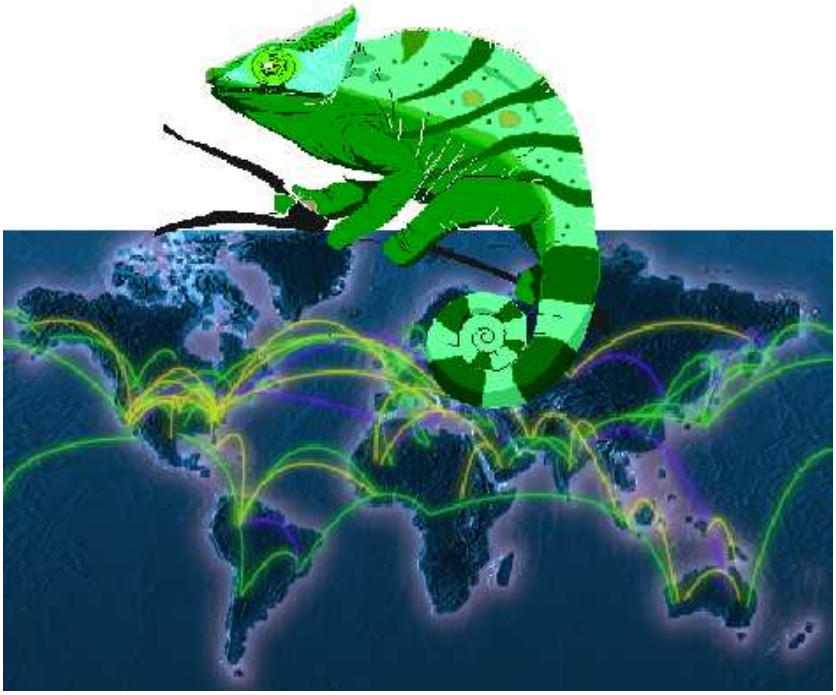
Volume 2, September 2007

DESEREC NEWSLETTER

DEPENDABILITY AND SECURITY BY ENHANCED RECONFIGURABILITY



>About the DESEREC Newsletter
DESEREC Newsletter is published by DESEREC, a research project partially funded by the European Commission under the 6th Framework Programme, http://www.deserec.eu

>Registration to this newsletter is available at
http://www.deserec.eu/newsletter.html

>Questions on the project should be sent to the coordinator:
http://www.deserec.eu/partners/partner_thc.html

>Editor
Marco Domenico Aime, POLITO, m.aime@polito.it

Table of contents

Selected links

DESEREC (Dependability and Security by Enhanced Reconfigurability)
http://www.deserec.eu

Related projects:
POSITIF (Policy-based Security Tools and Framework)
http://www.positif.org

ESFORS (European Security Forum for Web Services, Software and Systems)
http://www.esfors.org/

RESIST (Resilience for Survivability in IST)
http://www.resist-noe.org/

SERENITY (System Engineering for Security & Dependability)
http://www.serenity-project.org/

ENERGy (Empowered Network Management)
http://www.itea-energy.eu/

Information Society Technologies

Goals of the Newsletter

by Marco Aime,
Politecnico di Torino (POLITO)

This is the second issue of the DESEREC newsletter. This newsletter focuses on the achievements of DESEREC in order to assist the readability and visibility of the project's results. It has the following objectives:
- Inform on past and future DESEREC activities and events,
- Summarise ongoing activities,
- Provide links to detailed material on specific subjects,
- Foster liaison with other related projects,
- Disseminate and advertise the project's results,
- Announce significant events (e.g. conferences) in the field of dependability and security.

In this issue we:
- advertise new events related to the project,
- introduce the first prototype of the DESEREC architecture and of its design framework,
- and present a preliminary set of prototype tools as part of the DESEREC design framework.

Registration to the newsletter can be done online at: http://www.deserec.eu/newsletter.html. Each newsletter volume will then be announced by email to the registered email list. The DESEREC Newsletter is also published on the following web site: http://www.deserec.eu/.

Dissemination workshop

by Luca Durante,
Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni (IEIIT)

The 2nd ESFORS Workshop on "Trust, Security and Dependability in Service Oriented Infrastructures" took place in Maribor (Slovenia) on July 10th and 11th, 2007, in the premises of the local Faculty of Computer Science.
The workshop was organized by the Coordination Action ESFORS, with the cooperation of the European Commission DG INFSO unit F5, the European Technology Platform NESSI, the Slovenian Technology Platform NESSI, the Network of Excellence RESIST, and the Integrated Projects SERENITY and DESEREC.

