ICT-MobileSummit 2008 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2008
ISBN: 978-1-905824-08-3

Poster Paper

A Proof of Concept Architecture for Self-Configuring Autonomic Systems

Panagis MAGDALINOS[1], Spyros POLYMENEAS[1], Pantelis GLIATIS[1], Xenophon FAFOUTIS[1], Andreas MERENDITIS[1] , Costas POLYCHRONOPOULOS[1]
[1]National and Kapodistrian University of Athens*, Panepistimiopolis, Athens, 15784, Greece*
Email: {panagis,spyros,p.gliatis,fontas,amer,cpoly}@di.uoa.gr

Abstract: Reconfigurability is the collection of software and cognitive radio technologies that aims to differentiate user perception in volatile radio conditions while optimizing the use of network resources. This paper presents a proof of concept implementation of the end-to-end management and control system architecture being developed in E2R II project based on the concept of a Reconfiguration Management Plane. In addition, extensive experimentation has been carried out in order to validate of the applicability of the proposed ideas and accompanying technologies in real life scenarios.

Keywords: Reconfigurability, Reconfiguration Management Plane

1. Introduction

The recent explosion in the range and capabilities of wireless communications devices is perhaps unequalled in history. In addition to ongoing advancements in the generations of mobile communications systems, wireless provision for a wealth of different means either has become or will soon become commonplace, facilitating the capability to relay a complete plethora of different application services. The scopes of such solutions vary greatly, from local area networks (IEEE 802.11 WLANs), wide area or metropolitan area networks (e.g. IEEE 802.16), to, at the other end of the scale, personal area networks (e.g. IEEE 802.15). To the average soul in the street, configuring devices among the resulting various possible connectivity and software options is highly challenging, and is becoming far more complicated as time progresses. Moreover, the range of different devices possessed by the average person is becoming mind-numbing, such that they might easily lose track of the data that they control and encounter various issues in the management among the devices to hand.

Given these challenges posed, reconfigurability, twinned with autonomics capabilities, offers a serious solution to the complexity among devices that is set to confound a significant proportion of the population. Through properly managed and automated reconfiguration procedures, at wireless as well as higher layers, issues with devices can be dealt with in a way that is largely transparent to the user, enabling automatic configuration for optimal operation and indeed fixing various bugs and other problems which must be dealt with in order to improve the user experience. Moreover, reconfigurability across all layers offers devices, and indeed network elements, the opportunity to dynamically adapt the RATs/protocols being used, for such reasons as to enhance the user experience (e.g. QoS) or overall wireless resource efficiency based on the characteristics and available networks at that locality. This reason alone is sufficient to justify the interest in reconfigurability that has perpetuated the research community over recent years, and indeed is a major economic driver behind the effort in this direction that has been expended.

This paper presents the end-to-end management and control system architecture being developed in E2R II project based on the concept of a Reconfiguration Management Plane, which comprises a network-agnostic protocol-independent model that captures software and cognitive radio solutions enriched by self-ware capabilities. Moreover it presents a proof of concept implementation together with an extensive validation of the applicability of the ideas in a real life scenario.

2. Reconfiguration Management Plane

This section describes the overall E2R II system architecture for autonomously reconfigurable user equipment and network elements, integrating functional entities for software and cognitive radios whereas capturing autonomic communication aspects [9]. Figure 1 presents a high-level overview of the RMP logical model that provides elements (terminal equipment, base stations, and routers) with the necessary user, control, management, and OA&M capabilities so as to be autonomously reconfigured. The RMP offers the following "reconfiguration services": 1) Over-the-air software-download resulting in element reconfiguration, 2) Dynamic spectrum access and exchange, 3) Dynamic network attachment and intersystem handover, and 4) Reconfiguration of hardware resources aiming at the optimisation of physical layer performance (e.g., reduced power consumption).

The RMP model consists of the so-called Selfware Reconfiguration management and control Plane (SRP), which views the entire element as an autonomous entity, offering cross-layer control and management reconfiguration capabilities. The SRP caters for: 1) Autonomic decision-making and policy-based orchestration of reconfiguration operations, including negotiation control and decisions for mobility management between access systems, 2) Discovery of reconfiguration services and service provisioning leveraging cognition techniques, 3) Administration of the software-download process for unicast, multicast and broadcast modes, 4) Self-configuration and self-management, and 5) Retrieval and processing of knowledge and contextual information, including spectrum and radio resource optimization.

