Taming Monsters like Daidalos
UML combined with ITU-T languages in large Telecommunication Projects
ISSRE04: WITUL Workshop 2004

Yongzheng Liang, Jürgen Jähnert, Paul Christ . University of Stuttgart, Stuttgart
Telemaco Melia, Amardeo Sarma NEC Network Laboratories Heidelberg, Germany

Abstract

Modern telecommunication systems are showing an increasing degree of complexity. This is especially true for projects actively engaged in the process of convergence between Internet-type of data communications, telecommunications and broadcast – often including several forms of mobility. While it is intuitively obvious that such complex systems and projects will probably benefit from formal modeling, testing and simulation techniques – actually such benefits may not be easy to harvest.

Taking-off from "Daidalos", a large European Telecommunications project and its present usage of Telelogic's Tau 2.3, the paper is going to propose the development of a "complete UML/SDL-based tool chain" for Daidalos. Among the issues dealt with, two contributions concerning testing and reliability are proposed.

1 Introduction

Modern telecommunication systems show an increasing degree of complexity. This is especially true for projects actively engaged in the process of convergence between Internet-type of data communications, telecommunications and broadcast – often including several forms of mobility. While it is obvious that such complex systems and projects will probably benefit from formal modeling and simulation techniques – actually such benefits may not be easy to harvest.

The rest of the paper is organized as follows: In Chapter 2 we introduce Daidalos, one of the "monster" European FP6 "Integrated Projects" (IP). Tackling almost all of the issues with respect to the convergence process between Internet-type of data communications, telecommunications and broadcast, Daidalos also addresses the many forms of mobility related to such converged communication scenarios. In Chapter 3 we are going to demonstrate examples of the present use of "UML and SDL combined" within Daidalos. The initial success of these modeling activities then motivates the idea of a "complete UML/SDL-based Daidalos tool chain". Accordingly, in Chapter 4 we are going to relate the present usage of Tau, i.e. a tool based on UML/SDL, in Daidalos to the present and evolving situation in the related bodies such as the ITU, ETSI and the OMG. We propose elements of the envisioned Daidalos tool in fields of concern such as "communications", testing and reliability. Finally we provide a summary.

2 Daidalos

Comprising close to fifty partners from industry, operator side and Academia, the IST project Daidalos, [1], is one of the "monster" European FP6 "Integrated Projects" (IP). Tackling almost all of the issues with respect to the convergence process between Internet-type of data communications, telecommunications and broadcast, Daidalos also addresses the many forms of mobility related to such converged communication scenarios. The overall "vision" of Daidalos is a world where …

- Mobile users can enjoy a diverse range of personalized services – seamlessly supported by the underlying technology and transparently provided through a pervasive interface

- Mobility has been fully established through open, scalable and seamless integration of a complementary range of heterogeneous network technologies

- Network and service operators are able to develop new business activities and provide profitable services in such an integrated mobile world

In its overall realization strategy, Daidalos pursues a scenario based approach – providing a “Mobile University” and an “Automotive” scenario. Towards this end, corresponding storyboards of the scenarios were initially written and then divided into a series of numbered “atomic” scenes.

Technically, Daidalos aims at the construction of a “super-layer” architecture according to the following Figure 1 below.

Workpackage	Super-Layer	Elements	Attributes
5	Testbeds/Pervasive Applications	Location aware applications Web Services	Multi-provider scenarios
4	Pervasive Services Support Platform	Personalization, Rules, Security, Privacy, Context, Events Service Management	Multi-provider scenarios
3	Service Provisioning Platform	Content Adaptation framework Service discovery and Composition Policy-based Management, Mobility Management, AAA, QoS Brokerage and Management, Security	Multi-technology/ multi-provider scenarios

2	Access Networks	Mobile-IPv6 based abstraction layer for Mobility Management, QoS, security etc. on top of: Ethernet, WLAN, WiMAX, TD-CDMA, Bluetooth, DVB-H	Multi-technology/ provider scenarios;
			Unicast/ multicast
			Fixed/wireless;
			Infrastructure/ ad hoc

Figure 1: Daidalos ‘Super Layer’ Structure and Workpackages

Given this context, Daidalos is probably a very typical example of “a changing world” for SDL, [2].

