Large scale interoperability, integrating the Daidalos project

Miguel PONCE DE LEON[1], Frances CLEARY[1], Marta GARCÍA MORENO[2], Ana SOBRINO JULAR[2], Antonio ROMERO VICENTE[2], Mark Roddy[3], Paul Ryan[3], Czeslaw Jedrzejek[4].

[1] *Waterford Institute of Technology, Cork Road, Waterford, Ireland*
Tel: +353 51 302952, Email: miguelpdl@tssg.org, fcleary@tssg.org
[2]*Telefónica I+D, Address, City, Postcode., Spain*
[2]*Telefónica I+D, Parque Tecnológico de Boecillo, Parc. 119-120, 47151, Boecillo, Spain*
Tel: +34 983 367500, Email: martagm@tid.es, sobrino@tid.es, arv266@tid.es
[3] *Lake Communications, BIC, Ballinode, Sligo, Ireland*
Tel: +353 71 9156830, Email: mark.roddy@lakecommunications.com, paul.ryan@lakecommunications.com
[4]*IAiIII, Poznan University of Technology*
Tel: +48 61 6653532, Email: czeslaw.jedrzejek@put.poznan.pl

Abstract: This paper discusses the challenges of large scale integration within the context of the massive collaborative mobile & wireless systems beyond 3G research project Daidalos. It first examines the relationship of the Daidalos architecture. It then identifies an integration model that was used for the project, with an overview of the testing processes, tools, and methodologies employed. Next, discusses the Daidalos demonstration scenario, and shows how it was used in the integration execution validation and verification activity of the project. The article closes with a discussion of the research necessary to develop these capabilities further.

1. Introduction

In November 2003, the Integrated Project "Daidalos[1]" started as part of the strategic objective "Mobile & Wireless Systems Beyond 3G" of the EU 6th Framework Programme. Supporting a consolidated European approach, Daidalos' derived vision is a world in which a mobile user can use a diverse range of personalised services - seamlessly supported by underlying technologies at all system levels and from heterogeneous access methods and networks to service platforms. This approach will imply several players in future markets. Companies providing radio access will eventually, be different to those providing core networks. For instance, a user will be able to find peoples printers close to him and he could pay them for its use. The need of heterogeneous and interconnected platforms will change the vision of business as it stands today. One of the main project objectives is to demonstrate the results of its research work through a strong focus on user-centred and scenario-based development of mobile technology. The Nidaros demonstrator was defined as a scenario based description of the Daidalos architecture encapsulating the results of the first two years of research from the three technical work-packages integrating the individual components developed in each work package into a common prototypical solution.

Given the level of complexity in terms of number of software components (more than seventy), hardware nodes (more than twenty) and their dependencies, including the number

[1] DAIDALOS - Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services (EU Framework Programme 6 Integrated Project), http://www.ist-daidalos.org/

of partners involved (forty six), as well as time and budget constraints, the integration activity of the Daidalos project is a unique experience in the European R&D programme.

1.1 - State of the Art

Integration and testing is an expensive process phase [1], but is required for concept validation [2]. In a research project such as Daidalos, conceptual work had not completed in the early stages of the project [3], and therefore changes in project scope has been continuous, up to the release of the first prototype, Nidaros. In this kind of scenario, it is obvious that interface errors could arise because of specification misreading, misunderstanding and invalid timing assumptions.

Therefore, making integration a matter of discipline [4] on the project had to be employed. This requires a formalisation of the whole integration and testing cycle [5] in order to increase efficiency and contribute to the success of the Daidalos project in terms of achieving its objectives. Having identified the test sites, the project needed to benchmark and validate the progress and completeness of the integration process. An analysis was carried out of the Nidaros scenario and a series of conformance tests were developed, which are then used to validate the demonstrator. The conformance test specification is an important tool in measuring the progress being made in the integration activity, and given that the objective is to specify a number of test cases that will be used to validate the demonstrator, the WP5 partners had to take into consideration that test cases maybe run by people that may not have sufficient knowledge of the workings of all three technical work-packages and therefore it was made a requirement that this conformance test specification had clear details and instructions on:

- How each of the tests is to be set up
- How each of the tests should be performed
- How each of the GUIs should appear to the tester
- What results should be expected from the test

This paper will commence with an introduction to the Daidalos architecture, in Chapter 2. In Chapter 3 a proposed Integration Model will be presented, with an overview of the testing processes, tools, and methodology used, it also contains a brief overview of each of the major areas of testing as an aid for understanding the contents of the subsequent test details and test results chapters.

