified through issuing queries that verify one or more hypotheses about the new or changed data, or they will need further proof. In the former case, the new or changed data are immediately added to the knowledge base. Otherwise, they are marked for verification. This is beyond the scope of this paper; however, the essential point is that this set of processes enables the knowledge base for our system to evolve with experience.

Both this and the model-based translation function use the notion of semantic relatedness [18] to determine the relevance as well as the validity of the sensor information as well as inferences derived from those data. Semantic relatedness enables entities that are semantically related using synonymy (e.g., “bank” and “lending institution”), antonymy (e.g., “accept” and “reject”), and other lexical relationships such as meronymy (e.g., court is a part of government), as well as defined associations (e.g., router uses protocol). Our original work in this area used linguistic analysis; however, this has a high associated degree of computational complexity. We are thus investigating other means, such as using WordNet [19], which provides a set of APIs for computing common linguistic relations, as well as structural matching algorithms. This enables us to move from offline applications, which require on the order of 2-6 hours of computation, to more near-real-time applications.

FOCALE develops and uses a library of models and coded behaviors, much as a library of string processing functions is used by a programming language. This library is made reusable by realizing it in the form of objects, supported by both models and ontologies. The library is a result of the application of policy actions, which in turn are selected by a particular context as previously described.

FOCALE uses the concept of the Policy Continuum [17][20][21], which enables policies written using terminology and concepts for one domain, such as business analysts, to be translated to policies written using a different set of terminology and concepts for another domain, such as programmers. [15] develops a Knowledge Continuum that, similar to the Policy Continuum, defines a continuum for expressing knowledge, from conjectures through facts. This enables context-aware policies to be used to orchestrate behavior for business goals, social interaction, and other forms of interaction.

III. SALIENT FEATURES OF THE INFERENCE PLANE

The Inference Plane was a response to previous future Internet approaches that culminated in the Knowledge Plane [13]; it is shown in Figure 3.

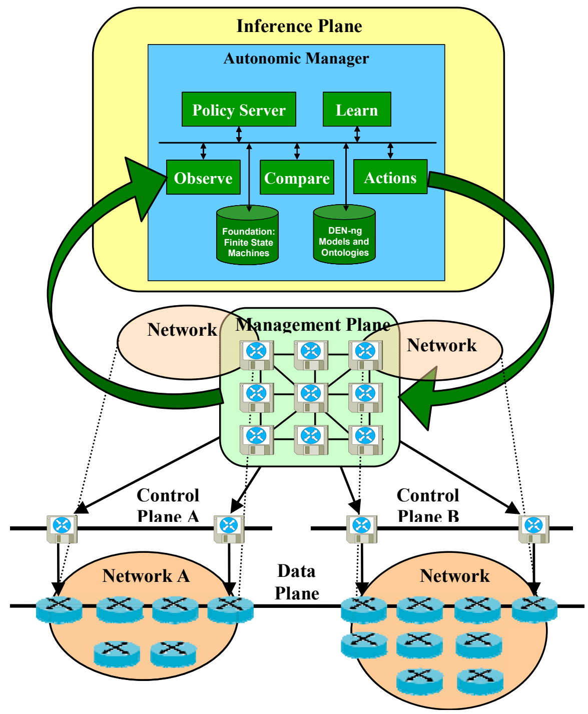

Figure 3. Conceptual Overview of the Inference Plane

The Inference Plane is an approach that enables new ideas, such as clean-slate approaches [3], to peacefully co-exist within an evolutionary framework that is compatible with existing approaches. The ability to maintain backward compatibility is very important to certain stakeholders, such as Internet service providers (ISPs), who have invested billions into their equipment and want to leverage those investments. Hence, the current Data and Control planes, which provide a path for data and services and the functionality governing network connectivity, respectively, are allowed to remain the same. The innovation of the Inference Plane comes in formalizing the Management Plane, which provides functionality governing deployment and operation of network resources and services, and of course the Inference Plane itself, which contains functionality for orchestrating network resource and service behavior according to context-aware policies.

Our approach consists of the combination of a Management Plane and an Inference Plane to manage UbiComp applications and services; we feel that this is an excellent model of Future Internet services. The Management Plane is used to coordinate the functionality applied by different control mechanisms, while the Inference Plane provides the ability to orchestrate decision-making components according to the current context, changing user needs, business objectives, and environmental conditions, and any dynamic constraints that are placed upon network services and resources. This approach was inspired by and grounded in our existing FOCALE autonomic networking architecture. The result is to enable the business to drive the services and resources supplied by the network as context changes, which will be explained in the next section [22].