Contemplating in such a situation on the introduction of formal techniques into the project, one has to take into account, that Daidalos layer borders are often also “cultural” borders: In the access network layer, one typically will find kernel hackers modifying or improving Linux, even RT-Linux modules. To them, ‘UML’ has no relation to modeling but expands simply into “user mode Linux”. The Service Provisioning Platform is dominated by typical IETF-protocol-style and Telecom guys. Finally, the Pervasive Services support layer is dealt with mainly by computer science people and – by chance - partly with people from the TMF world with a certain inclination towards Model Driven techniques.

In summary, in Daidalos it is considered as an initial success that all workpackages agreed to support a common formal approach to the system engineering process. A set of typical scenes was selected from

the two scenarios and for each selected scene a detailed document comprising related pre- and post-conditions and a high-level use case diagram was produced. In addition, so-called vertical teams studied the interactions between the Daidalos “super-layers”. It was also agreed, that a global Daidalos model would be produced. As “Daidalos modeling tool”, Tau – at present version 2.3 – was contracted with Telelogic.
The given description of Daidalos clearly motivates the strategic interest of the project in the usage of modeling tool of the category of Telelogic’s Tau - namely

- to validate critical components, especially related to protocols
- to solve the complexity problem when several functions are to be integrated and to cope with the resulting feature interaction problem
- to separate concerns and to enforce the elaboration of well-defined interfaces in the “monster” project as basis for the integration effort
- to foster the project’s internal integration via a common language framework.

3 Present usage of UML and SDL in Daidalos

In the following we describe the present state of the usage of UML and SDL in Daidalos.

The overall Daidalos architecture, showing the functional building blocks in accordance with Figure 1 above, is depicted in the following
. This figure shows especially the federation between different service providers at different ‘super layers’.