Thanks to the valuable help of well known experts – such as Dr. Thomas Skordas (Project Officer, DG INFSO, EC), Prof. Antonio Lioy (Politecnico di Torino), Prof. Miroslaw Malek (Institute of Information, Humboldt-University of Berlin), Prof. Paulo Verissimo (University of Lisbon) and Dr. Gregory Chockler (IBM Haifa Research Laboratory) – researchers and managers, coming from academia and industry of several European countries and involved in European projects in the ICT field, found a good and valuable opportunity for exchanging and merging research experiences and drawing the lines for future activities.

The program of both days started with plenary sessions for introductory and keynote speeches, continued with the presentation of the European Projects officially represented and then went on with parallel sessions.
On the first day parallel sessions were dedicated to R&D gap analysis and future (long-term) research topics. These sessions were based on focused discussions to get common consensus around a selected set of topics, such as – but not limited to – security patterns, service composition and runtime issues, formal methods and specifications for design, development and testing of services, and presentations proposed by attendees and organizers.
On the second day parallel sessions were based on a "brainstorming" methodology, and instead of structured inputs, open contributions have driven the discussions, in particular focused on "Resilience in Services and Service Infrastructures".
Finally, a concluding plenary session has synthesized and merged the results coming from the previous sessions, designing future directions of research on Trust, Security and Dependability of service

oriented infrastructures, from their design to run-time management.

DESEREC has provided several contributions to the workshop organization and sessions thanks to Prof. Antonio Lioy (Politecnico di Torino, Italy), keynote speaker on "Some thoughts for future RTD in secure software systems and services", and to Dr. Luca Durante (IEIIT/CNR, Italy), member of the program committee and chair of session 1 of day 1 "Engineering dynamic & ad-hoc service coalitions: Design and operational (run-time) TSD aspects", and to other speakers from several Deserec partners.
More information is available at:
http://www.esfors.org.

Training workshop

by Sofoklis Efremidis, (ICOM)

Preparations for the Second Training Workshop of DESEREC are underway. The workshop site at http://www.intracom-telecom.com/DESEREC-2TW/. Registration to the event is open through the same link. The workshop will take place in the premises of Intracom Telecom in Peania, Greece, on September 24 and 25. The objective of the workshop is to raise in-project and public awareness on the issues targeted by DESEREC and present the mechanisms and technologies that are developed within the project for increasing the dependability of information systems. The approach taken by DESEREC to improve the dependability of critical information systems is to monitor them, make intelligent decisions, and react when incidents are detected by reconfiguring them. Two external speakers from related EU projects (ITEA ENERGY and IST HIDENETS) will present their projects' view on dependability aspects.

All sessions of the 2-day event will be videotaped for the purpose of creating along with the presentation slides a DVD with all workshop material.

DESEREC publications

-M.D. Aime, A. Atzeni, and P.C.Pomi, *AMBRA – Automated Model-Based Risk Analysis*, to appear in the proceedings of the 3rd ACM Workshop on Quality of Protection (QoP 2007).

- M. Woda, T. Walkowiak, *Multi agent event monitoring system*, in the 3rd International Conference on Information Technology ICIT 2007.

-L. Bagrij, K. Nowak, Method for Quality of Network Services Analysis Using Queuing Modelling of Information Systems and Computer Simulation Techniques, in the 3rd International Conference on Information Technology ICIT 2007.

-D. J. Martinez, M. Gil Perez, G. López, A. F. Gómez-Skarmeta, *A proposal for the definition of operational plans to provide dependability and security*, In the Critical Information Infrastructure Security CRITIS'07.

-A. Atzeni, and A. Lioy, *An estimation of attack surface to evaluate network (in)security*, in Proceedings of the 9th Int. Conference on Enterprise Information Systems.

-P. Krekora, and D. Caban, *Dependability analysis of reconfigurable information systems*, in Proc. of the 2nd Int. Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2007.

-K. Nowak, and L. Bagrij, *Using distributed multilevel agent-based monitoring technique for automated network modelling approach*, in Proc. of the 2nd Int. Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2007.