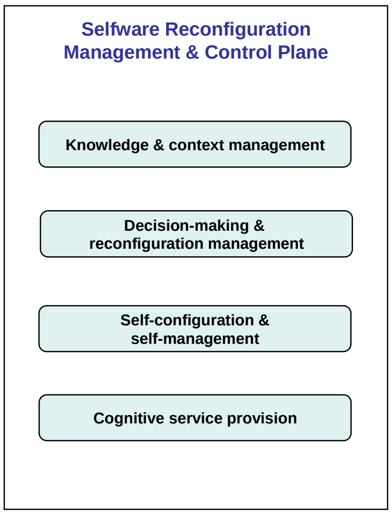
Selfware Reconfiguration Management & Control Plane
Knowledge & context management
Decision-making & reconfiguration management
Self-configuration & self-management
Cognitive service provision

Figure 1: Reconfiguration Management Plane (RMP) Logical View

The SRP consists of the Knowledge and Context Management module, the Decision-Making and Reconfiguration Management module, the Self-Configuration and Self-Management module, and the Cognitive Service Provision module. Each module consists of functions addressing specific reconfiguration tasks. The detailed description of the functions follows.

The Knowledge and Context Management module consists of the Performance Management (PeM) function, the Profile Management (PrM) function, and the Resource Management (RsM) function. This module also includes a set of functions for spectrum and radio resource efficiency, namely the Cognitive Pilot Channel Controller (CPC-C), the Joint Radio Resource Management (JRRM) function, the Traffic Estimator (TE), the Dynamic Network Planning and flexible Management (DNPM) function, the Spectrum Allocation Management (SAM) function, the Spectrum Economic Management (SEM) function, and the Spectrum Trading (STM) function [1].

The Decision-Making and Reconfiguration Management module consists of the Autonomic Decision-Making and Reconfiguration Management (ADM&RM) function, the Policy-Management and Self-Governance (PMSG) function, and the Mobility Management (MM) function.

The Self-Configuration and Self-Management module consists of the Cognitive Self-Healing (CoSH) function, the Self-Configuration (SeCo) function, the Software Download Management (SDM) function, the Pervasive Access and Security Control (PASeC) function, and the Reconfiguration Charging Control (RCC) function.

The Cognitive Service Provision module consists of the Content and Service Adaptation (CSA) function and the Reconfiguration Services Discovery (RSD) function.

3. Proof of Concept implementation

The self configuring autonomic systems prototyping environment (CASPER) is the proof of concept implementation of the RMP. Its architecture accompanied by implementation details has already been presented in [3] while its theoretic foundations have been outlined in [2].

The Self-Configuring Autonomic System – Prototyping Environment (CASPER) is composed of three entities: 1) A realization of the Reconfiguration Management Plane (RMP) on the network side which undertakes the responsibilities of reconfiguration management and control as well as value added service provisioning together with decision making capability. 2) A realization of RMP on the terminal side, technologically adapted to the low capabilities of the end user device. 3) An implementation of the Cognitive Pilot Channel Controller concept [3].

Figure *2* provides an overview of the implemented prototype accompanied by the technologies employed for the realization of each module.

