Reliability Engineering and System Safety 132 (2014) 220–232

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Automated synthesis of dependable mediators for heterogeneous interoperable systems

F. Di Giandomenico [a], M.L. Itria [a], P. Masci [c], N. Nostro [a,b,*]

[a] ISTI-CNR, Pisa, Italy
[b] University of Florence, Italy
[c] Queen Mary University of London, United Kingdom

ARTICLE INFO

Article history:
Received 10 January 2013
Received in revised form
30 July 2014
Accepted 9 August 2014
Available online 19 August 2014

Keywords:
Model-based analysis
Dependability
Performance
Dependability mechanisms
Dynamic networked systems
Interoperability

ABSTRACT

Approaches to dependability and performance are challenged when systems are made up of networks of heterogeneous applications/devices, especially when operating in unpredictable open-world settings. The research community is tackling this problem and exploring means for enabling interoperability at the application level. The EU project Connect has developed a generic interoperability mechanism which relies on the on-the-fly synthesis of "Connectors", that is software bridges that enable and adapt communication among heterogeneous devices. Dependability and Performance are relevant aspects of the system. In our previous work, we have identified generic dependability mechanisms for enhancing the dependability of Connectors. In this work, we introduce a set of *generic strategies* for automating the selection and application of an appropriate dependability mechanism. A case study based on a global monitoring system for environment and security (GMES) is used as a means for demonstrating the approach.

1. Introduction and motivation

The classic and well understood way of building dependable systems [1] is based on the application of rigorous development methods. Special programming techniques are used for software, such as model-driven development [2], and specific architectures are used for hardware, such as modular redundancy [3]. Dependability-critical domains, such as avionics and power plants, *require by law* the adoption of these techniques, and define standards that must be followed.

This classic approach to dependability is challenged when critical systems are made up of networks of heterogeneous devices from different manufacturers. The GMES (Global Monitoring for Environment and Security) European Programme for the establishment of a European capacity for Earth Observation provides an excellent example of heterogeneity in interoperable applications and devices for critical applications. It started in 1998, and includes six main thematic areas: land monitoring, marine environment monitoring, atmosphere monitoring, emergency management, security and climate change. The emergency management service directs efforts towards a wide range of emergency situations; in particular it covers different catastrophic circumstances: floods, forest fires, landslides, earthquakes and volcanic eruptions and humanitarian crises. As another example, in the healthcare domain, currently there is not a standard for medical device interoperability. Nevertheless, this has not prevented the adoption in hospitals of networks of heterogeneous technologies. In some cases the lack of interoperation just causes minor disturbances, e.g., patients not recognised by palm-sized wireless medical devices because the devices are not enabled to gather this information on-the-fly from a central database [4]. In other cases, problems are more serious, e.g., surgical fires caused by lack of dependable interoperation between electrosurgical devices and oxygen-delivery devices [5].

The problem is that standardised interoperability at the application level is essentially *non-existent*. In fact, standards like Universal Serial Bus (USB) and IEEE 802.11 (WiFi) enable interoperability at a level *lower* than the application logic. The consequence of this is that heterogeneous networked devices might be able to interoperate but at the same time they might not be able to fully benefit from each other's services. This situation might create serious problems, e.g., in safety-critical systems safety interlocks defined at the application level may be ignored or overridden. Even more challenging is the situation where the heterogeneous systems have a dynamic and evolving behaviour, thus requiring adaptation if interoperability is to be enabled.

* Corresponding author at: ISTI-CNR, Pisa, Italy.
E-mail addresses: f.digiandomenico@isti.cnr.it (F. Di Giandomenico), massimiliano.leone.itria@isti.cnr.it (M.L. Itria), paolo.masci@eecs.qmul.ac.uk (P. Masci), nicola.nostro@unifi.it (N. Nostro).

http://dx.doi.org/10.1016/j.ress.2014.08.001
0951-8320/© 2014 Elsevier Ltd. All rights reserved.

1.1. Problem statement

The problem at stake is *application-level interoperability* in networks of heterogeneous devices. Interoperability is the ability for a device to connect to or be used with another device, and perform individual functions without alteration of the individual device. In heterogeneous networks, devices may have compatible transmitters and receivers that allow us to exchange messages, but interoperability may still be *not* enabled because of mismatches in communication protocols used at the application level by the devices. For instance, assume two devices $D1$ (e.g., a mobile phone) and $D2$ (e.g., a printer) which need to interoperate to accomplish a task (e.g., print an electronic document). If $D1$ requires handshake H 1 to start communication, and device $D2$ only accepts handshake H 2, then the two devices are not able to interoperate at the application level, and so the task cannot be accomplished. Another example is when the two devices have identical protocols but different dependability or performance requirements. For instance, device $D1$ requires maximum latency X, but device $D2$ can only guarantee latency $Y > X$ during communication. In this case too, the application-level interoperability is not possible.

