Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 8 (2012) 177 – 183

Procedia
Computer Science

New Challenges in Systems Engineering and Architecting
Conference on Systems Engineering Research (CSER)
2012 - St. Louis, MO
Cihan H. Dagli, Editor in Chief
Organized by Missouri University of Science and Technology

A Dependability Assessment Process for Ensuring Consistent Provisioning of Network Recovery

Joseph Kroculick[a,1,], Cynthia Hood[b]

[a] *127 White Oak Drive, Jim Thorpe, PA*
[b] *10 West 31st Chicago, IL*

Abstract

We have developed an engineering method to detect errors in provisioning automated recovery processes in multilayer and multi-protocol communications transport networks. Our dependability assessment process leverages inference techniques provided by Semantic Web technologies in order to detect network-device provisioning errors. Provisioning should be accompanied by methodologies, processes, and activities to ensure that it can be trusted to achieve a desired network state. Our method takes into account unique constraints in the telecommunications domain including bottom-up evolution of physical layer technologies to provide connectivity and lack of a universal model of network functionality. We apply our method to assessing the correctness of provisioning decisions for a protection switching application in a transport network in both the spatial and temporal domains.

Keywords: Assurance Case, Dependability Case, Network Provisioning

1. Introduction

Network management of mission-critical services will increasingly rely on gathering network state information and sending configuration commands to the interfaces of many different heterogeneous devices [11]. As the number of types of devices and device components increase, it will become necessary to define more scalable and automated verification tools and methodologies that can ensure that the network infrastructure will operate consistent with end-to-end service requirements. According to Deca, Cherkaoui and Puche [5], there are complex dependencies between service parameters and device state and parameters. Furthermore, the overall ”quality of network (QoN)” or an end-to-end network dependability is influenced

[*] Principal corresponding author
Email addresses: krocjoe@winifredassociates.com (Joseph Kroculick), hood@iit.edu (Cynthia Hood)
[1] This paper is a summary of Joseph Kroculick's thesis research at the Illinois Institute of Technology

1877-0509 © 2012 Published by Elsevier B.V. Open access under CC BY-NC-ND license.
doi:10.1016/j.procs.2012.01.036

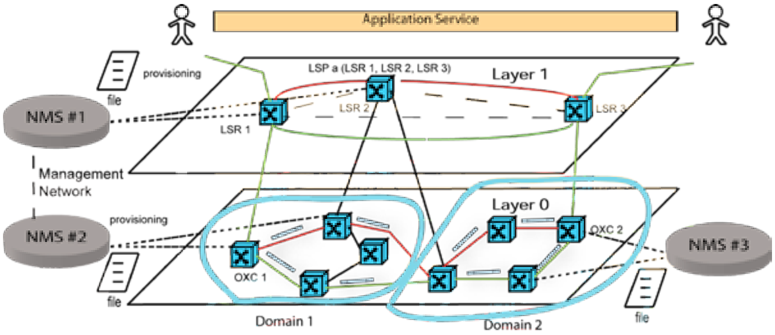


Fig. 1. Multilayer Provisioning Infrastructure

by a combination of hardware, software, network operations procedures and human intervention [14]. A formal model of a network should be flexible to allow dependability properties to be demonstrated as a network administrator adjusts device settings of a network. Verification of declarative properties using inference can be applied to a logical model of a network to demonstrate important structural and temporal requirements [8].

Predicting the behavior of multiple types of resources requires "reasoning and computation" over information, which is distributed across multiple systems and components [17]. The current approach to verification of services at the device level is to provide set of technology-specific, layer-specific, and device-specific tools that don't integrate well. Furthermore, many individual configuration settings need to be applied to multiple distributed objects and fragmented device information models to successfully set a service parameter [17]. As an example of the complexity involved in provisioning end-to-end recovery service, consider Figure 1, which depicts configuration of devices using device-specific settings and attributes. Each of these settings needs to cause recovery operations to occur to achieve an end-to-end network configuration that provides connectivity between access ports, where traffic is injected and extracted from a network.