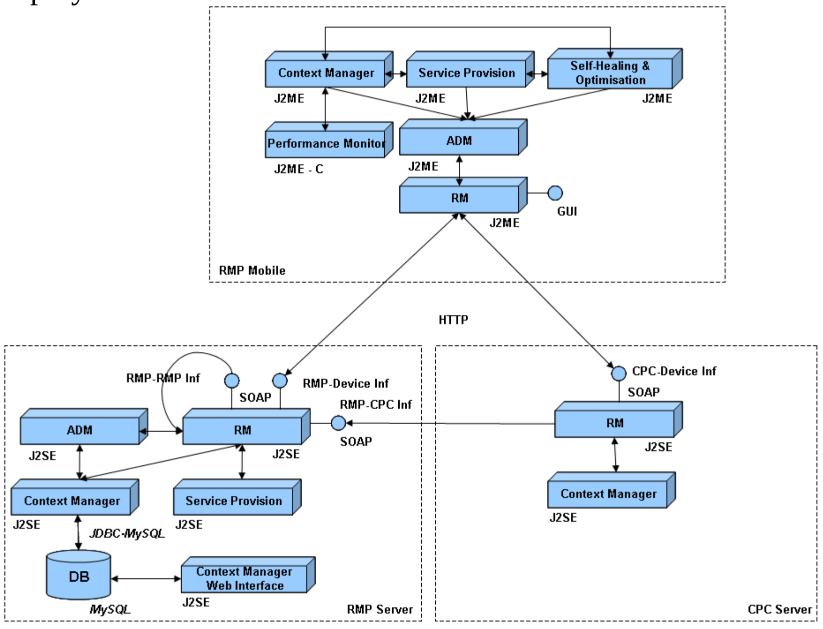

Figure 2: Overview of CASPER

4. Experiments

A series of experiments was performed on CASPER, in order to measure the effectiveness of the employed technologies. These experiments focus on the memory usage and the time required for the execution of several modules of the platform.

On the server side, the cases of Terminal Classification, Software Classification and Content/Profile Retrieval have been evaluated. In each scenario, a simulation of simultaneous requests has been performed. Each of the cases above has been evaluated with 50 simultaneous requests (100 repetitions per case). All experiments have been carried out in a 1.8 GHz Pentium IV processor, with 1GB RAM running Ubuntu Operating System, version 7.10. The x axis of the following diagrams represents the Time or Memory requirements of each procedure. Memory requirements signify the load imposed by the procedure on the system; with respect to the system state when idle (a value of 18 in the x axis signifies 18% increase in memory requirements). The y axis represents the percentage of requests. All experiments were performed on randomly generated contextual information (terminal and software profiles). Classification procedures employ an ontology (OWL [4]) with policies expressed in semantic web rules (SWRL [5]) while the decision is derived through reasoning of the ontology and the corresponding rules.

Figure 3 contains the results of the terminal classmarking. The obtained results signify that the vast amount of requests is processed within 12ms, a time frame totally acceptable for non time critical procedures, like the one performed in this case. Meanwhile, each request imposes a mean load of 15% on the system.

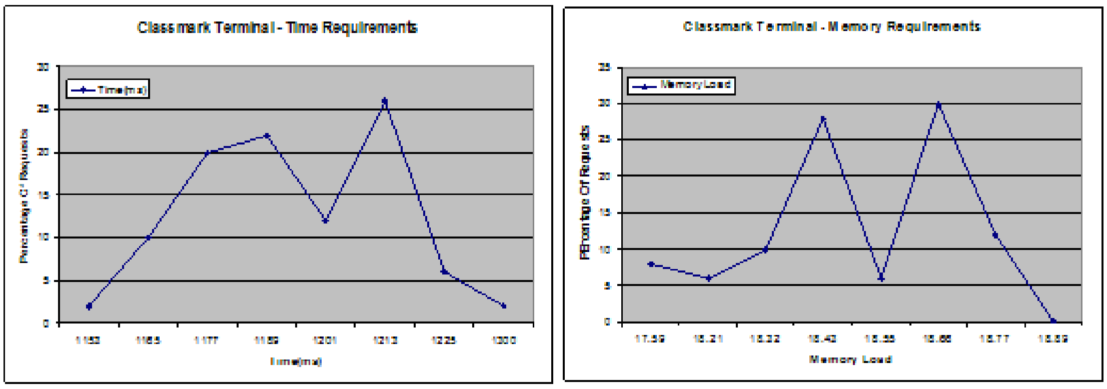

Figure 3: Terminal Classmarking – Time and Memory Requirements

Figure 4 contains the results of the software classification. The obtained results signify that the vast amount of requests is processed within 2.5ms, a time frame totally acceptable for time critical procedures. Each request imposes a mean load of 25% on the system.