-M. Cheminod, I. Cibrario Bertolotti, L. Durante, R. Sisto, and A. Valenzano, *Evaluating the combined effect of vulnerabilities and faults on large distributed systems*, in Proc. of the 2nd Int. Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2007.

-P. Pérez, and B. Bruyère, *DESEREC: Dependability and Security by Enhanced Reconfigurability*, European CIIP Newsletter, Jan./Feb. 2007, Volume 3, Number 1.

-M. Cheminod, I. Cibrario Bertolotti, L. Durante, R. Sisto, and A. Valenzano, *Experimental comparison of automatic tools for the formal analysis of cryptographic protocols*, in Proc. of the 2nd Int. Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2007.

-D. J. Martínez-Manzano, G. López Millán, and A. F. Gómez-Skarmeta, *Multidomain Virtual Security Negotiation over the Session Initiation Protocol (SIP)*, in Proc. of the 1st International Workshop on Critical Information Infrastructures Security, CRITIS 2006, LNCS 4347-0249.

-M. Sánchez, G. López, O. Cánovas, J. A. Sánchez, A. F. Gómez-Skarmeta, *Un sistema de control de acceso para la distribución de contenidos multimedia*, in Proc. of the 9th Reunión Espanola sobre Criptología y Seguridad de la Información, RECSI 2006.

The DESEREC architecture

by Benoit Bruyère,
THALES Communications (THC)

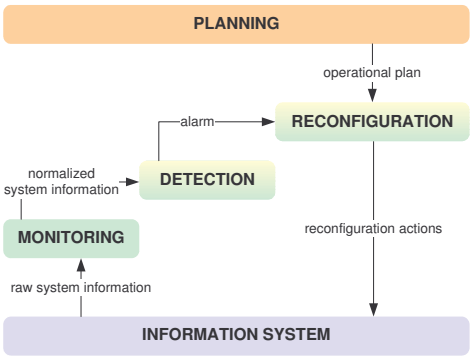
Project approach

The DESEREC project is looking for increasing the dependability of existing and new networked mission-critical Communication and Information Systems (CIS) from a Business Services (BS) point of view. Automated and semi-automated reconfiguration is the DESEREC lever to achieve this goal.

The DESEREC approach is to implement a three-tiered framework:

1. Planning of optimal configuration for anticipated operational modes on central level
2. Human controlled reconfiguration with priority to critical business services on global level
3. Automated Incident detection and quick containment on local level

From a time response perspective, the three layers operate in two modes:
- Tier 1 is responsible for planning, modelling, evaluating CIS configurations and possible reconfiguration scenarios; it runs on-demand with loose time constraints.
- Tiers 2 and 3 are online tools performing detection, decision and reaction functionalities at both local and global levels.



In the present paper, we focus on the online part of the DESEREC architecture as the planning part is detailed in other articles present in this newsletter.

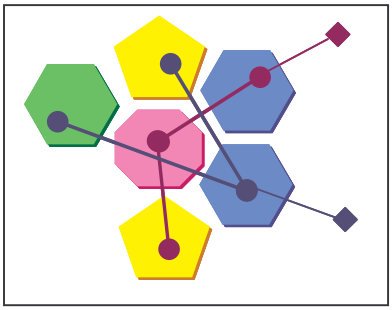
The molecule concept

In order to master such multi-level reconfiguration loop, DESEREC introduces the concept of *molecule*: this approach aims at providing high dependability with a few (down to nil) spare resources provisioned, common to all the services. The *molecule* is designed as an encapsulated set of devices with some dependability properties: it is self-observable, i.e. complex enough to provide external "observers" with a reliable status on its own dependability and performance. The molecule boundaries define the area where the local level of reaction applies.
The molecular approach is self-manageable, smart enough to translate high-level management requests received through its interfaces into detailed actions on its components. The CIS is designed from a small set of molecule prototypes, instantiated as needed to host the various applications providing the services.

Business services

As commonly accepted, we named "*Business Services*" the various perceptions of the end user at his/her interface to the part of the CIS supporting a Business Process.
Within DESEREC, a Business Process is an *orchestration* of technical processes, the overall *choreography* being limited to the management of dependability and performance commitments agreed in the *Business Services Level Agreement* (B-SLA).
As an example, "TV over ADSL" service provides *Business Services* such as: Registration and Subscription management, TV on demand (Pay-per-view subscription and management), Broadcasted channels (Browsing the program guide, channel selection).

