

A Proof of Concept Environment for Cognitive and Self-x Functionalities

Panagis Magdalinos¹, Christos Tsilopoulos¹, Dimitris Makris¹, Panagiotis Spapis¹, Nancy Alonistioti¹

¹National and Kapodistrian University of Athens, *Panepistimiopolis, Athens, 15784, Greece*
Email: {panagis,tsilochr,dmakris,pspapis,nancy}@di.uoa.gr

Abstract: The recent explosion in the range and capabilities of wireless communications devices is perhaps unequalled in history. Therefore there is a need for introduction of new approaches that tackle the raising complexity. To this end, we present the proof of concept environment for cognitive and self-x functionalities as identified in the context of the End-to-end Efficiency project. The platform comprises an integral part of the overall E3 demonstration environment and has been implemented in accordance to the E3 architectural specification. Through experiments we validate the viability and suitability of the key introduced concepts in future telecommunication environments.

Keywords: Cognitive Radio, Self-x, Autonomic Capabilities.

Introduction

The recent explosion in the range and capabilities of wireless communications devices is perhaps unequalled in history. In addition to ongoing advancements in mobile communications systems, wireless provision has become or will soon become commonplace, facilitating the capability to relay on 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. 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, autonomic capabilities offer a serious solution to the complexity among devices that is set to confound a significant proportion of the population. Through properly managed and automated 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.

This paper presents a proof of concept environment that exploits autonomic capabilities twinned with knowledge mining and management techniques in order to enhance mobile devices with cognitive and self-x capabilities. The work is directly linked with the research conducted in the End-to-End Efficiency Project. Therefore we commence the presentation by providing an overview of the architecture produced by the project (Section 2) as well as its prototyping environment for cognitive and self-x functionalities (Section 3). Afterwards we focus on the particular platform and analyse its functionalities. In section 4 we provide a

detailed description of the supported demonstrations and analyze the obtained validation results. We conclude the paper and sketch future research directions in section 5.

The E3 Functional Architecture

In order to identify the E3 functional architecture we followed well known techniques widely employed in the scientific area of telecommunications (, ,). At first, a number of high level scenarios have been identified which were decomposed into elementary use cases. Each use case addressed a number of functionalities or equivalently implied a certain functional requirement from the system. Due to the vast amount of identified use cases and the overlapping set of functional requirements an organization was required. Consequently, following a top-down approach we merged the use cases into groups with respect to the high level aspects which were addressing and placed all requirements under these use case groups. The grouping procedures of requirements and use cases enabled the identification of building blocks. Building blocks are high level blocks which addressed specific high level functionalities of the system. Afterwards all requirements have been mapped on the identified building blocks. Finally, the requirements have been grouped into logical functions, thus defining the Functional Entities of the architecture.