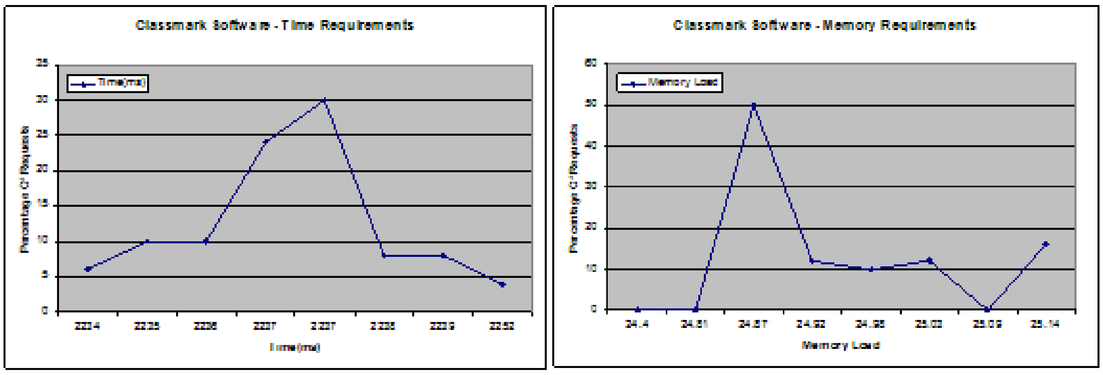

Figure 4: Software Classification – Time and Memory Requirements

However, as seen in the experiment's results, using ontology reasoning for decision making is a time consuming procedure. Consequently one should avoid implementing time critical operations with the use of these technologies. Additional effort should be also put towards the direction of the procedure's optimization. The latter can be accomplished either by multiple small and fast-reasoned ontologies or by policy based code generation which will remove the reasoning part completely.

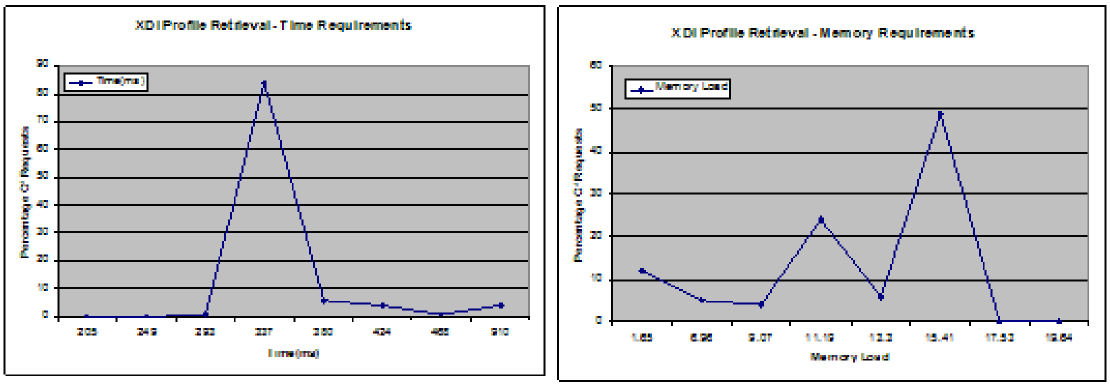

Figure 5: XDI Content Retrieval – Time and Memory Requirements

Figure 5 contains the results of the XDI [6] network profile retrieval. This procedure retrieves the profile of the network managed by the RMP from the database and converts it into XDI form. XDI is a promising technology in the field of profile definition because it allows the sharing and linking of existing data in a very intuitive way, following the metaphor of the web and without modifying their native formats. As expected, XDI profile retrieval required some extra time but this overhead is acceptable and doesn't prevent XDI from being a valuable tool. Nevertheless research could be made to further enhance XDI's response time and thus its abilities and usefulness. Finally the use of caching scheme is considered as a simple and efficient way of ameliorating the results.