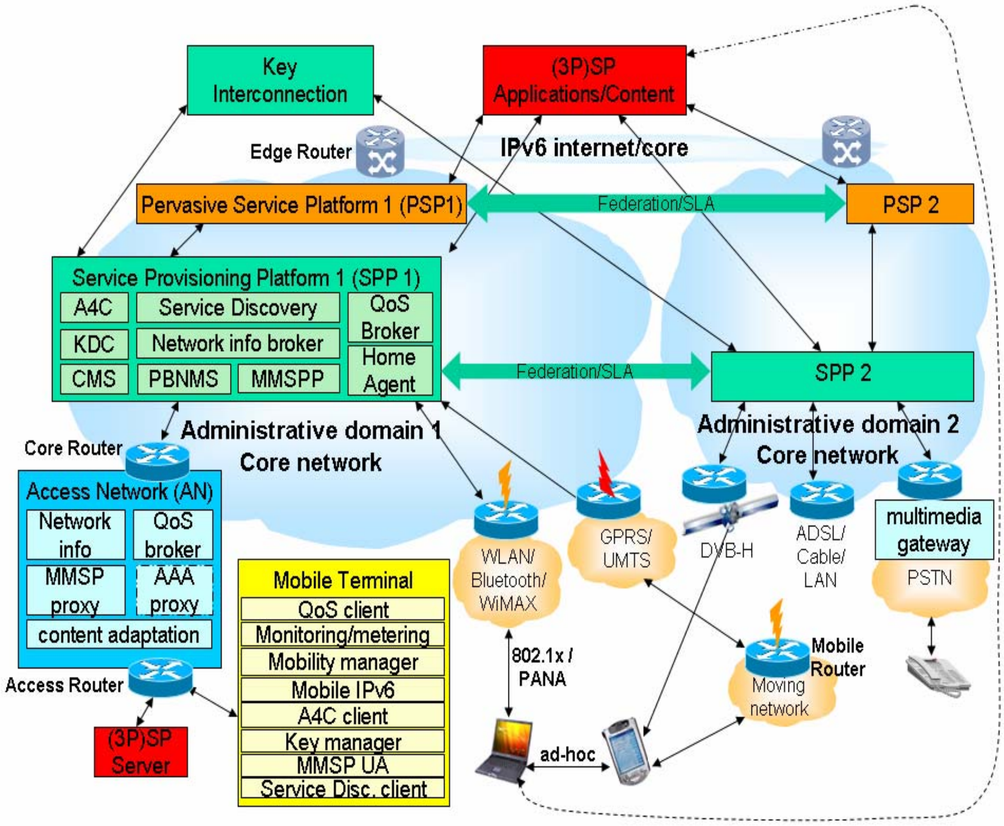


Figure 2: Overall Daidalos Architecture

At the overall architectural level, parties from the different Workpackages have contributed mainly UML-like detailed sequence diagrams in MS ppt and visio, Rational and EA. This input has been manually converted to Telelogic Tau G2 UML.
On the other hand, there are also high-level 'architecture diagrams' – mainly in MS ppt.

A typical conversion of such an architecture diagram from ppt to UML for the Service Provisioning Platform is shown in the following Figure 3 and

Figure 4.