Chapter 4 will give an overview the test methodology for the Nidaros test cases, which are derived from the Nidaros scenario. Chapter 5 will highlight the results of this testing, especially in the “Execution and Validation” activity in WP5. Finally Chapter 6 will offer a discussion on the activities and research necessary to develop these capabilities further.

2. Daidalos architecture – and scenario

The Daidalos architecture is predicated from the integration of the three technical work packages WP2 “Network Integration”, WP3 “Service and Network Management” and WP4 “Pervasive Systems”. The objective of WP5 with respect to architecture is to identify the interfaces between each of these work packages and to co-ordinate the integration of these interfaces.

WP2 network architecture is comprised of single-hop radio access networks, multi-hop ad-hoc networks and moving networks. This WP covers Terminal mobility, Network mobility and Ad-hoc networking The WP3 architecture defines a Service Provisioning Platform (SPP) comprised of services for Quality of Service, Network Management, Network Monitoring, Security, Authentication, Authorisation, Accounting, Audit, Charging and Multimedia. This Service Provisioning Platform makes the tools available for creating services and applications on top of integrated heterogeneous access networks including

broadcast networks. The goal of WP4 is the development of a Pervasive Service Platform (PSP). The Pervasive Service Platform (PSP) co-operates with Service Provisioning Platforms (SPP) to achieve its main task: the provisioning of pervasive services. Wp4 is made up of the following sub systems, Context Management, Rule Management, Event Management, Personalization, Pervasive Service Management, Security and Privacy Management.

Scenarios in Daidalos are used to guide and merge the development of the various technologies and conceptual models. Daidalos selected scenarios (being mobile university and automotive mobility) were broken down into steps and analyzed in detail using use case analysis methods. Each step was analyzed to identify technologies used, and possible interfaces among the components used in the step. Different views of the scenarios (e.g. end-user view, operator view) were also created.

Scenarios were evolved in parallel to technical development in Daidalos in order not to hamper these developments. This was accomplished through the initialisation of various task forces to further evolve and implement the Nidaros scenario.

- Scene Analysis Task force (Input from all WP2/3/4)
- Scene Step Analysis Task Force (Input from WP5 assigned Step responsible)
- Mapping and Deployment Task force.

Scenario descriptions were created in close cooperation with technical experts from WP2, WP3 and WP4. Portions of the two scenarios were chosen for defining and describing an integrated demonstrator where most of our integration work would be focused. The selection of a sub-scenario for this integrated demonstrator (called Nidaros) was guided by many factors: most innovative and useful technologies to be demonstrated, most promising technical development since project started to be included, combining the two scenarios into one demonstrator, demonstrating key concepts such as mobility, broadcast and pervasiveness.

Figure 1 Daidalos: Nidaros Scene Demonstration

3. Proposed Integration Model

To tackle the complexity of the architecture the proposed model to integrate these three technical strands was to address the Integrated Testbed, Integration-oriented Developments, and Conformance Test Specification.

There were two main integration sites for the project, one at the Sophia Antipolis, in France and one at the Aveiro, in Portugal. Within Daidalos, work packages WP2, WP3 and WP4 pertain to layered functionality of the telecommunication system and develop corresponding subsystems. Detailed integration procedures and integration experience is presented in D521 [6], but to summarise the function of WP5 was to test 5 subsystems as intra-workpackage integration activities by WP2 (1 subsystem), WP3 (3 subsystems), and WP4 (1 subsystem) using the distinctive testing classifications as expressed in section 3.1.