On the terminal the following use cases were evaluated:

- User Notification due to Service Update: This procedure implements two similar scenarios. The first scenario is when the Tariff Class or the Quality of Service of the running service changes by the service provider. The second is when the user's preferred Tariff Class or Quality of Service change while consuming a service. In both scenarios the action taken is the same. ADM rechecks the policies with the updated profiles and notifies the user or the application in case it is required,
- Software Reconfiguration Decision: In this scenario the ADM retrieves the current CPU, memory load and available battery from the Resource Monitoring Module (RMM) as well as the already installed software and decides whether to allow or deny the installation of a specified software component according to a set of pre-installed policies,
- Protocol Reconfiguration Decision: In the protocol reconfiguration use case, the ADM module of the mobile terminal checks the new components of the protocol, comparing their requirements against the available device resources, and decides according to the installed policies and the RMM output whether a protocol reconfiguration procedure should be triggered or not,
- Service Reconfiguration Decision: The service reconfiguration decision follows the general principals of the use cases analyzed previously. ADM decides based on the contextual and policy information whether or not a service should be reconfigured,
- ADM Network Selection and Connection procedures: This last use case embraces all actions regarding the various handover decisions taken by the terminal. In this case ADM evaluates the various network profiles which are received by the CPC together with their corresponding policies against a set of predefined policies and user requirements. Finally it connects to one of the available networks.

All experiments have been carried out on a Motorola A910 mobile terminal [7]. The following figures demonstrate the results of the experiments performed on the terminal (20 repetitions per case/experiment). The x-axis represents each experiment and the y-axis represents its memory and time requirements. All the experiments were tested on worst case scenarios (every possible rule has been evaluated). All of the scenarios tested comprise an integral part of the ADM instance in the terminal. Due to the fact that modern mobile devices do not support the Protégé and Jess libraries (used for OWL/SWRL reasoning in the server side) the ADM uses policy based class loading. Each OWL property that forms an SWRL rule is pre-coded and pre-compiled. The ADM loads the proper classes according to policies in order to implement a decision.

Figure 6 contains the results of the user notification due to service update procedure. This scenario requires only a few milliseconds in order to be executed and results in minor memory load.

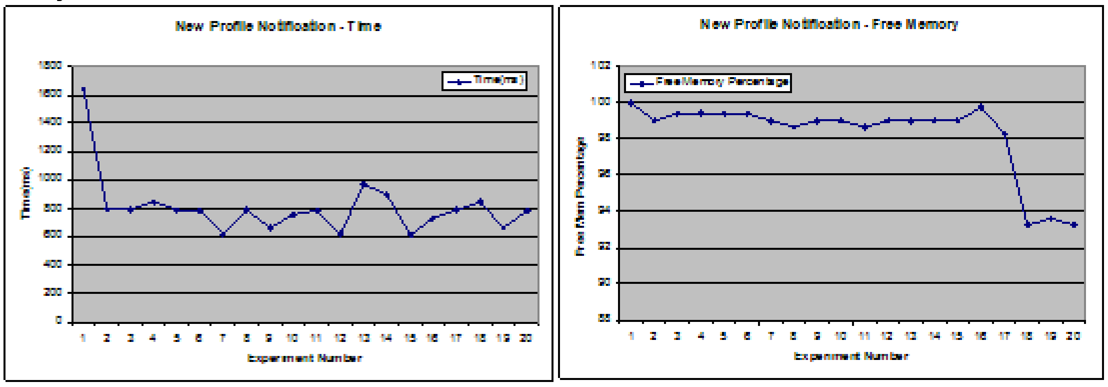

Figure 6: User Notification Due to Service Update

The next experiment (Figure 7) is the implementation of the software reconfiguration decision making. Each experiment was performed with the assumption that 10 software components have been installed on the device. Consequently, the evaluation process validated the new software against ten software profiles. Again, the time and memory requirements are extremely low, verifying the fact that rule based code generation can significantly decrease the resource requirements of the decision making process.