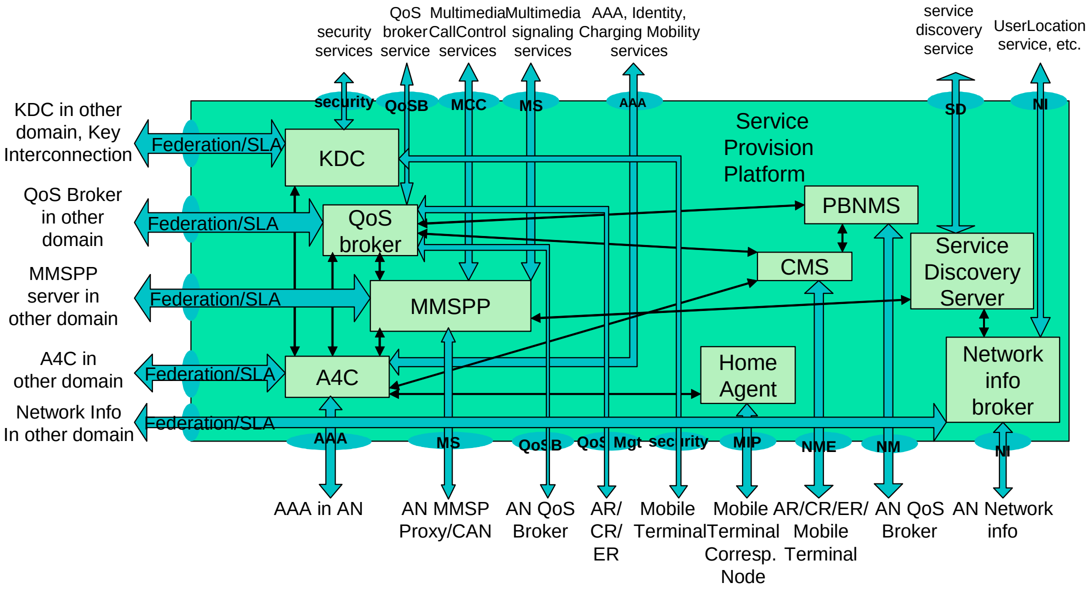


Figure 3: Daidalos Service Provisioning Platform

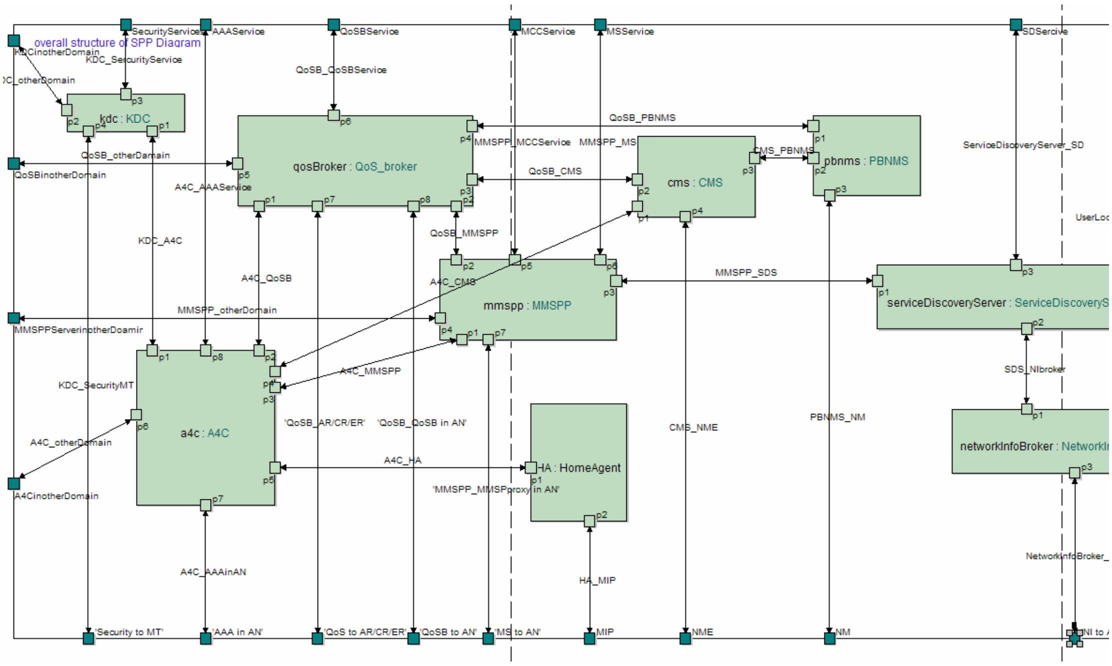


Figure 4: Daidalos Service Provisioning Platform initial architecture diagram

The next step, carried out at present, is the alignment of all the existing related sequence diagrams with this architecture diagram in a consistent manner.

This cumbersome process, carried out at three organizational levels – Workpackage – Activity – Task - has revealed and still is revealing many inconsistencies between the models of the different working parties in Daidalos.

While at the overall architectural level the modeling process in Daidalos so far is restricted to the construction of UML models, in the Mobility Management field of Workpackage 2, Tau’s SDL was used from the very beginning. The following Figure 5 is showing a typical ‘Manhattan Grid’ where a Mobile Terminal would move from access router to access router and between provider domains. Comprising in addition different access technologies, a full blown scenario would be three-dimensional – a ‘Manhattan cube’.

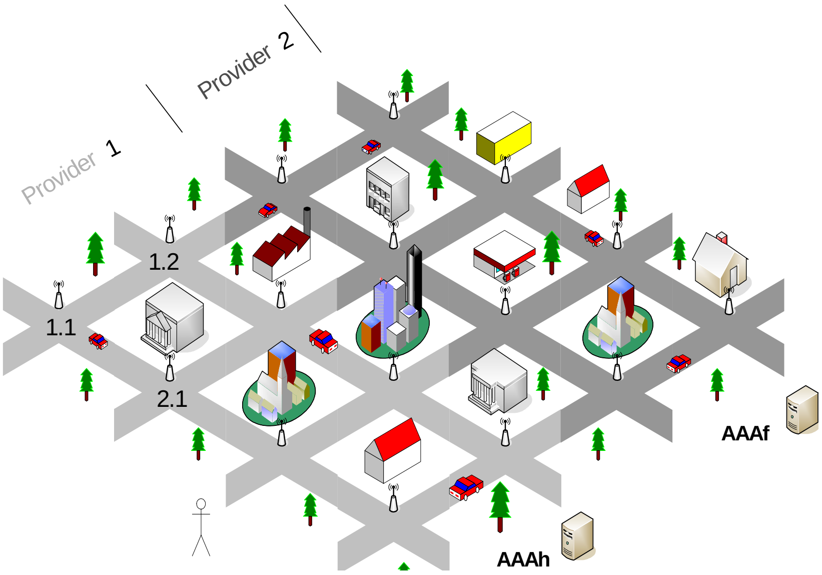


Figure 5: A Daidalos 2d 'Manhattan Grid'

The following Figure 6, Figure 7,
Figure 8 and Figure 9 are showing the corresponding UML architecture diagrams for one of the providers, the SDL state chart diagram for an access router, the mobile terminal and the execution by the Tau Model Verifier. The random movement of the terminal is 'caused' by an external C++ program.