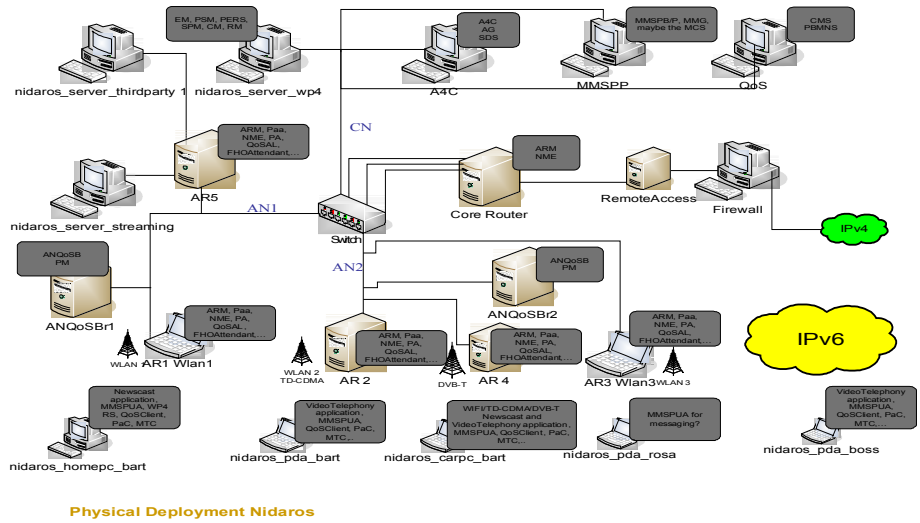

Figure 2: View of Sophia Antipolis Demo Site

3.1 Testing and Test Management

Best practise in testing components/subsystems can be classified into several groups [7] the most obvious debugging and testing techniques, widely recognized and documented are listed below:

- Development of Functional Specifications, as the basis for the testing process.
- Reviews and software inspections.
- Formalisation of preconditions/postconditions: this method relies on the idea that each piece of code, function or program has a precise entry and output criteria.
- Usage of Functional tests and variations.

As the Daidalos project was mainly concerned with demonstration and verification of a concept, such purpose requires “Use Case Scenario Testing”. Scenario Testing goes beyond detecting simple errors (a single feature doesn't work). Testing activity in WP2, WP3 and WP4 may consider features in isolation. Within WP5 the context may be broader and a tester must have sufficient knowledge of the domain, particularly to test a system for cases reflecting the ways in which skilled users will use the program.

There are several strategies to perform system integration – building a system from components. The most desirable testing strategy is sandwich testing that allows for parallel testing activities [8] for Daidalos a WP3 (subsystem)-WP4, or a WP3 (subsystem)-WP2 testing has been proven most effective. The formalisation of roles and integration workflow specification contributed to the project harmonisation across three testbeds. For Daidalos, testing involves three groups of participants: Developing partners, Test site manager partners and Test Managers.

Developing partners have the responsibility of working together to provide components that have already been unit tested and undergone a degree of integration testing with ‘near neighbour’ components for the final integration testing. They contribute most during test development and evaluation. Test site managers were concerned with providing appropriate hardware, integrating specialised hardware sent by developing partners and installing software components. Each of the three technical work-packages assigned Test Managers and it was the responsibility of these Test Managers to define the Nidaros Conformance test cases, taking into account the individual work-package conformance tests and the interface conformance tests with the other work-packages.

Given the participants involved, the management of testing has three overriding goals:

- To schedule and perform tests that are clearly defined and are chosen to significantly contribute to the current and future Daidalos integration and research.
- To ensure that test commissioning, execution and reporting is as agile as possible without sacrificing the quality and usefulness of the tests.
- Minimise the resources in terms of time, equipment and partners expended during the testing process.