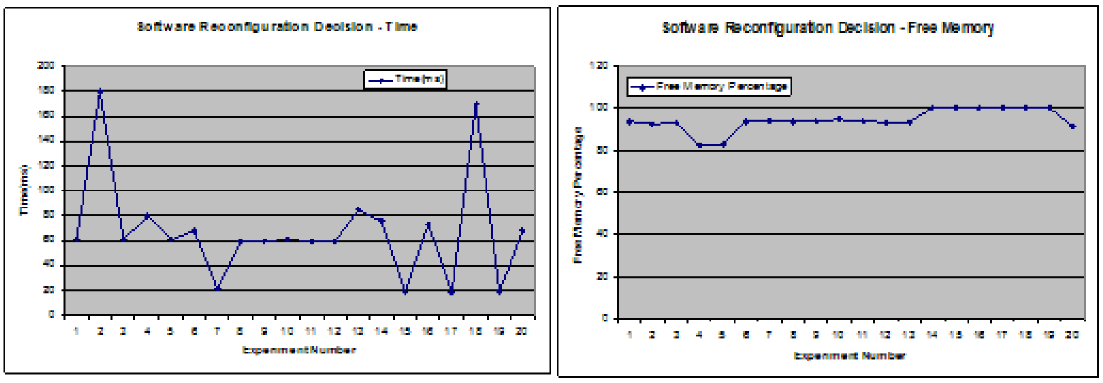

Figure 7: Software Reconfiguration Decision

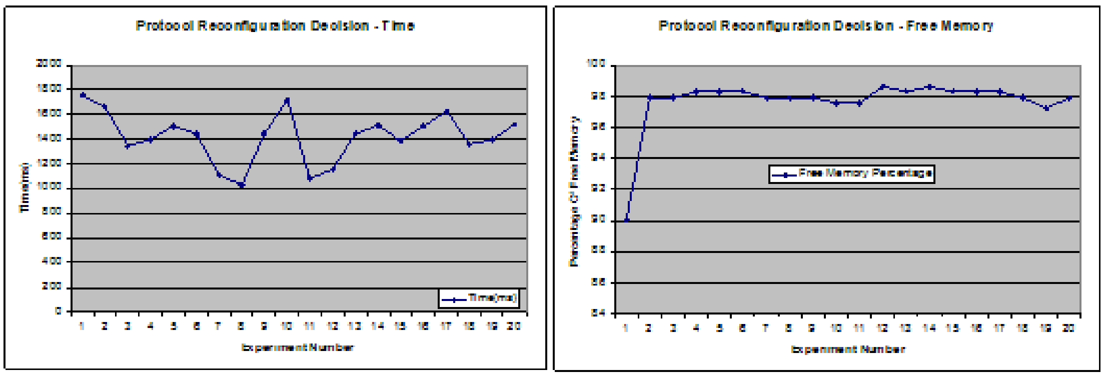

Figure 8: Protocol Reconfiguration Decision

The experiment on Figure 8: Protocol Reconfiguration Decision depicts time and memory requirements for the triggering of the protocol reconfiguration procedure. The experiment has been carried out with a protocol consisting of two components. The results are complementary to the extensive evaluation presented in [8].

Figure 9 illustrates the results of the experiment on the service reconfiguration scenario. In this scenario the ADM decides whether to accept or deny new service applications. The decision is also based on the installed policies and the RMM's output. Each experiment was performed on 10 application profiles.