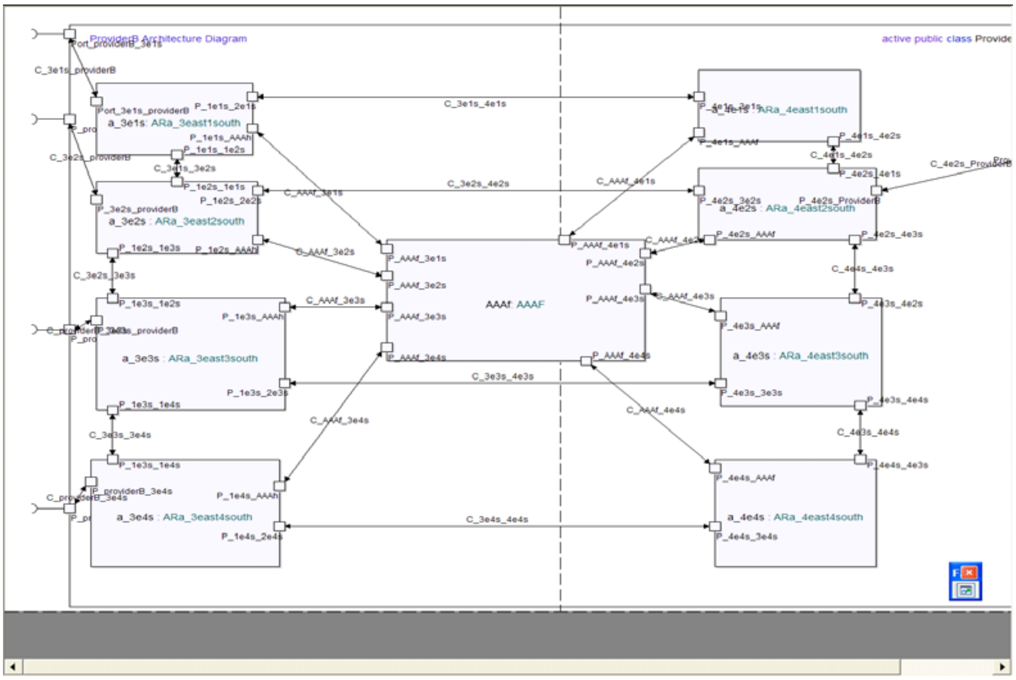
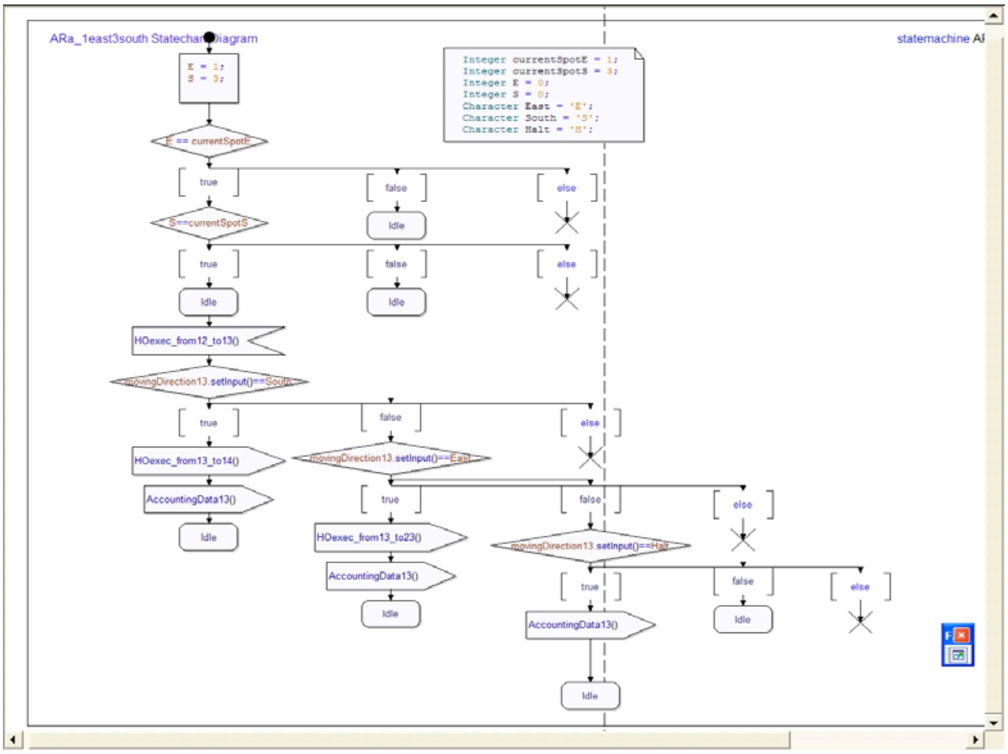