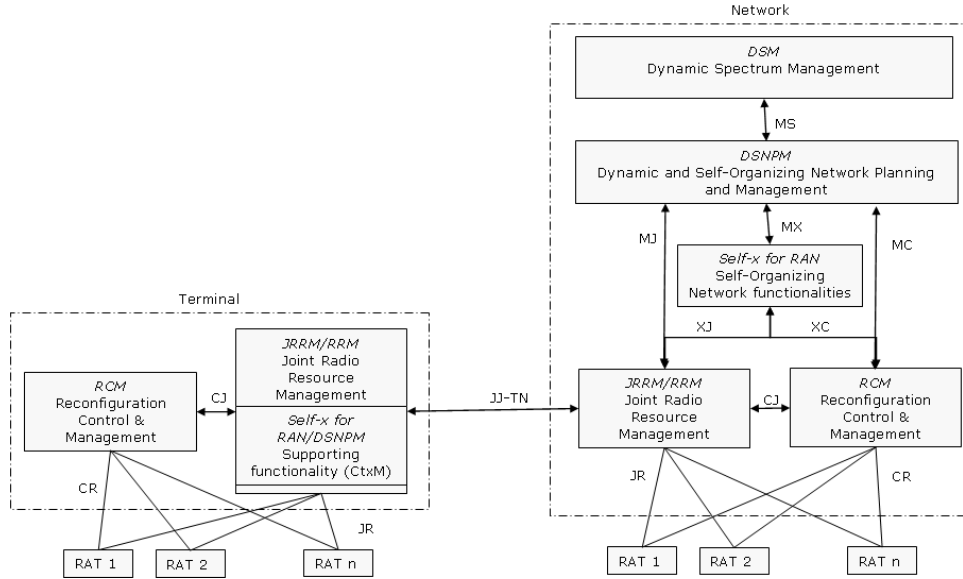


Figure 1: The E3 functional architecture

Figure 1 depicts the E3 functional architecture. On the network side the Dynamic Spectrum Management (DSM) supports spectrum management for different radio systems. The Dynamic Self-Organising Network Planning and Management (DSNPM) provides decision support upon the reconfiguration actions of a network segment. The Joint Radio Resource Management (JRRM) performs the management of radio resources that might belong to heterogeneous RATs. The Self-x for Radio Access Networks (Self-x for RAN) enables the automation of operational tasks. It targets the self-organising functionalities for the radio access network, mainly providing short to medium term decisions. The Reconfiguration Control Module (RCM) is mainly responsible for the execution of the reconfiguration of a terminal or a base station, following the directives provided by the other building blocks, typically the DSNPM, the Self-x for RAN and the JRRM. Finally, Context Management (CtxM) is considered as support operation for Self-x for RAN, DSNPM and JRRM and therefore is identified as an integral part of the former.

In order to support the operations of the E3 functional blocks we introduce the notion of “cognition enablers”, special building blocks that support the collection of information from the network environment. Specifically, terminals use the Cognitive Pilot Channel (CPC)

and extract information related to a number of parameters related to the networks which operate in their geographic vicinity. Cognitive Control Radio (CCR) is an out-band peer-to-peer communication radio between network nodes and enables the exchange of information between terminals. Finally the Spectrum Sensing (SS) block focuses on gaining knowledge related to the available radio systems. Through either CCR or CPC, SS information can be distributed among network nodes.

CCR is extremely useful in the case of ad-hoc and mesh network where there is no central network management entity and terminals are self-organized using the information communicated through CCR. In this context, E3 proposes the deployment of an additional entity, namely the Autonomic Entity Management (AEM) which undertakes all self-management tasks in such environments. The Functional Architecture in this case is depicted in Figure 2. More information related to the E3 functional architecture can be found in .