This paper is is organized into a discussion of assurance-based design and its application to provisioning network infrastructure functionality. In this section, we lay the ground work to build a service offering to a customer from setting the parameters of heterogeneous distributed resources and device functions such as networking protocols. Section 2 discusses assurance-based design and dependability or assurance cases. Section 3 presents an overview of the activities of our network engineering method to provide confidence that the available network infrastructure functionality will be provisioned to achieve an end-to-end service requirement such as service continuity. In Section 4, we summarize our work and discuss opportunities to build temporal and structural domain ontologies to represent network recovery functionality and end-to-end relationships such as connectivity between customer end-points.

2. Applying Assurance-Based Design to Network Provisioning Using Ontologies

We wish to demonstrate that a given network infrastructure service can be made trustworthy to perform as expected. From the grouping of dependability attributes proposed by Norros, Kuusela, and Savola, we select the "controllability of protocols" and align it with the topic of configuration [15]. Configuration reconfigures tunable parameters in system functions to achieve a new behavior or system structure [21]. The DAP then applies dependability case technique [10] to testing overall service dependability from the configuration of device and associated protocol parameters. The service dependability is then affected by identifying configurations that don't meet the service requirement and then using a knowledge model to infer resource relationships from documented relationships.

We apply an assurance-based design approach to validating network provisioning based on assurance cases to detecting misconfigurations of network recovery functionality in a multilayer system in order to prove that a provisioned network configuration will meet dependability claims. An *assurance-based design* approach is characterized by a system model accompanied by an assurance case [7]. In an assurance case, argumentation is created from a set of facts to demonstrate the validity of one or more requirements [19, 2]. When the requirement being demonstrated is an aspect of dependability of a system or service, the assurance case is known as a "dependability case" [10]. Dependability can be defined as "the ability to deliver service that can justifiably be trusted" to perform consistent with the expectations of a service consumer [1]. Dependability involves multiple aspects that are often interdependent such as reliability and availability [1].

3. Dependability Assessment Process

The Dependability Assessment Process (DAP) is a process designed to assess the dependability of network infrastructure functionality that is provisioned manually by an end-user in order to meet a service requirement. The DAP replaces the argumentation step of a dependability case [10] with reasoning over over an integrated ontology, which organizes dependability requirements, network infrastructure concepts and error categories [1]. The goal of the process is to detect provisioning errors in multiple distributed components that implement a network service. A family of ontologies of dependability terms, network infrastructure concepts, and conflict classes are integrated to produce a complex statement that can be checked to determine whether a provisioning conflict exists.

3.1. Knowledge Management

Our approach to knowledge management progressively elaborates network information in a family of ontologies that are integrated with each other. Ontologies provide semantic descriptions of the application domains and connect knowledge by providing a logic-based knowledge representation formalism that allows inferences to be drawn. They also allow the end-user to think in terms of the essential aspects of the system. Ontologies provide a set of axioms that support logical reasoning over a set of facts [18, 9]. Network concepts can include anything of importance in the network domain such as a transmission unit of traffic or a set of links that are bundled together.

Assertions about network resources are declared and represented as relationships between a subject and an object in a simple statement. The syntactic form of the statement is represented as a triple [20]. A triple is a three-tuple consisting of a subject, predicate and object. The subject represents a concept or entity in the network. The predicate represents a property of the subject. The object represents another concept or entity that is related to the subject. Subject properties are limitless allowing for complex relationships to be expressed.

3.2. Dependability Assessment Process Steps

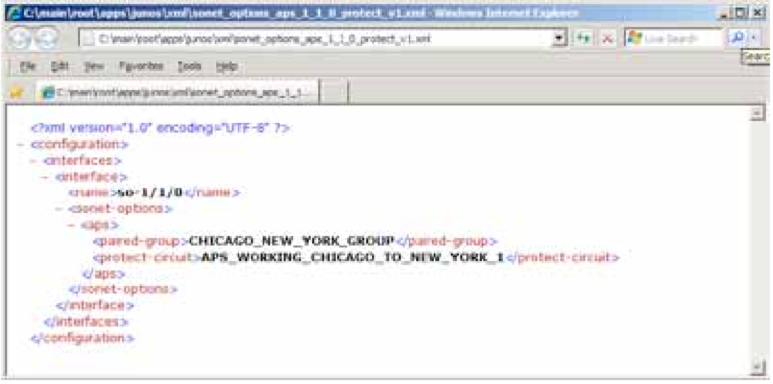
The dependability assessment process consists of a set of activities that produce multiple artifacts as their output. A logical model of a network is expressed in a modeling language based on ontologies so that constraints can automatically be checked through inference. The process requires models to be validated and transformation practices to convert the device specific representation or physical model into an updated logical model. The DAP can be depicted as a UML activity diagram as shown in Figure 5.