Scheduling involves co-ordination with test site managers who also have a managerial role in negotiating the scope and schedule of the integration testing, in ensuring the tests are conducted and results disseminated within Daidalos and to the wider research community. It also includes remote work from developers, aiding the testing process and saving partners travel expenses. One of the biggest challenges during this process was to organise around a hundred people and split them up in testbeds. Performing many tests increases the risk of some developers occasionally being idle during some test allocated days even when they have to travel to the cities where testbeds are situated.

3.2 Test Cycle Management

The test cycle consists of five phases:

1. Definition: Agreement of scope and functionality to be tested.
2. Commissioning: Setting up of the test environment.
3. Execution: Performance of the tests.
4. Reporting: Recording the results of the tests and communicating these with interested partners.
5. Evaluation: Disseminating the results of the tests and taking other appropriate actions.

Sample tasks In the Definition phase for the Test Site Managers include the following

- Co-ordinate with the developers on the scheduling of the tests.
- Input the site capabilities into test definitions.
- Agree the content and schedule of the tests based on development cycles, site availability and overall integration goals of the project.
- Agree test outcomes and dissemination activities for the tests in question

A useful solution for prioritising information processing is the ‘Wiki', technology. The Wiki has been deployed and introduced to practice at http://wiki.ist-daidalos.org/. The website is protected with a username/password to prevent casual browsers using it. It allows the consortium to maintain a considerable amount of information in a manner more convenient and efficient than maintaining a document on MoreGroupWare. The contents of the Wiki have been divided into several distinct sections. The ones found most effective are related to fastest changing information: such as weekly integration plans, booking tables for machines used for installation and order of component installations.

4. Test Methodologies

Each test definition includes the expected output of the test report. All test errors are noted and described in the test report, after which the developers responsible for that part of the code will correct the bug for the next release. If the bug was a showstopper developers were available remotely to fix these bugs. All test cases include visualisations, which can be used as a debugging tool to trace any problems found. The visualisation tool was developed in WP4, and allows for the real-time display of the workings of main system components during a testing.

The Nidaros Test Case naming convention is defined as follows:

N(Nidaros)<Nidaros step # (one digit)> : <test number(two digits)> (Eg – “**N2:03**”, corresponding to Test Case 3 inside Step 2.)

In D512 [9] the Test Cases (TCs) are structured according to the logical flow of Nidaros steps. Within each step several Test Cases were specified, each comprising two main parts: Test Definition and Test Report.
The Test Definition must include the following information:

- The name of the test case, The Systems/Subsystems under test, The Components Involved
- The Protocols, APIs and Interfaces (IFs) involved (Intra- and Inter-WP)
- Test summary (short description of the scenario to be tested)
- A list of Initial Conditions, e.g. what other services should already be running and are needed for the execution of the current TC
- Break of the TC into a series of time-successive actions that must be followed by the tester

The Test Report must include:

- The verdict for each action specified.
- A final verdict for the complete TC
- Name of the person executing the test, the date and the test site
- Test Case closure actions
- Impact on Demos if test fails
- Actions to be performed if test fails to solve problem
- Other comments

Results and Analysis

Two main conformance testing efforts were carried out in the Daidalos project, one in November of 2005 at the Sophia Antipolis site, in France and one in March 2006 at the Aveiro test site in Portugal. However it was found that at the first integrated test bed in Sophia Antipolis there were a number of issues with network stability, related to Mobile IP, which prevented a full detailed conformance test to be carried out.

At the next integrated test-bed in Aveiro, there where continuing problems with mobile IP and its related network in-stability continued until it was eventually discovered the problem was actually a part of the Mandrake Linux distribution and it was decided to switch over from Mandrake to the Debian-based Ubuntu Linux.

Once the conformance test got underway, a total of 120 test cases were planned to be carried out following the test specification of D512 [9]. The tests followed Nidaros scene step by step (from step -1 to step 6). Each test was planned to be performed several times (at least five) to guarantee consistency in the results. During the course of the testing 89 test cases were attempted, and 59 overall being successful in the first iteration. Faults detected were reported to the corresponding developers and were later re-tested multiple times in the next test phase. Figure 1 below provides an overall visual report of the planned, attempted , successful and failed test cases executed during the first iteration in Aveiro.