A *Technical Service* is an operational activity realized by the information system, representing a technical step in the execution process of a *Business Service*. Each instantiated molecule hosts one or many *Technical Services*. The orchestration of *Technical Services* to provide a *Business Service* is built by activating these services on the relevant molecules and configuring them properly.
Multiple classes of molecules may exist within the CIS. The more instances of the same molecule class exist within the CIS, the more flexible is the system in terms of reconfiguration. The following diagram illustrate the symbolism used for molecules (having different shape/colour to show classes), technical services (dot within molecule) and business services (square dot at the end of the collaboration chain of technical services).



This set of molecules is then under the supervision of a Business Service manager that deals with maintaining the performance over the B-SLA commitments. The coarse grain organisation provided by the molecular approach and the object-like instantiation pave the way to manageable containment and reconfiguration features.

Agent-based architecture

The figure below shows how agents are implemented into a distributed multi-agent architecture.

At local level

- **DItemAgent:** agent running on target is composed of the two elements DSensor and DProxy. Hence DItemAgent serves as a container for the various DSensors and DProxys of a defined context
  - **Dsensor**: passive element that monitors the target element, collects events of interest, and translates them into DESEREC normalized event format.
  - **Dproxy** (also called effector): passive element that executes the DLocalAgent's orders to control the target element of the underlying CIS. The DProxy translates the commands and configurations from DESEREC format into raw, target specific actions / commands (e.g. SNMP SET) and configuration files (e.g. httpd.conf).
- **DLocalAgent:** analyses the local level situation and triggers appropriate self-healing reaction when needed. Therefore, a DLocalAgent is an active element that is able to detect known critical situations (incidents), take decisions on its own and in turn is able to autonomously perform self-healing reaction deploying pre-defined set of actions and re-configuration patterns to its molecule's DitemAgents

At global level

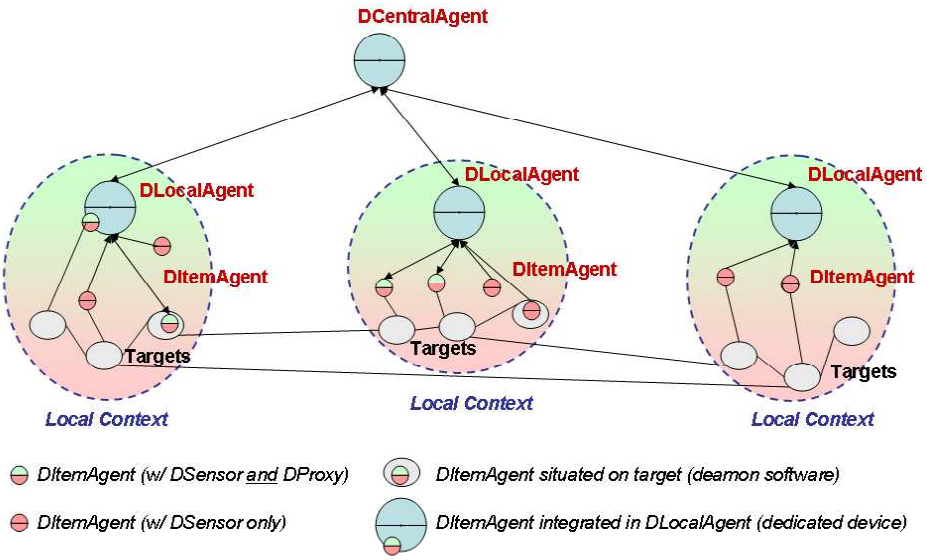
- **DCentralAgent**: analyses the global situation of the CIS and its services. It is responsible for monitoring and controlling the overall CIS with the help of the DLocalAgents. Therefore, a DCentralAgent is an active element that is able to detect distributed threats and incidents based on pre-configured detection scenarios. It takes decisions either through user intervention or autonomously and it then deploys reactions via the DLocalAgents.

Conclusions

As seen, this molecular approach provides multiple advantages to the DESEREC framework:

- Allow the multi-level reaction mechanisms,
- Define the scope of local detection and reaction,
- Simplify the reconfiguration of resources at the global level.

Although this molecular approach was devised with dependability in mind, additional benefits are even anticipated in terms of proactive maintenance and provisioning.