3.2.1. Define the Problem Context

During the define problem context activity, the assumptions, constraints, and requirements for a network architecture are expressed. Such constraints include the availability of technologies, architectural alternatives, costs of implementation, and requirements of the end-user.

3.2.2. Build Network Domain Model

The build domain model activity produces a logical model of network infrastructure that represents essential resource types and relationships between them as an ontology.


C:\main\root\apps\junos\xml\sonet_options_aps_1_1_0_protect_v1.xml - Windows Internet Explorer
C:\main\root\apps\junos\xml\sonet_options_aps_1_1_0_protect_v1.xml
Live Search
File
Edit
View
Favorites
Tools
Help
C:\main\root\apps\junos\xml\sonet_options_aps_1_1...

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <configuration>
  - <interfaces>
    - <interface>
        <name>so-1/1/0</name>
      - <sonet-options>
        - <aps>
            <paired-group>CHICAGO_NEW_YORK_GROUP</paired-group>
            <protect-circuit>APS_WORKING_CHICAGO_TO_NEW_YORK_1</protect-circuit>
          </aps>
        </sonet-options>
      </interface>
    </interfaces>
  </configuration>
```

Fig. 2. Example JUNOS XML Fragment for SONET Automatic Protection Switching

3.2.3. *Specify Service Requirements*

The specify service requirements activity specifies desired functionality provided by a system to an end user [12]. A service level agreement (SLA) provides actual terms of service delivery including a service level specification and as service level objectives [13]. A network service is a path through a network over which traffic can flow.

3.2.4. *Identify Dependability Property*

The identify dependability property activity provides a tangible property that can be checked. To automate network dependability assessment, an observable and measurable property needs to be checked.

3.2.5. *Specify Provisioning*

The specify provisioning activity specifies provisioning commands expressed in terms of the router configuration language interface. Each device configuration is established by modifying the value of different parameters attached to network resources using commands applied at the operational interface.

3.2.6. *Build Transformation*

The build transformation activity maps an XML device model to an OWL/RDF domain model. This step builds semantics to device-specific tags and relate them to OWL classes and properties. An example XML fragment for the JUNOS `sonet-options` tag [3] is depicted in Figure 2.

3.2.7. *Update Domain Model*

The update domain model activity converts provisioning commands to changes to a logical model of the domain of interest.The effect of the recovery operations needs to be predicted and determined at provisioning time and the logical model needs to reflect the network state at the time the actions are performed such as when a failure occurs.

3.2.8. *Define Conflict Statement*

The define conflict statement activity builds a single triple representing a formal dependability property, which is a single measurable property of the network infrastructure[16] from individual triples which represent relationships between local properties such as interfaces or timestamps. The formal dependability property holds when the condition is true as depicted in Figure 3. The condition is a goal that holds when the entailed triple is found in an inferred graph from individual triples.