To enable interoperability, different communication protocols need to be *harmonised*. A generic approach to harmonise heterogeneous communication protocols relies on the synthesis of *mediators* that bridge functional (i.e., semantic) gaps between communication protocols, and non-functional (i.e., dependability- or performance-related) mismatches between protocols.

Generic approaches for addressing functional mediation have been investigated since Yellin and Strom's seminal work on component adaptors [6]. Communities that are particularly active on this topic are those of Self-Adaptive Systems (e.g., see [7,8]) and Service Oriented Architectures (e.g., see [9–11]).

Generic approaches for addressing non-functional mediation have been largely neglected. Only few examples can be found in the literature. In [12], the control science theory is used to define a framework with composable modules and an overlay of agents that enable security, privacy and dependability in heterogeneous networks of embedded systems. Another example is [13], where an approach based on stochastic modelling is explored to synthesise mediators that meet given performance requirements.

In our previous work [14,15] we have presented a generic model-based framework to support the synthesis of *dependable mediators*, that is mediators that meet dependability and performance requirements. Recently, in [16], we have identified generic templates which can be used in several practical cases to enhance the dependability and performance level of synthesised mediators. In this work, we unify the two contributions of our previous works, and illustrate in detail the model-based approach used to automate the synthesis of dependable mediators.

1.2. Contribution

The contributions of this work are (i) an automated approach to select and instantiate generic dependability and performance templates during the synthesis of dependable mediators; (ii) a detailed example based on a global monitoring system for environment and security (GMES) that demonstrates the proposed approach.

1.3. Structure of the paper

The presentation proceeds as follows. In Section 2, we provide an overview of the Connect framework, as this work is contextualised within it. In Section 3, the performed model-based dependability analysis is presented. In Section 4, we illustrate the proposed generic methodology for selecting dependability mechanisms in networks of heterogeneous interoperable devices. In Section 5, we demonstrate the benefits of the proposed approach within an example based on a global monitoring system. The selected scenario is one of the demonstrative examples developed in the Connect project. Section 6 describes related work and conclusions are drawn in Section 7.

2. Context

The context of this work is that of Connect,[1] a research project that explored generic approaches to the automated synthesis of "Connectors", software mediators that enable application-level interoperability. A model-based approach is used to identify gaps and mismatches between communication protocols, and then generate a Connector that bridges the identified gaps and mismatches. Modelling is composed of two phases: (i) building of a model that reflects the behaviour of the components of the system and their interactions; (ii) analysis of the model to obtain a Connector that enables application-level interoperability. The Connect framework supports this model-based approach using an overlay network of five types of active units: Discovery, Learning, Synthesis, Dependability, and Monitoring. The role of these five units is now illustrated.

Discovery and Learning: These units gather information about functionalities requested and provided by networked systems. Specifically, the Discovery unit discovers mutually interested devices, and retrieves information about their interface behaviours. The unit assumes that devices are discovery enabled, i.e., they provide a minimal description of their intent and functionalities. When a networked system just provides a partial specification of its behaviour, the Learning unit completes the specification through a learning procedure (e.g., usage on model-based testing and model inference [17]).

Synthesis: This unit performs the dynamic synthesis of mediating Connectors to enable *functional* interoperation among mutually interested devices. The unit performs a graph-based analysis to identify mismatches between the communication protocols identified by Discovery and Learning. A formal definition of the synthesis approach has been presented in [11]. The approach consists of the following steps:

1. The functional specification of the protocols identified by Discovery and Learning is translated into Labelled Transition Systems (LTSs). An LTS is a directed labelled graph used to represent state machines: nodes in the graph represent machine states; directed edges represent transitions between states; labels on the edges identify the event that triggers the transition. In this case, nodes represent protocol states.
2. Ontologies [18] are used to establish a mapping relation between events in the heterogeneous protocols.
3. Communication protocols are "sliced" according to the mapping relation, and the trace of events is systematically generated for each slice. Differences between traces generated for corresponding slices identify mismatches between the protocols.
4. A new LTS (the Connector) is generated to reconcile mismatching traces and thus enable interoperability.

Dependability: This unit supports Synthesis during the generation of Connectors to estimate whether given *non-functional* requirements are met by the synthesised Connector. To this end, the unit performs a stochastic model-based analysis that takes into account the structure of the synthesised Connector and the

[1] http://www.connect-forever.eu