The DESEREC design framework

by Marco Aime,
Politecnico di Torino (POLITO)

The design framework lets the DESEREC user to
- describe its system and requirements,
- input internal and external constraints,
- generate possible configurations for the system,
- and rank them based on a set of security and dependability metrics.

Its main components are
- a modelling framework, i.e. a set of common meta-models and tools that help to build and manage formal descriptions of the target system and its behaviour,
- an analysis framework, i.e. a set of techniques and tools to evaluate the security and dependability properties of the different configurations of the target system,
- a planning framework, i.e. a set of tools to generate configurations satisfying the user requirements, complying to his constraints, and achieving predictable security and dependability properties. The generated configurations are validated using the analysis framework in order to build a set of predefined selected configurations.

The selected configurations are then pushed to the DESEREC runtime environment for deployment on the target system. Moreover, the design framework let the user to model a set of reconfiguration rules, i.e. events-action pairs that specify when and how the current enforced configuration should be changed in reaction to adverse conditions.

A central role in the design framework is played by models and meta-models: they are in fact at the basis of the integration between the different tools in the design framework, and the primary interface with the runtime framework.
The modelling framework that we have designed so far not only lets us describe the system taking into consideration business layer as well platform characteristics, structural as behavioural information, system-wide security policies as well specific service configuration. It also lets us integrate different analysis techniques and tools based on emulation, simulation, and formal analysis in order to compute a set of high-level measures fit for decision making, both at design time for selecting best configurations, and at run time for triggering changes to the current configuration.
To make this possible we feed our analysis tools with both the description of the system and its configuration and a set of additional models that help us analyse the behaviour of the system in front of adverse conditions. For the moment, these analysis models include:
- a vulnerability model that describes the ways an attacker can maliciously interact with the system,
- a fault model that describes what adverse natural events may occur in the system and their predictable effects,
- load and resource consumption models that let us assess the capacity and QoS achievable by the system.

Sections in the following pages discuss in further details the modelling framework and tools, and some of the analysis tools we have prototyped so far.

Modelling tools

by Marco Aime,
Politecnico di Torino (Polito)

We model the system and its configuration following a layered approach:

- the requirements layer lets us model the business services the target system should provide, and the system infrastructure, that is the hardware and software resources actually available to provide the services,

- the policy layer lets us model constraints on how services should be deployed on the system infrastructure while complying company user internal/external rules on service provisioning, security, high availability, etc…,

- the configuration layer model a specific system configuration including service deployment and configuration of every involvement element so that all the constraints are possibly satisfied.

The configuration layer is based on the DMTF's CIM model (www.dmtf.org) in a XML representation and describes abstract element configurations: this configuration is then translated into a specific element configuration within the DESEREC runtime framework.

For the policy layer we have defined an ad-hoc XML-based language, the SCL (Service Constraints Language). We distinguish between high-level policies and service specific configurations. High-level policies have a system-wide scope: e.g. security policies (e.g. data security, service authorisation policies) must be addressed from a global perspective. While more specific rules are needed to model how a public service should be published to the rest of the world, and how interactions between service components should take place.

The system infrastructure is described with an extension of the SDL (System Description Language) developed within the POSITIF project (www.positif.org). SDL lets use model both network elements (such as networks, hosts, hardware platforms), logical ones (that is running software, technical services), and their interconnection. SDL has an XML-based representation.

We use the W3C's WS-CDL (Web Service Choreography Description Language) for the business service modelling (www.w3c.org). Though originally built for describing web-service based architecture it may be used to model generic services (e.g. web based). WS-CDL follows a workflow service description paradigm: we favoured this approach over more abstract ones as it better supports data security analysis and is closer to engineering practice. Also for WS-CDL we use an XML-based representation.

Associated to the above models we provide a set of modelling tools that

- help the DESEREC user to visually build, inspect, and update models,

- let validate a system description against the relevant meta-models,

- detect inconsistencies within a single model and across different models (e.g. between resource description and service configuration),

- provide simplified interfaces for accessing information included in the models to help the integration and of planning and analysis tools.

Formal analysis tools

by Luca Durante,
Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni (IEIIT)

The formal analysis tool deals with the exhaustive analysis of potential vulnerabilities and attacks exploitable by a malicious user. This kind of analysis requires the knowledge of the existing vulnerabilities and the ability of modelling such information in a machine readable format.
More precisely, a vulnerability must be defined in terms of pre-conditions and post-conditions. That is: the necessary conditions that make a vulnerability exploitable and its effects on the system and on the attacker knowledge.