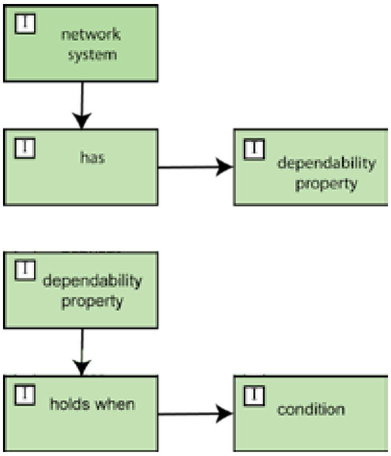


Fig. 3. Triples to be connected by merging statements

3.2.9. Detect Conflict

The detect conflict activity queries an inferred ontology model for whether an entailment or inferred triple exists in the corresponding RDF graph. The entailment could represent an inferred end-to-end ”is-connected-to” relationship between a source and destination vertex. Conflict detection is implemented as a SPARQL query on an inferred RDF graph. An example query is represented in Figure 4.

3.2.10. Validate Model

The validate model activity allows the system context to be updated as knowledge is discovered about the problem context through addition of new domain knowledge, service requirements, or provisioning provided by a human operator. The knowledge to represent devices can change as new technologies are developed or improved modeling concepts are derived.

4. Conclusion

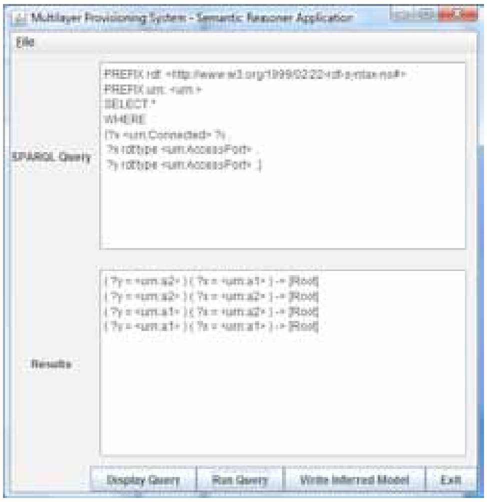
We have developed a dependability assessment process designed to assess the dependability of network infrastructure functionality that is provisioned manually by an end-user in order to meet a service requirement. The DAP uses automated reasoning to check for provisioning errors in networks. Our process provides an automated argumentation process to check the result of provisioning commands against a logical model and provide assurance that a correct network is deployed in a router configuration language. Semantic interoperability between router provisioning options is a crucial element in achieving consistent behavior in implementing a network service such as recovery. A common well-defined terminology can help to validate that the network infrastructure will implement options in standards consistently in providing a crucial network service such as recovery.

Reasoning in ontologies automates the assurance-case argumentation process where evidence is chained to dependability subgoals in a dependability case. A knowledge representation method that uses ontologies allows network infrastructure concepts and requirement concepts to be merged and provides verification of key properties that have been identified in well-engineered networks.

References

[1] Avižienis, A., Laprie, J-C., Randall B., Landwehr, C. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE Transactions on Dependable and Secure Computing, 2004, Vol. 1, Pages 11 – 33, January-March

[2] Bloomfield, R., Masera, M., Miller, A., Sydjari, O.S., Weinstock, C. B. (2007). Assurance Cases for Security: The Metrics Challenge, 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN’07)



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX urn: <urn:>
SELECT *
WHERE
{?x <urn:Connected> ?y .
 ?x rdf:type <urn:AccessPort> .
 ?y rdf:type <urn:AccessPort> .}
```

```
( ?y = <urn:a2> ) ( ?x = <urn:a1> ) -> [Root]
( ?y = <urn:a2> ) ( ?x = <urn:a2> ) -> [Root]
( ?y = <urn:a1> ) ( ?x = <urn:a2> ) -> [Root]
( ?y = <urn:a1> ) ( ?x = <urn:a1> ) -> [Root]
```

Fig. 4. Example SPARQL query of a network ontology for end-to-end connectivity between customer endpoints.