The Inference Plane is a coordinated set of intelligent decision-making components that represent the capabilities of the computing elements being controlled, the business and technical constraints placed upon using different functions, and the context in which they are being used. This enables the services and resources supplied by heterogeneous networks to be assigned and driven by business goals and rules. It has two complementary but distinct purposes: (1) to analyze the current operation of the systems that it is governing with respect to business and other goals to ensure that the services and resources offered satisfy those objectives, and (2) to coordinate the actions of the Management and Control Planes. Figure 4 shows the use of FOCALE in realizing the Inference Plane.

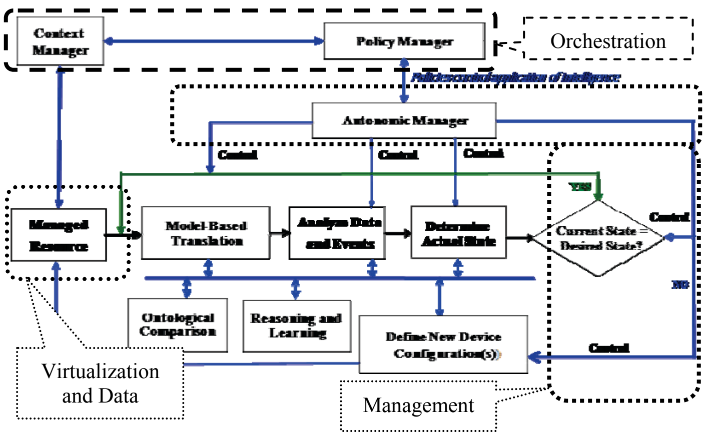

Figure 4. Using FOCALE to Realize the Inference Plane

FOCALE uses the DEN-ng information model to represent the characteristics and static behavior of managed resources. These include virtual (as well as concrete) resources and services. Hence, the Managed Resource could be a physical, virtual, or hybrid managed entity (e.g., a physical resource with virtual device or service support). Since FOCALE is a model-driven architecture, it can dynamically generate code from models. We use this feature to orchestrate behavior by generating commands to change the state of managed entities in real time. The DEN-ng model elements as nodes and edges in our FSMs; context-aware policies are used to control the changing or maintaining of state, thus orchestrating behavior. State transitions are realized by dynamically generating code to change the state of the appropriate managed resources; this is defined by the configuration commands associated with the new state that is being transitioned to.

This enables FOCALE to incorporate virtualization features (which are often associated with *clean-slate* architectures) into an *evolutionary* framework, because resources, whether virtual or physical, are “just” managed entities represented in FOCALE FSMs and the DEN-ng model. In particular, FOCALE uses the concept of an administrative domain from DEN-ng to make this approach easier to implement with existing legacy architectures. The key characteristic of an Administrative Domain is that all of its entities are managed by the same set of policy rules. Hence, if this is combined with FOCALE’s context-aware policy rules, this approach has the ability to use policy rules to orchestrate the behavior of managed entities according to changes in context.

IV. A Future Internet Management Architecture

We believe that current and future networks and networked applications have vastly different requirements; this means that a single architecture cannot simultaneously meet these different needs. Our approach for designing Future Networks is to build a set of *service-aware* networks that can exchange semantic information to enable their collaboration. We realize service-aware networks by extending the FOCALE architecture to manage the difficult demands of UbiComp services as a model for Future Internet services. This takes the concepts of the previous section and defines a set of semantic translators that enable data from a variety of sources to be reasoned about and converted into a normalized form for further processing by the autonomic manager in FOCALE. This is shown conceptually in Figure 5.