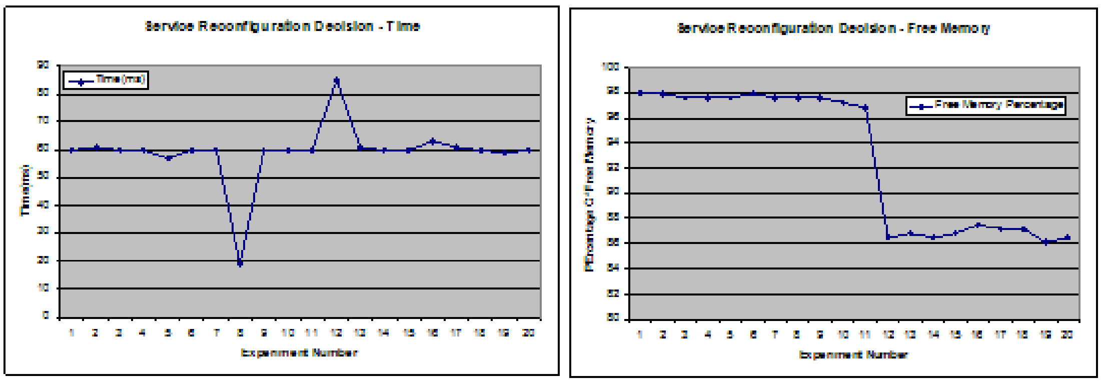

Figure 9: Service Reconfiguration Decision

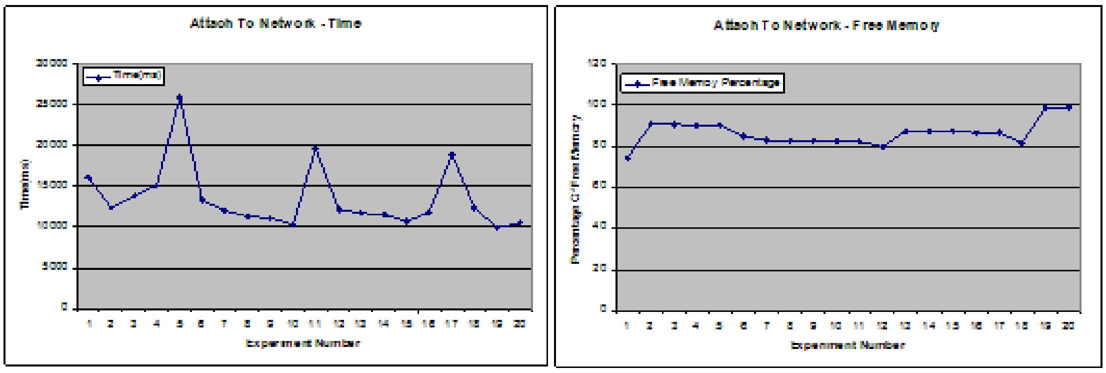

Figure 10: Scanning of Available Networks, Network Selection and Attachment Procedures

The last experiment of the terminal RMP entity was performed on the Attach to Network procedure of ADM module. Figure 10 illustrates the results of this experiment. During this procedure the device uses the RMM input concerning the available networks and together with the CPC contextual information, chooses the one to connect according to user preferences expressed in the installed policies.

5. Conclusion

Based on the specified functionalities, the performance analysis and the experimental results, it is obvious that the CASPER platform integrates advanced functions and mechanisms for the support of reconfigurable and autonomic systems, thereby increasing the overall system flexibility. However it should be noted that the trade-off for the incorporation of such advanced features in a system are the performance degradation and the requirement for increased resource availability; such requirements are not completely fulfilled by existing communication infrastructures. As an alternative solution a subset of the specified functionalities could be selected and deployed in target network elements based on a per case study and analysis of the requirements (i.e. hardware resources).

Acknowledgement

This work was performed in projects E2R II and E3, which have received research funding from the European Community's Sixth and Seventh Framework programme, respectively. This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein. The contributions of colleagues from E2R II and E3 consortia are hereby acknowledged

References

[1]. C. Kloeck et al., &#8220;Functional Architecture of Reconfigurable Systems&#8221, Proc. WWRF#14, San Diego, California, July 20056.
[2]. IST-2005-027714 Project E²R II Deliverable D2.2, "System Architecture, Draft Reconfiguration Management and Control Network Architecture and Analysis of Support Protocols and Mechanisms", December 2006.
[3]. IST-2005-027714 Project E²R II Deliverable D5.3, "Report on the Developed HW and SW Modules", June 2007.
[4]. "E^2R Cognitive Pilot Channel concept", P. Cordier et al, IST SUMMIT 2006, Myconos, Greece
[5]. OWL: Wel Ontology Language: http://www.w3.org/TR/owl-features/
[6]. SWRL: A Semantic Web Rule Language: http://www.w3.org/Submission/SWRL/
[7]. XDI: XRI Data Interchange¨ http://www.oasis-open.org/committees/xdi/
[8]. A910 mobile: http://www.motorola.com/motoinfo/product/details.jsp?globalObjectId=114
[9]. Z. Boufidis et al., End-to-End Architecture for Cognitive Reconfigurable Wireless Networks, Proc. 16th IST Mobile and Wireless Commun. Summit, Budapest, Hungary, July 2007
[10]. E. Patouni, S. Gault, M. Muck, N. Alonistioti, K. Kominaki, "Autonomic Communications: Exploiting Advanced and Game theoretical Approaches for RAT Selection and Protocol Reconfiguration", in the
Proceedings of the Autonomic Networking Conference, Paris, France, September 2006