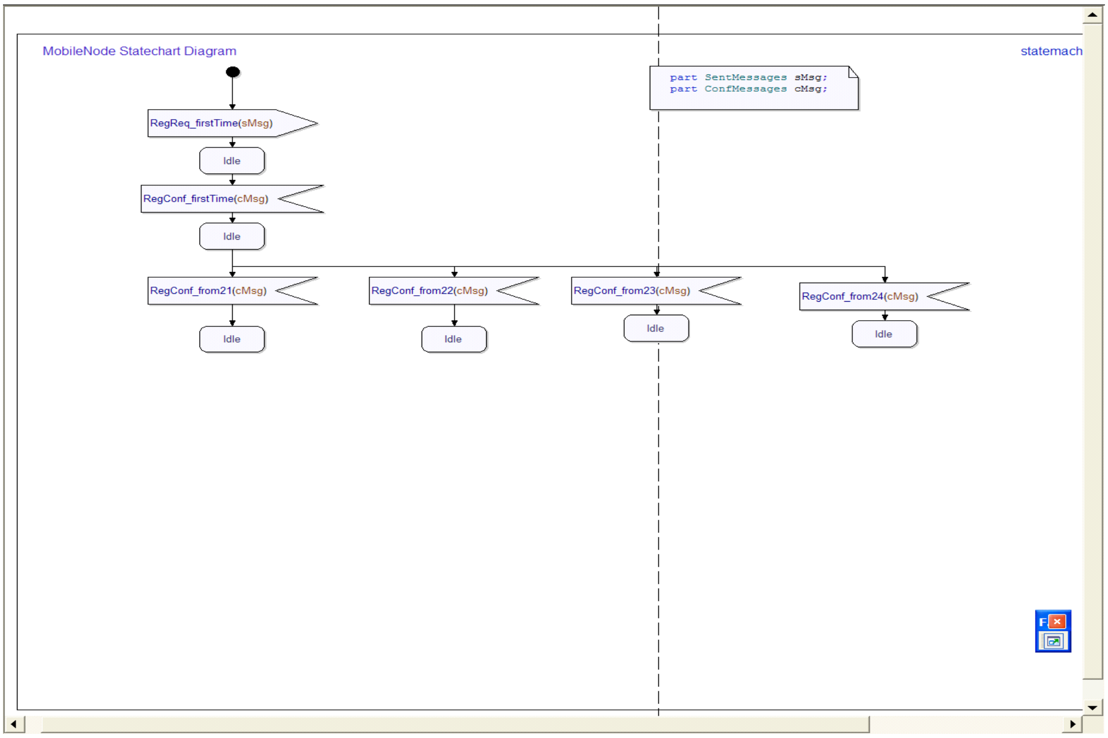
Figure 6: Daidalos Manhattan Grid architecture diagram



```
E = 1;
S = 3;
```

```
Integer currentSpotE = 1;
Integer currentSpotS = 3;
Integer E = 0;
Integer S = 0;
Character East = 'E';
Character South = 'S';
Character Halt = 'H';
```

Figure 7: Manhattan Access Router



```
part SentMessages sMsg;
part ConfMessages cMsg;
```