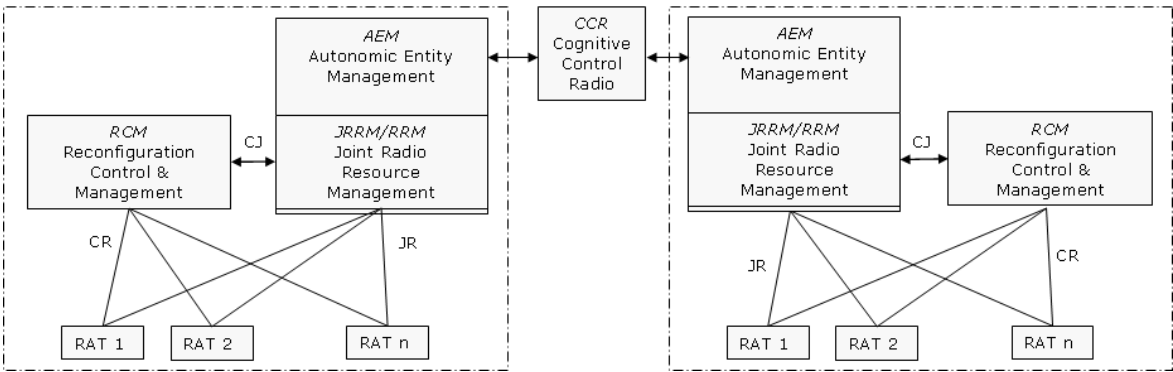


Figure 2: E3 Functional architecture for the ad-hoc case

A Prototyping Platform for Cognitive and Self-x functionalities

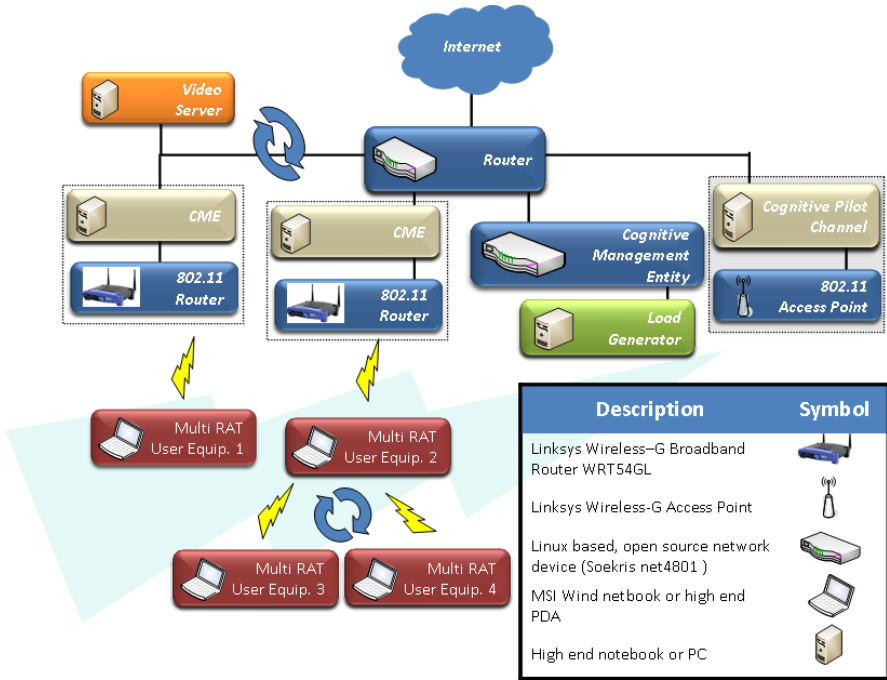


Figure 3: The prototyping platform for self-x functionalities

The E3 prototyping environment for cognitive and self-x functionalities is a prototyping platform that provides the means for demonstration and experimentation with the key concepts and innovation of the E3 project related to AEM and Self-x For RAN building

blocks. This platform is only part of the overall E3 environment which is thoroughly analyzed in .

Figure 3 presents a high level view of the developed platform; the platform highlights E3 concepts such as the CPC, Autonomic Decision making on the terminal side, the Cognitive Control Radio (CCR) etc. The platform consists of several multi-RAT terminals that are receiving context information through the (outBand) CPC. When they have received all the necessary information they evaluate it against their capabilities and they decide to connect to one of the available RATs. The aforementioned RATs are being controlled by a Cognitive Management Entity which aggregates various building blocks of the E3 Functional Architecture. Each CME is responsible for only one Access Point. In other words, the CMEs control the network in a decentralized manner. Moreover, the terminals are capable of exchanging information through ad-hoc connections (CCR) in order to investigate the possibilities of forming clusters to exploit their similar behaviour.