DESEREC has thus reviewed a number of existing vulnerabilities databases, but none of them provides all the necessary information. In particular, there is a lack in a formal definition of what are the effects of a vulnerability.
This is a rather important feature in order to conduct a formal analysis on the system. A textual description of a vulnerability is useless other than to study a single node in a network. To collect the effects of vulnerabilities on the whole system it is mandatory to have a precise description of the effects of every vulnerability. Moreover, an exploited vulnerability can change the state of the system in such a way that another vulnerability becomes

exploitable as well. This is the idea behind the study of chains of vulnerabilities.

For this reason, starting from the existing OVAL project (http://oval.mitre.org), an extended meta-model has been defined, precisely implementing the post-conditions of vulnerabilities.

The actual meta-model uses a rather detailed description of the analysed hosts. Such fine grained description is not always available in the other DESEREC models. Nonetheless the formal analysis tool can make some conservative assumptions in order to cope with the potential lack of details.

The exploitation of a single vulnerability on a single network's node can be of minor impact. On the contrary exploitation of multiple vulnerabilities one after the other, on more than one host, can lead to disruptive consequences.
The vulnerability meta-model has thus been designed to easily allow the creation of chains of vulnerabilities by mapping post-conditions on pre-conditions. The precise mapping is done, internally, by the tool.

The attack analyser uses both the vulnerability model and the resource model (as well as the service model and policy model). The entire system is internally modelled in prolog language, every element of the system (physical hosts, services,…) is represented by a "node" of some "class". Every node is then associated with some *facts*. The attacker itself is a node within the system and, from there, the analyser will check which vulnerabilities are actually exploitable by the intruder. If the set of such vulnerabilities is not empty, this will result in an increased attacker knowledge as well as some state changes in the system. Indeed, exploited vulnerabilities have effects on the affected nodes. These effects will be translated in new facts associated to the nodes. It is worth noting that the tool uses the monotonicity assumption, that is, every asserted fact will be never retracted. This is necessary in order to deal with the scalability issue in analysing large networks.
As long as new exploitable vulnerabilities are found, the tool will repeat the analysis. Eventually, the *maximum* set of exploitable vulnerabilities and of compromised hosts will be found.

From this result the tool can then build the attack graph, explicitly listing all possible attack paths that will bring the system in an undesired state.

Each node of this graph contains information on the host that has been compromised by the attacker and on the set of vulnerabilities the attacker has exploited. The attack graph analyser gets information from both the attack graph and the vulnerability model, allowing a deep analysis of which steps the attacker has to perform in order to put the system in an undesired state. Moreover, a subset of the attack graph can be extracted. For instance, in order to analyse all the possible ways an attacker has to compromise a particular host, the related attack tree can be extracted. Although the tool at the moment focuses on vulnerability analysis, it is nonetheless able to perform a (simple) dependability analysis.

Simulation tools

by Dariusz Caban,
Wroclaw University of Technology (PWR)

Simulation is used in the DESEREC framework to rank the various possible system configurations and reconfigurations (operational planning), to determine observable metrics (performance and dependability metrics). Simulation is able to distinguish proper operation of the CIS from a degraded one (detection scenarios), to predict the impact of security and dependability incidents on the overall system performance (quality of service).

The project aims at increasing the dependability of networked mission-critical Communication and Information Systems (CIS) with a Business Services point of view. Thus, the simulation needs to analyze the structure and interdependencies of the running services, not just the network configuration and protocols. This can not be achieved using most of the off-the-shelf network simulators, commercial or open source. A partial exception is the ACE tool provided with the OPNET simulation framework (as discussed in the next article).

DESEREC simulation tools extend the capabilities of network simulators by providing behavioural models capable of interpreting the business services orchestration, provided by WS-CDL like descriptions (www.w3.org). A number of open source simulators were considered as the basis for this development. Two were found particularly interesting: Omnet++ (www.omnetpp.org) and SSFNet (www.ssfnet.org, prime.mines.edu). The last one was chosen, as it promised simpler integration with the DESEREC system models through the DML modelling support. SSFNet was extended to support

the concepts of business services orchestration, dependability and vulnerability.