Figure 8: Mobile Terminal for Manhattan Grid

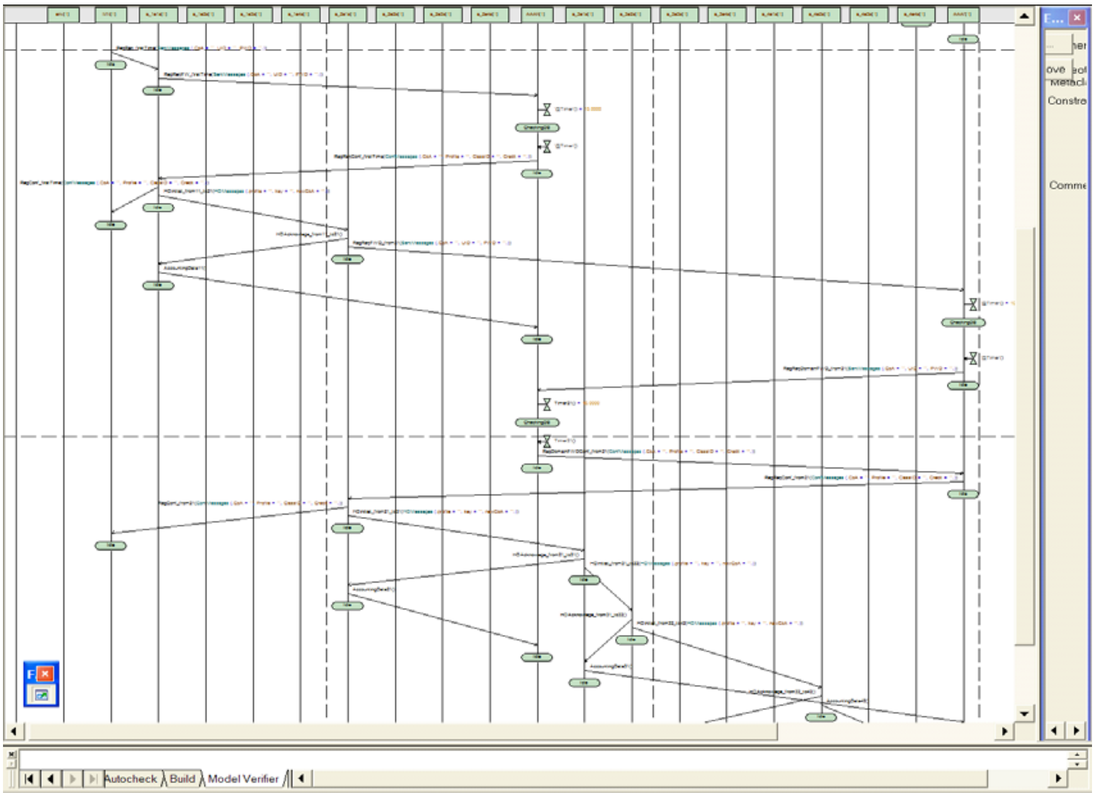


Figure 9: Model Verification of the Manhattan Grid

The initial effort and preliminary results of modeling in Daidalos have to be judged with respect to the overall project life cycle: The current phase 1 of Daidalos requires initial implementations and demonstrations already after year 1, i.e. end of October 2004. Following year 2, the project has to enter a 6 month demonstration phase. Therefore, Daidalos has to maintain a delicate balance between modeling and implementation work.
On the other hand, as mentioned above, already the initial import of all disparate modeling work into the unified UML/SDL framework of Telelogic's Tau G2 has demonstrated the necessity und usefulness of a modeling approach.

4 UML and ITU-T Languages – towards an UML/SDL-based tool chain for Daidalos

In this chapter we are going to relate the usage of Tau, i.e. a combined usage of UML and SDL in Daidalos to the present and evolving situation in the related bodies such as the ITU, ETSI and the OMG. We try to propose an extension of this "UML/SDL framework" towards a complete tool chain both sufficiently powerful and still of acceptable complexity for the average Daidalos team member.

We firstly reemphasize the high practical utility of the model verification artifact of the combined UML/SDL tool in Daidalos. Accordingly, we propose to support the work in ITU-T, ETSI and OMG concerning the "common use of UML and SDL".