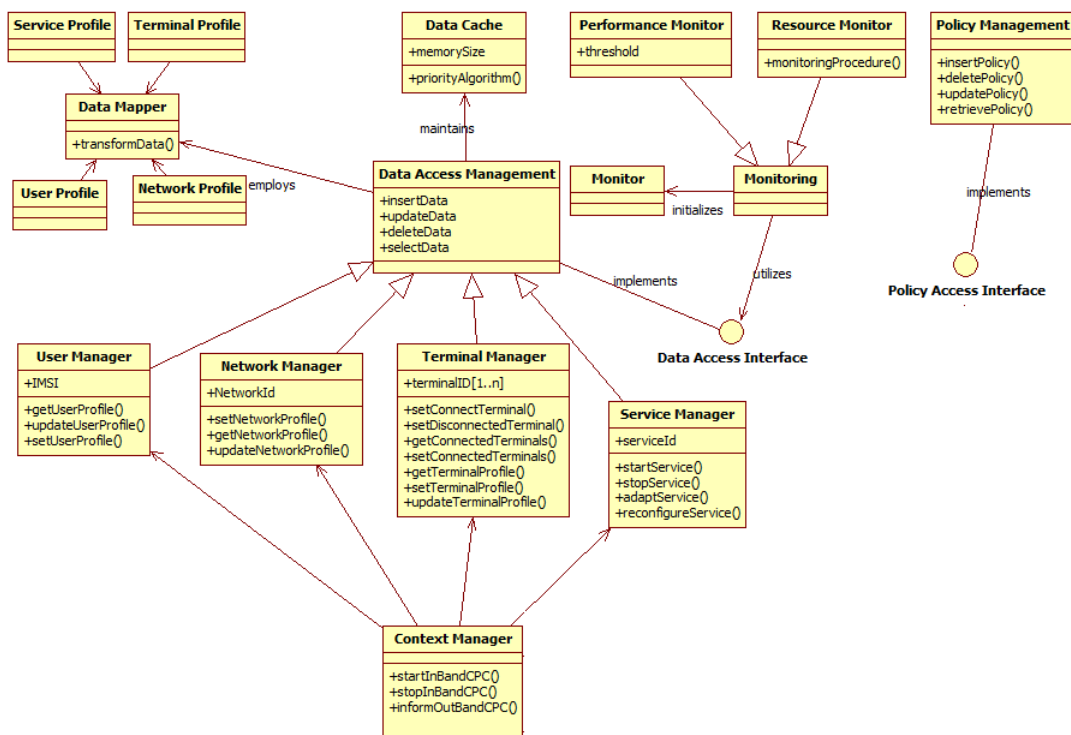


Figure 4: Context and Policy Management

Figure 4 illustrates the Context and Policy Management implementation which is core part for the realization of the cognitive loop (MDE cycle). The core class of the prototype is the Data Access Management that enables the retrieval, insertion, deletion and update of available information. The latter is provided through various monitoring entities (Monitor) Before storing information, the Data Mapper undertakes the mapping of raw data to a higher level representation (Service Profile, Terminal Profile, User Profile, Network Profile) so as to be useful for learning and decision making purposes. All context information is managed by the Context Manager (CtxM). Policy Management implements the same methodology but for user or manufacturer defined policies.

Figure 5 depicts the proof of concept implementation on the prototyping platform for self-x functionalities in the terminal side. As it is shown in the figure, the Context Manager communicates with the InBandCPC Client and the OutBand CPC (both realizations of the Joint Radio Resource Manager (JRRM)) and the Ad-hoc Monitor (a realization of CCR). The Decision Making class comprises the realization of AEM and directs the Controller to

implement specific decisions. The latter is the realization of Reconfiguration Manager (RCM) on the terminal side.