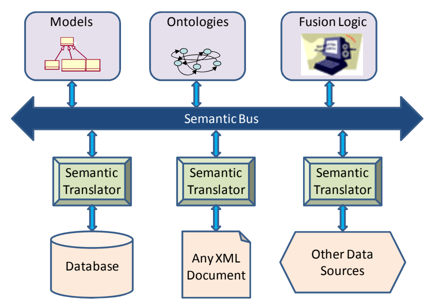

Figure 5. Semantic Translation and Processing

The Fusion Logic in Figure 5 enables vendor-specific data to be translated into machine-understandable semantics. Thus, instead of trying to share fault and other types of data between domains, our approach instead exchanges semantic data describing the *effect* of those data. For example, consider two network devices from different network manufacturers. Each cannot understand the configuration data of the other. Our approach *understands* what the configuration is being used for, and instead passes the appropriate semantic data such that the FOCALE autonomic manager can generate appropriate vendor-specific configuration data.

Existing networks are divided into administrative domains. We support and enhance this idea, by ensuring that each domain has a set of context-aware policies that it uses to govern the functionality that it offers. This is shown conceptually in Figure 6. The concept of the current data plane is enhanced by supporting virtual devices and services. Hence, in Figure 6, a physical device can support multiple virtual devices, any of which can be part of an individual data plane. It is important to note that the administrative domain includes a set of physical devices, and therefore the associated virtual devices that are hosted by those physical devices. The set of physical devices that are subject to the same policies (which are now context-aware) are grouped into administrative domains; a FOCALE instance governs each administrative domain using appropriate control mechanisms that are coordinated by autonomic management loops. This is similar to the concept of existing control planes, except that now, each of the different control plane functions are governed by a single management function. In addition, autonomic control planes differ in that they are able to self-manage and -configure many different functions based on semantics, a feature that is notably lacking in existing control plane implementations. Finally, a FOCALE instance orchestrates the overall functionality of the administrative domain according to business goals and objectives; these are represented using the concept of the Policy Continuum, which enables policies written using terminology and concepts for one domain, such as business analysts, to be translated to policies written using a different set of terminology and concepts for another domain, such as programmers.

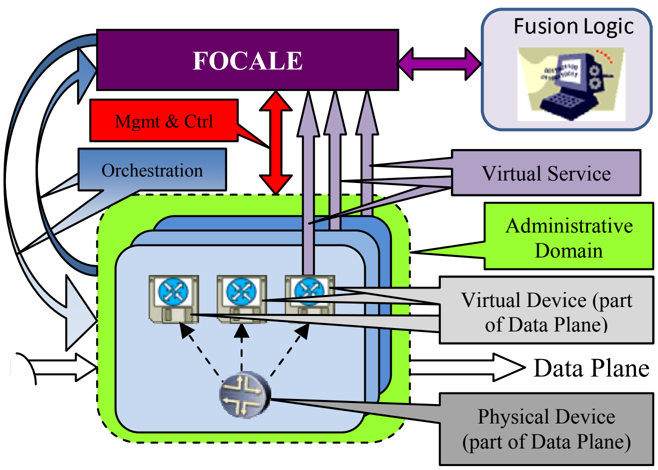

Figure 6. Service-Aware Domain for the Future Internet

Figure 7 shows how service-aware networks are built. A set of administrative domains are concatenated to provide a data plane from the source to the desired destination(s). The set of resources are not important, as they are hidden from the user. Furthermore, the abstraction of an administrative domain is independent of service provider or other governing body; in fact, the only important feature of an administrative domain is that it uses context-aware policies to manage the services it provides. This paradigm can support current Internet services, which are dependent on a particular Service Provider that the user subscribes to, as well as a new genre of services that are not as restricted as this business model. This is inspired by UbiComp applications, where the user is interested not in a particular service provider, but rather in the data and services that can be used. These are represented conceptually as the set of Managed Service(s) that are delivered by the Master FOCALE instance, and is the subject of a separate paper.

Each administrative domain is managed as described above and depicted in Figure 6, and exchanges semantic descriptions of its state, services, and other pertinent data to other FOCALE instances that are interested in the services and/or resources managed by it. This enables the FOCALE instances to collaborate and help each other provide high-value services.