To start with, the corresponding initial ITU-T recommendation in the given context, Z.109 will be superseded by the work of "Question 17" of ITU-T SG 17. Within ETSI, this work is supposed to be supported by the effort of the Specialist Task Force 250: UML profile for Communicating Systems. At present, the publication of the corresponding Work Item 'DTS/MTS-00085' is scheduled for 2005-04-27. While this work item suffered some delay, in its "scope" chapter an initial draft, [3] - clearly in the spirit of Daidalos - describes as one of its important goals "to make UML smaller and easier to use"

The next item in the envisioned UML/SDL-based tool chain might be "testing". The UML profile in question, described in the UML 2.0 Testing Profile Specification, [4], is supposed to be 'profilable' to various platforms and application domains. "The UML Testing Profile can be used stand alone for the handling of test artifacts or in an integrated manner with UML for a handling of system and test artifacts together". While some recent contributions see this profile in a potential cooperation with TTCN3, [5], we propose here to study the possible direct exploitation of UML TP in cooperation with UML/SDL system artifacts.

Concerning domain-specific profiles in the Daidalos context, basically all Web service and service-architecture related profiles are of interest and should be studied from a "UML/SDL-based tool chain" point of view.

Approaching non-functional aspects such as dependability, QoS and Performance in the context of Daidalos, we propose to study the utility of related UML profiles such as UML RT and the UML Profile for QoS and Fault Tolerance and profiles such as EAI and EDOC.
Concerning especially reliability, it is our understanding that up to now there were mainly two directions of work, both of which could be called proactive: The first line of work tries to 'instrument' UML models with tagged values and constraints and eventually to feed classical reliability models via XMI. The second line of thought – also proactive – follows a model driven approach towards reliability – often in a domain specific way, see e.g. [7]. While both approaches are of interest, Daidalos' first priority is clearly on correct functionality. However, given the system characteristics of Daidalos, we are proposing here a reactive approach to reliability in the spirit of the old paradigm of reflection and the more recent one of autonomic computing, [8], [10]: As systems will always fail one has to prepare for detection and healing of errors – an idea which is obviously not new and applied in many engineering disciplines. However, from a modeling point of view this results in a "causal connection of the real system to its meta-model", [9].

Finally, among the issues not mentioned so far are semantics in general and the notion of time. Here we propose to take into account the results of recent and ongoing IST projects such as OMEGA.

5 Summary

In the context of the large IST project Daidalos we have proposed to develop a "Daidalos tool chain" along the line of the historically successful combined usage of UML and SDL. As contributions we have suggested in the field of testing a simple approach of combining the

UML TP directly with UML/SDL system artifacts. And, given the nature of Daidalos-type of systems, we have put forward a meta-modeling approach towards reliability following the reflection paradigm.

6 References

[1] www.ist-daidalos.org

[2] E. Sherratt: SDL in a changing world. SAM'04, Ottawa 2nd June to 4th June, 2004

[3] ETSI: Draft ETSI TS 1XX XXX V<0.0.1> (2003-10) UML Profile for Communicating Systems

[4] OMG: UML 2.0 Testing Profile Specification. OMG Final Adopted Specification, 2003 – 2004

[5] I. Schieferdecker, Z.R. Dai, J. Grabowski, A. Rennoch. The UML 2.0 TestingProfile and its Relation to TTCN-3. Testing of Communicating Systems (Editors:D. Hogrefe, A. Wiles). Proc. of the 15th IFIP Intern. Conf. on Testing of Communicating Systems (TestCom2003), LNCS 2644, Springer, May 2003, pp. 79-94.

[6] http://www.uml.org/#UMLProfiles

[7] Nunes Rodrigues G.: A Model Driven Approach for Software Systems Reliability. CS University College London, 2002

[8] Gordon S. Blair, Hector Duran-Limon, Geoff Coulson, Paul Grace, Nikos Parlavantzas, Lynne Blair, Rui Moreira: Reflection, Self-Awareness and Self-Healing; in: ACM SIGSOFT Workshop on Self-Healing Systems (WOSS'02) November 18-19, 2002 Charleston, SC

[9] Marc-Olivier Killijian and Jean-Charles Fabre: Adaptive Fault Tolerant Systems: Reflective Design and Validation. 2002 (via google)

[10] Pattie Maes: Concepts and experiments in computational reflection. Conference on Object Oriented Programming Systems Languages and Applications. Orlando, Florida, United States. 1987