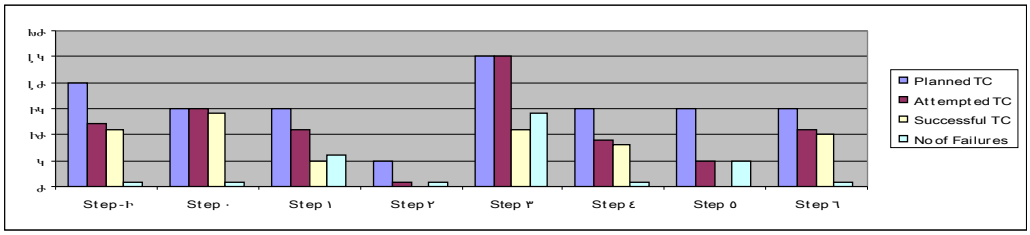

Figure 3 Test Case Execution Overview

In order to obtain performance measurements all the test cases selected, had to pass, with no failures. An example performance test cases was ‘measuring the time for the Newscast Application to go on hold’, after running this test case multiple times it was

determined that it took approximately between 5-7 seconds for the application to hold, the performance measurements for this test case are documented below.

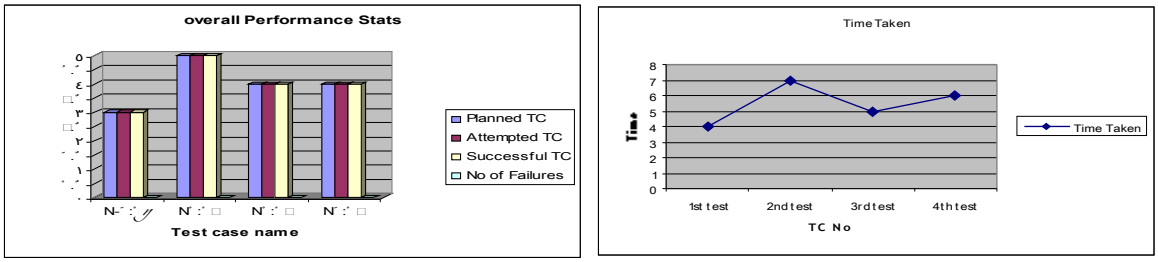

Figure 4 Overall performance statistics & Time Taken for Newscast to go on hold.

There were a number of limitations encountered during conformance testing which hindered the progression of test case execution. When faults were detected, they were immediately reported to the respective developers, and then were re-tested during the iteration phase II of testing. Mobile IP proved to be very unstable and also some the machines hosting Nidraos developed software (eg Bartpda) crashed and had to be rebooted, and the application restarted, which proved time consuming. Even if the Nidaros integrated software was running for some time, it was found that stability was not as good as was needed. When faced with a large number of components, both developed by Daidalos and also third party software, a minor problem in one of them compromised the whole system. A lot of undocumented initial steps (in D512 [9] test cases), had to be figured out and configured prior to the test case execution this was also extremely time consuming. Having stated these limitations, overall, the results were satisfactory. The main integrated functionality involving the joint work of software produced by the three developing Work Packages (WP2, WP3, and WP4) has been successfully tested. Other iterations for conformance testing were also carried out at the Aveiro test site, in order to correct bugs and faults identified during the first Aveiro conformance testing week. This in turn enabled further regression testing and the execution of previously failed test cases, which were successfully executed. This a further documented in the D531 [10] validation report.

Conclusions

When the work in conformance testing started, it was found that the test-bed was not stable enough when working for long periods of time. The cause was not the Daidalos software but the implementation of MIPv6 used (a base software not implemented by the project).