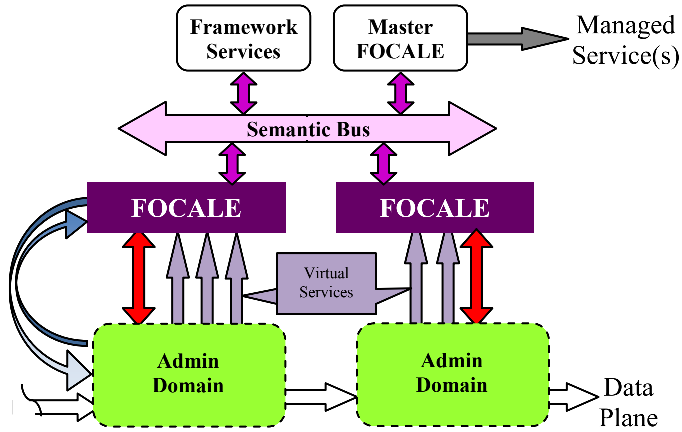

Figure 7. Service-Aware Future Internet Architecture

A set of Framework Services provide the infrastructure necessary to support a distributed implementation. As a minimum, they include registration services (used to support location transparency), repository services (used to provide a consistent logical view of all managed entities in the system, according to access privileges of the user and policies of the system), naming services (used to generate and resolve unique names for entity instances contained in the repository), policy management services (used to coordinate the behavior of the entities in different administrative domains), and security services (used to ensure that the operation and behavior of the system does not violate any security policies).

A set of Federation Services enables two types of federation to be performed: (1) based on services offering similar functions (service level federation), or (2) based on matching the underlying data (both system and user) to be shared (repository level federation). The Framework and Federation Services are connected by a bus, so that the FOCALE system that is governing the set of lower-level FOCALE-based administrative domains can use the Framework and Federation Services as needed.

The architecture is conceptually similar to a programmable grid, except that our architecture uses a model-based approach [12][15][17] to build reusable libraries of behavior that are then matched based on their semantics. This combines the features of Service Oriented Architectures with the ability to dynamically generate configuration changes based on business needs [15]. UbiComp applications express their needs to our system using a semantic description of the services that they want to use; this is then parsed and the system allocates appropriate resources based on first, the semantic description of any existing networks that support similar services; second, the ability to construct a service from its behavior libraries that will satisfy the needs of the application; third, the ability to dynamically build a new service that satisfies the needs of the application.

[26] described an implementation and associated simulation for FOCALE's policy-based architecture for end-to-end network management, while [27] describes a set of extensions to [26] that describe our model-based approach to policy tool and language generation. This approach defines a set of

Domain Specific Languages to represent the needs of different constituencies, and makes use of a set of novel policy tools that were constructed from open source platforms. We are currently extending both of these implementations, as well as developing a set of policy languages, to automate the transformation of concepts between different levels of the Policy Continuum.

V. SUMMARY AND FUTURE WORK

This paper has defined a new management architecture for the Future Internet, using the demands of UbiComp applications as a model. It has several unique aspects. First, it provides an evolutionary framework for managing Future Internet services by extending the FOCALE model-based approach, which can dynamically generate code to make configuration changes. Second, this evolutionary framework provides the ability to lower operational expenditures by automating configuration tasks. Third, our approach addresses business, socio-political, and other aspects (in addition to technical aspects) by representing knowledge and policies along continua that enable different behavior to be orchestrated based on changing context. Fourth, our solution enables clean-slate approaches to be tested within a stable, evolutionary technical framework; an example is the seamless management of physical, virtual, or hybrid resources. Finally, by using the notion of semantic translators, our solution does not require vendors to retool their existing network devices. Rather, our solution is conceptually an intelligent middleware that bridges the different languages and associated semantics used by existing network vendors and produces a single lingua franca that enables diverse data to be combined and harmonized.

Future work will be done in multiple areas. Our first will be exploring how management decisions can be made in the face of uncertain, inaccurate, and/or incomplete data. Important tasks include determining when management data is corrupted, taking into account probabilistic behavior, and detecting cheating and falsification of data. Second, we will validate our semantic translators over a broader range of devices and device operating systems. Third, we will conduct more detailed experimental data across a wider variety of networks.

ACKNOWLEDGMENT

This work is partially sponsored by Science Foundation Ireland under under grant numbers 03/CE3/I405 (AMCNS) and 08/SRC/I1403 (FAME).

REFERENCES

[1] D. Clark, et al., “NewArch: Future Generation Internet Architecture”, NewArch Final Technical Report, http://www.isi.edu/newarch/.

[2] Blumenthal, M., Clark, D., “Rethinking the design of the Internet: the end to end arguments vs. the brave new world”, ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 70-109, Aug. 2001.

[3] Feldmann, A., “Internet clean-slate design: what and why?,” ACM SIGCOM Computer Communication Review, Vol. 37, Issue 3, July 2007.