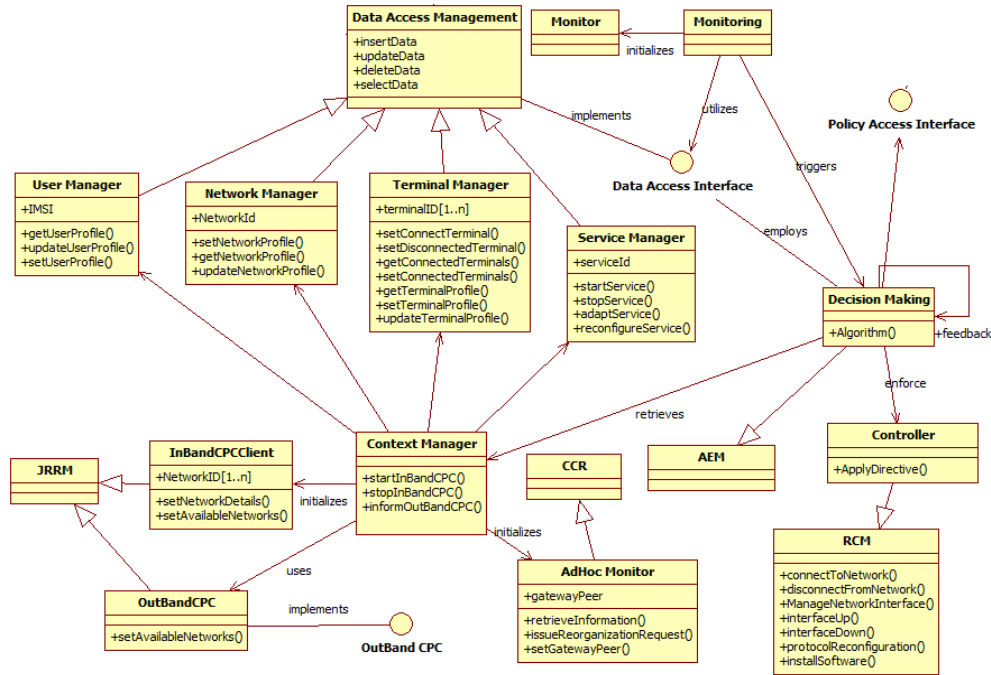


Figure 5: AEM, JRRM, CPC, CCR proof of concept implementation on the prototyping platform for self-x functionalities (terminal side)

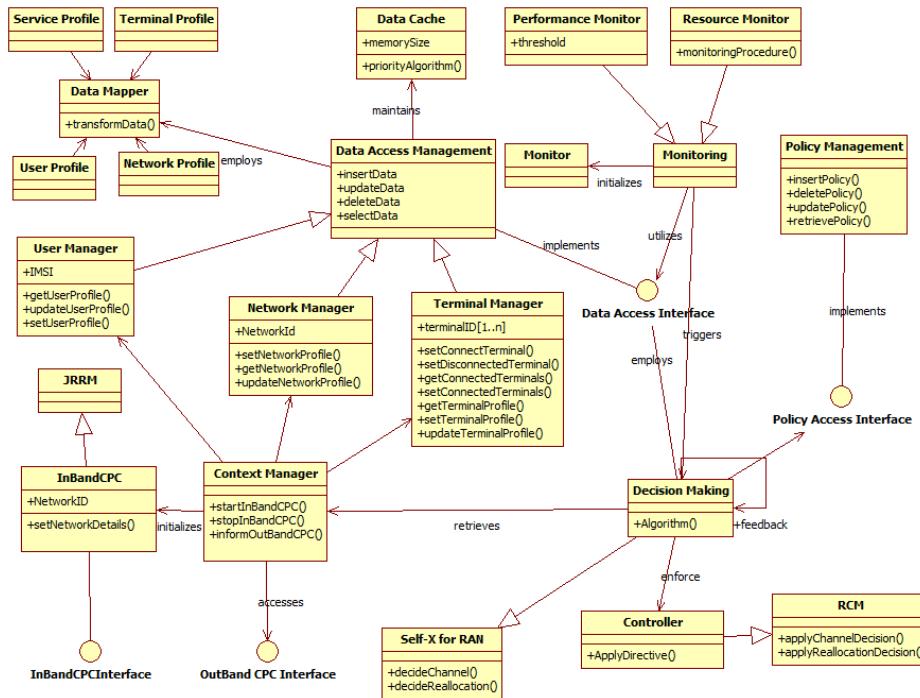


Figure 6: Proof of concept implementation of JRRM, Self-X for RAN, CPC and RCM on the network side. All modules constitute the CME entity of former deliverables

Similarly, Figure 6 depicts how the CME is implemented in the prototype and what are the interrelations between the developed functional components. As it is shown, InBandCPC realizes the JRRM functionalities and the Decision Making class instantiates the Self-x for RAN (Radio Access Network) which enforces the decision to the Controller that realizes RCM on the network side.

The simplest way to provide a proof of concept implementation of CPC is to employ protocols residing in the higher levels of the OSI protocol stack. To this end, the IP based CPC platform employs HTTP in order to transmit the message from a central indexing entity (the CPC controller) to the terminals. Additionally, IP based CPC enables the transmission of network related information from each network to the Controller which in turn periodically informs each terminal. The IP based Outband CPC which is implemented in the Prototyping Platform for Cognitive and Self-x functionalities consists of three parts, namely the CME, the OutBand CPC controller and the receiver on the mobile terminal side (Figure 7) At first the Context Manager informs the OutBand CPC about its current network profile through the OutBand CPC Interface. The latter periodically triggers the OutBand CPC interface from the terminal side in order to inform it for the available RATs in the corresponding area. The realization of the Outband CPC is depicted in Figure 7 and is integrated in the Prototyping Platform for Cognitive and Self-x functionalities.