Because this stability was required to carry out the tests, this software was updated to an available more stable version. To do this upgrade, a change in the kernel and Linux distribution was needed, and this affected other modules implemented within Daidalos, due to dependencies with the kernel. Conformance testing was performed on the Aveiro testbed during Spring of 2006. At this point the main conclusions from testing include:

- Tests performed on the test-bed successfully used Mobile IP, it remained very stable on Ubuntu in comparison to earlier attempts using mobile IP on top of Mandrake.
- WP2 WLAN drivers were stable compared to their performance in Sophia.
- This was the first successful integrated testing that utilised all work packages.
- Route Optimisation worked over Mobile IP, with very noticeable performance improvements as compared to earlier tests on WP4 standalone test-beds.
- There was no USB software installed on the test-bed, which meant that most of the authentication and authorisation test cases failed. However these tests had passed in Sophia Antipolis. There was also no TD-CDMA available.
- Registration module and several other components were dummy's because of lack of integration on Ubuntu (previously worked under Mandrake).

- Excluding the tests involving any of the missing components, almost all of the tests were passed successfully.

This analysis shows that even though the Nidaros software was not completely integrated at the time of conformance testing in Aveiro, the overall results are quite satisfactory. In a last iteration of integration, some missing components were finally integrated and tested. The key point now is that developers have realised the importance of meeting deadlines, and to run different cycles of the process, in order to identify any possible mistakes at an earlier stage. From a managerial point of view, WP5 (Integrated System and Evaluation) set up a very well defined process in which different cycles of installation, integration and testing could take place. But then again, the problem of having such a big project, with so many partners meant it was very difficult to make everyone meet their deadlines and having all the necessary pieces of code delivered on time. There was some culture difference between developers and integrators. Due to this testing procedures changed from envisioned by WP5 black-box-testing to primarily white-box testing performed by key developers and the final stages of the project. As Daidalos moves forward into its second phase, it will be recommended that developments be clustered in two cycles of work.

- Integration of more mature work will result in implementations, integration, prototypes, demos.
- R&D for more futuristic work will result in concepts, validation via modelling, simulations, analysis, early proof-of-concept prototypes.

There will be several further iterations of conformance tests during the lifetime of the project allowing developers identify errors between interfaces and functionalities earlier. Integration during the technical weeks was a critical activity. However integration activities need to be started earlier, with clear step-by-step objectives and with better planning. They were very important for face-to-face discussion and problem resolution, even if this was somewhat ad hoc at times. Overall the Nidaros scenario was integrated and validated successfully, providing beneficial and interesting results and conclusions, useful as a learning curve for future development, integration and validation work phases for Daidalos.

References

[1] Dalal, S., Mallows, C: Factor-Covering Designs for Testing Software. In: Technometrics 40(3), August 1998, pp. 234–243.

[2] Sommerville, I.: Software Process Models. In: ACM Computing Surveys, Vol. 28, No. 1, March 1996, pp 269 - 271

[3] MacCormack, A., Verganti, R. , Iansiti, M. : Developing Products on "Internet Time": The Anatomy of a Flexible Development Process: Management Science, Volume: 47 | Issue: 1 2001, pp: 133-150

[4] Berczuk, S. , Appleton, B.: Software Configuration Management Patterns: Effective Team-work, Practical Integration, Addison-Wesley, 2002.

[5] Giraudo, G., Tonella, P., Designing and Conducting an Empirical Study on Test Management Automation, Empirical Software Engineering, Volume 8, Issue 1, Mar 2003, Pages 59 – 81.

[6] DAIDALOS Deliverable D521, First Phase Integration Report, http://www.ist-daidalos.org

[7] Whittaker, J.,: "What is Software Testing? And Why it is so hard?" http://www.sba.oakland.edu/faculty/rajagopa/Courses/515/testing2.pdf ".

[8] Bruegge, B., Dutoit, A. H.: Object-Oriented Software Engineering:Using UML, Patterns, and Java, 2nd Edition: Prentice Hall, Upper Saddle River, NJ, September 2003.

[9] DAIDALOS Deliverable D512, Conformance Tests Specifications, http://www.ist-daidalos.org/

[10]DAIDALOS Deliverable D531, First Phase Evaluation Report, http://www.ist-daidalos.org/