[4] Future Internet Network Design (FIND), http://find.isi.edu.

[5] Global Environment for Network Innovations, http://www.geni.net.

[6] http://www.future-internet.eu/

[7] Future Internet Research and Experimentation (FIRE), http://cordis.europa.eu/ist/fet/comms-fire.htm.

[8] AKARI, “Architecture Design Project for New Generation Network,” http://akari-project.nict.go.jp/eng/index2.htm.

[9] Future Internet Forum (FIF), http://mmlab.snu.ac.kr/fif.

[10] M. Weiser, “Open House”, in Review, the web magazine of the Interactive Telecommunications Program of New York University, March, 1996.

[11] J. Strassner, M. Ó Foghlú, W. Donnelly, N. Agoulmine, “Beyond the Knowledge Plane: An Inference Plane to Support the Next Generation Internet”, IEEE Global Information Infrastructure Symposium (GIIS 2007), 2-6 July, 2007, pages 112-119.

[12] J. Strassner, N. Agoulmine, E. Lehtihet, “FOCALE – A Novel Autonomic Networking Architecture”, ITSSA Journal, Vol. 3, No. 1, May 2007, pages 64-79, ISSN 1751-1461

[13] D. Clark, C. Partridge, J. Ramming, and J. Wroclawski, “A Knowledge Plane for the Internet”, Proceedings ACM SIGCOMM 2003

[14] J. Strassner, J. de Souza, D. Raymer, S. Samudrala, S. Davy, K. Barrett, “The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking”, LANOMS 2007, pages 114-125, ISBN 9781424411825

[15] J. Strassner, “Enabling Autonomic Network Management Decisions Using a Novel Semantic Representation and Reasoning Approach”, Ph.D. thesis, 2008

[16] J. Strassner, Y. Liu, M. Jiang, J. Zhang, S. van der Meer, M. Ó Foghlú, C. Fahy, W. Donnelly, “Modelling Context for Autonomic Networking”, 5th IEEE International Workshop on Management of Ubiquitous Communications and Services (MUCS), April 11, 2008, Brazil

[17] J. Strassner, “Autonomic Networking – Theory and Practice”, 20th Network Operations and Management Symposium (NOMS) 2008 Tutorial, Salvador Bahia, Brazil, April 7, 2008

[18] A. Budanitsky, “Lexical semantic relatedness and its application in natural language processing” Technical Report CSRG390, University of Toronto, 1999

[19] http://wordnet.princeton.edu/

[20] J. Strassner, “Policy Based Network Management”, Morgan Kaufman, ISBN 1-55860-859-1

[21] S. Davy, B. Jennings, J. Strassner, “The Policy Continuum – A Formal Model”, in Proc. of the 2nd International IEEE Workshop on Modelling Autonomic Communications Environments (MACE), eds. B. Jennings, J. Serrat & J. Strassner, Multlicon Lecture Notes – No. 6, Multicon, Berlin, 2007, pages 65-78

[22] J. Strassner, "How Policy Empowers Business-Driven Device Management," in: Third IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'02), June 5-7, 2002

[23] R. Wieringa, J. Gordijn, P. van Eck, “Value-based Business-IT Alignment in Networked Constellations of Enterprises”, Proceedings of 1st International Workshop on Requirements Engineering for Business Need and IT Alignment, Pages 38-43

[24] P. DiMaggio, E. Hargittai, W. R. Neuman, J. P. Robinson, “Social Implications of the Internet”, Annual Review of Sociology, August 2001, Vol. 27, Pages 307-336

[25] G. A. Moore, “Technology and social communication”, Technology in Society, Vol. 18, Issue 2, Pages 253-260

[26] S. van der Meer, A. Davy, S. Davy, R. Carroll, B. Jennings, J. Strassner, “Autonomic Networking: Prototype Implementation of the Policy Continuum”, IEEE 1st International Conference on Broadband Convergence Networks, pages 163-172, April 7, 2006, Vancouver, Canada

[27] K. Barrett, S. Davy, J. Strassner, B. Jennings, S. van der Meer, and W. Donnelly, “A Model Based Approach for Policy Tool Generation and Policy Analysis”, 1st International Global Information and Infrastructure Symposium, pages 99-105, July 2-6, 2007, Marrakech, Morocco