It is generally accepted that system analysis, based on simulation, is only as good as the inputs it is exposed to. In DESEREC this is represented by a set of system use-cases, that is the distinct usage scenarios that have to be analyzed. These encompass both the proper scenarios, potential exploits or misuse, attack sequences (derived from formal analysis) and failures. The load model describes these use-cases, their rate of occurrence, timing and duration. It also covers traditional problems of network loading by background traffic.

The load model plays an important role in accumulating knowledge about potential problems in system management. Any risk, identified during analysis or while running the system (self-learning), may be recorded as a load model update, so all subsequent modifications are tested against it.

The main problem with simulation of networked business services stems from the fact that it is not sufficient to model the behaviour of the network, its hardware and protocols. It is also necessary to predict the behaviour of the application specific software, which supports these services. As mentioned these are modelled with an orchestration/choreography language. However a further problem is predicting the performance of the modelled services, that is assessing the request processing time, the number of requests that are processed in parallel, the saturation point (when requests are dropped). This is addressed by the resource consumption model, which tries to correlate the software performance with available processing time, memory (and possibly disk access bandwidth). This model is still in its early stage of development, especially when one considers sharing of resources between services.

An alternate solution, also considered in the project, is to emulate (or simulate with a very high detail) the system hardware: computer hosts and networks. Then, all the software (operating systems, technical services and custom business applications) can be installed and run in this emulated environment. The approach is very attractive, since it avoids the problem of insufficient data about the software performance. Unfortunately it has a number of drawbacks, foremost among them inadequate scalability and limited handling of faults. For this reason, the DESEREC simulation tools utilize this approach indirectly. The resource consumption models are identified via emulation on a fragment of the system. It is then used for simulating the whole system. For emulation capabilities, DESEREC has so far experimented with SIMICS (www.virtutech.com).

The complete picture of the simulation framework also includes a tool for statistical analysis of the data obtained from simulation. Based on an open source engine, the tool is used to derive high-level dependability and performance metrics, as well as to extract facts pertinent to build dependency graphs and incident detection scenarios.

Simulation of Business Applications

by Bert Boltjes, (TNO)

High Fidelity (HF) simulation models of IT infrastructure elements and applications for discrete event simulation can be combined to gain insight in network properties and performance. With network simulation one can address issues such as:

- Efficiency. What is the load and utilization of connections in the network? What is the chance of congestion?
- Flexibility. What happens when the network topology changes, for instance when new links are introduced or when a link fails?
- Services. Can the network support the new services I want to offer?
- Availability and dependability. How many clients can my network satisfy in the future? How often will a service request fail?

To address those issues via HF simulation, a certain number of steps are required.
First, valid models must be implemented or acquired of the elements and applications in the network.
Next, scenarios and network topologies must be proposed. The scenarios contain the traffic requirements of the clients in the network and the value of model parameters of elements in the network (link speed, QoS, BER values, et cetera). Link failure and recovery can also be part of a scenario.
After creating the network model and the scenarios, simulations can be performed during which statistics on the performance of the network are collected.
After analysing the results, statements can be made about the properties and performance of the (proposed) network. Predictions can be made on when

the network will start to perform unsatisfactory. Also, 'rules of thumb' can be derived.
Experience over the years it has shown that it is quite hard to obtain network simulation models whose performance (delays, throughput, packet loss, etc.) are compared in sufficient detail with the entities in real life they are supposed to represent. Although implemented features such as protocols and algorithms are implemented meticulously, performance validation is often lacking.

One category of simulation models which are difficult to obtain are accurate models of specialized business applications. Models of these applications are of interest within DESEREC in the use cases that are being modelled. TNO has experimented OPNET modelling environment for this. The OPNET environment includes the Application Characterization Environment (ACE) with which data transactions over the network can be analyzed and modelled efficiently and accurately. For this OPNET uses traffic capture daemons which can be distributed throughout the network and orchestrated from a central console. After the data transaction has ended the central console collects and filters, correlates the collected traffic and shows a clear insight in the traffic between the tiers in the network. These data transactions can than be simulated in a network model for more than one user.
The same techniques and workflow may be used in DESEREC to give inspiration and statistical estimates on how to model the various business applications without having to analyze the internals of them. Open Source and free network simulation environments such as SSFNet can benefit from this to model specific applications.