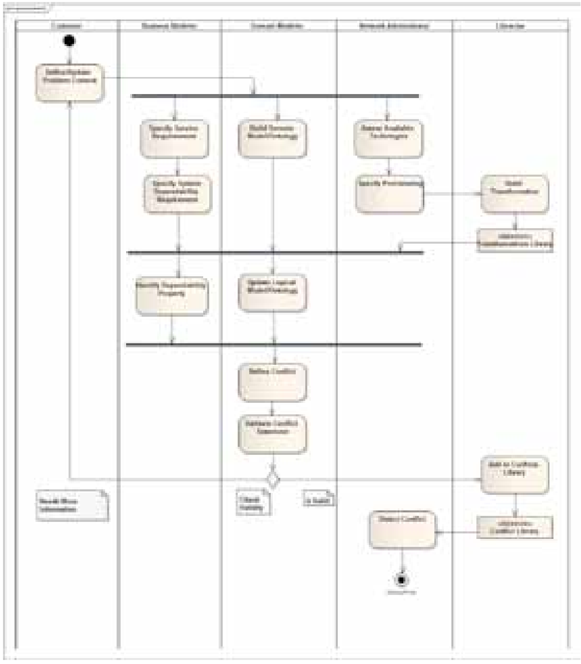

Customer
Business Modeler
Concept Modeler
Network Administrator
Librarian
Specify Service Requirements
Build Service Model/Ontology
Assess Available Technologies
Specify System Dependability Requirements
Specify Provisioning
Build Transformation
«datastore» Transformations Library
Specify Dependability Property
Update Logical Model/Ontology
Define Conflict
Validate Conflict Signature
Add to Conflicts Library
«datastore» Conflict Library
Detect Conflict
Need More Information
Check Validity
is Valid

Fig. 5. Dependability assessment process for detecting provisioning errors. There are multiple concurrent phases, actors, and activities to connect requirements, technologies, libraries, models, and operator input.

[3] Bushong, M., Gadecki, C. & Garret, A. (2008). JUNOS For Dummies, Wiley Publishing, Inc

[4] CMMI Product Team (2010). CMMI for Services, Version 1.3, no. CMU/SEI-2010-TR-034, November, http://www.sei.cmu.edu

[5] Deca, R., Cherkaoui, O., & Puche, D. (2004). A Validation Solution for Network Configuration, Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04), IEEE

[6] Dobson, S., Sterritt, R., Nixon, P., & Hinchey, M. (2010). Fulfilling the Vision of Autonomic Computing, January 2010, Computer

[7] Graydon, P., Knight, J., &Strunk, E. (2007). Assurance-Based Development of Critical Systems, 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)

[8] Hoffman, L. (2008). Talking Model-Checking Technology, Communications of the ACM, July 2008, Vol. 51., No. 7

[9] Horrocks, I. (2008). Ontologies and the Semantic Web, Communications of the ACM, December 2008, Vol. 51, No. 12

[10] Jackson, D. (2009). A Direct Path to Dependable Software, Communications of the ACM,, vol. 52, no. 4, April 2009

[11] Nguyen, K., Mahkoum, H., Jaumard, B., Assi, C. & Lanoue, M. (2007). Toward a Distributed Control Plane Architecture for Next Generation Routers, Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN'07), IEEE

[12] Maiden, N. (2006). Servicing Your Requirements, IEEE Software, September/October, 2006

[13] Meddeb, A. (2010). Internet QoS: Pieces of the Puzzle, IEEE Communications Magazine, January 2010

[14] Network Maturity Model Development Team (1999). The Network Maturity Model for Internet Development, IEEE Computer

[15] Norros, I., Kuusela, P., Savola, P. (2008). A Dependability Case Approach to the Assessment of IP Networks, The Second International Conference on Emerging Security Information, Systems and Technologies, IEEE

[16] Paul, R., Yen, I-L., Bastani, F., Dong, J., Tsai, W-T., Kavi, K., Ghafoor, A., & Srivastava, J. (2008). An Ontology-Based Integrated Assessment Framework for High-Assurance Systems, The IEEE International Conference on Semantic Computing, 2008

[17] Sollins, K. R. (2009). An Architecture for Network Management, ReArch'09, ACM

[18] Spear, A.D. (2006). Ontology for the Twenty First Century: An Introduction with Recommendations, Saarbruucken, Germany, 2006

[19] Strunk, E. A., Knight, J. C. (2006). The Essential Synthesis of Problem Frames and Assurance Cases, IWAAPF'06, 2006, ACM

[20] W3C (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax (2004). W3C, W3C Recommendation, February 2004

[21] Wang, Y., Keller, E., Biskeborn, B., van der Merwe, J., Rexford, J. (2008). Virtual Routers on the Move: Live Router Migration as a Network Management Primitive, SIGCOMM'08, ACM