Finally the simple nature of the implementation enables the successful deployment of the Inband CPC concept, where the terminal queries the network management entity of its network in order to retrieve contextual information related to the operational status of its RAT. The realization of Inband CPC has already been described previously.

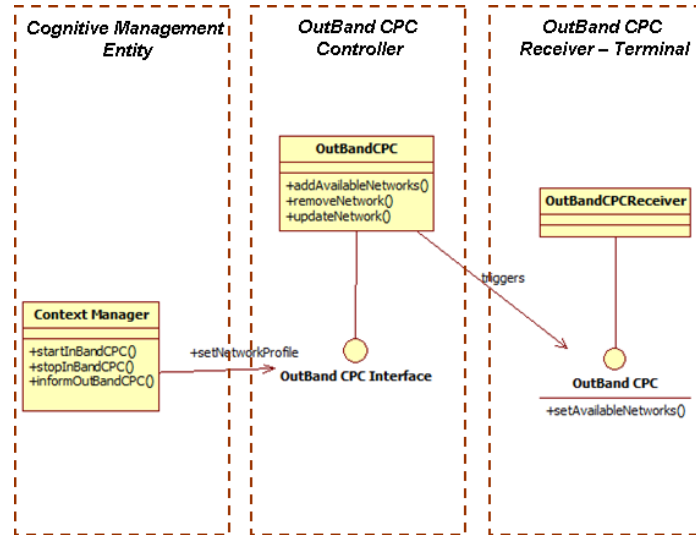


Figure 7: OutBand CPC realization

Experimental Results

In order to prove the viability and suitability of the platform as well as the E3 architecture in the context of future telecommunication networks we carried out a series of experiments. In particular, we tested some key concepts introduced by E3 and report here the obtained results. Due to space limitations, we report only a small subset of the results. The rest can be accessed in numerous project deliverables , .

We validated the following four key concepts of E3: i) connection to CPC, ii) service initiation, iii) ad-hoc cluster formation, iv) cluster handover to a new RAT. We employed the hardware architecture as depicted in Figure 3. The CME and CPC software entities are implemented in Java while all system related functionality (i.e. start/stop network interfaces, monitoring and scanning of interfaces etc) is implemented either through system calls (Ubuntu 9.04) or ANSI C. Message exchanges are implemented through Java Web Services with the use of the JDK internal web server (). Each experiment was repeated a number of times and we report here the exhibited mean value, the standard deviation as well as the significance of the produced results and the implied added value.

The first experiment concerns the connection of a terminal to the Cognitive Pilot Channel. Terminals located in a specific area, connect to the CPC in order to retrieve contextual information. We use a dedicated 802.11 network in order to emulate the CPC and terminals are connected to it after scanning their interfaces and identifying the corresponding ESSID. Figure 8-(a) depicts the produced results. Each time, we measured on the terminal side the time requirements for scanning the wireless interface, identifying the CPC channel and connecting to it. We report a mean value of 731.58ms with a standard deviation of 131.11ms over 100 executions. These results highlight the added value of CPC in the context of heterogeneous networks. Terminals receive contextual information from the CPC instead of scanning their interfaces and thus reduce the consumption of valuable resources such as battery, memory and CPU. Additionally, the cost of connecting to CPC is small, given the fact that it takes place only few times in the daytime of a device.

A second experiment validated the case where a terminal initiates a specific service, from a set of available ones. We used an HTTP video streaming service provided by VLC . The same experiment was executed 20 times. Each time, we measured on the terminal side the time requirements for evaluating the user request in terms of terminal capabilities and service requirements and initializing the connection to the server. We report a mean value of 443.05ms with a standard deviation of 688.98ms. The overall results are presented in Figure 8 (b). Although this experiment does not provide any valuable insight to E3 concepts and techniques it is a necessary step for the validation of the following two cases.

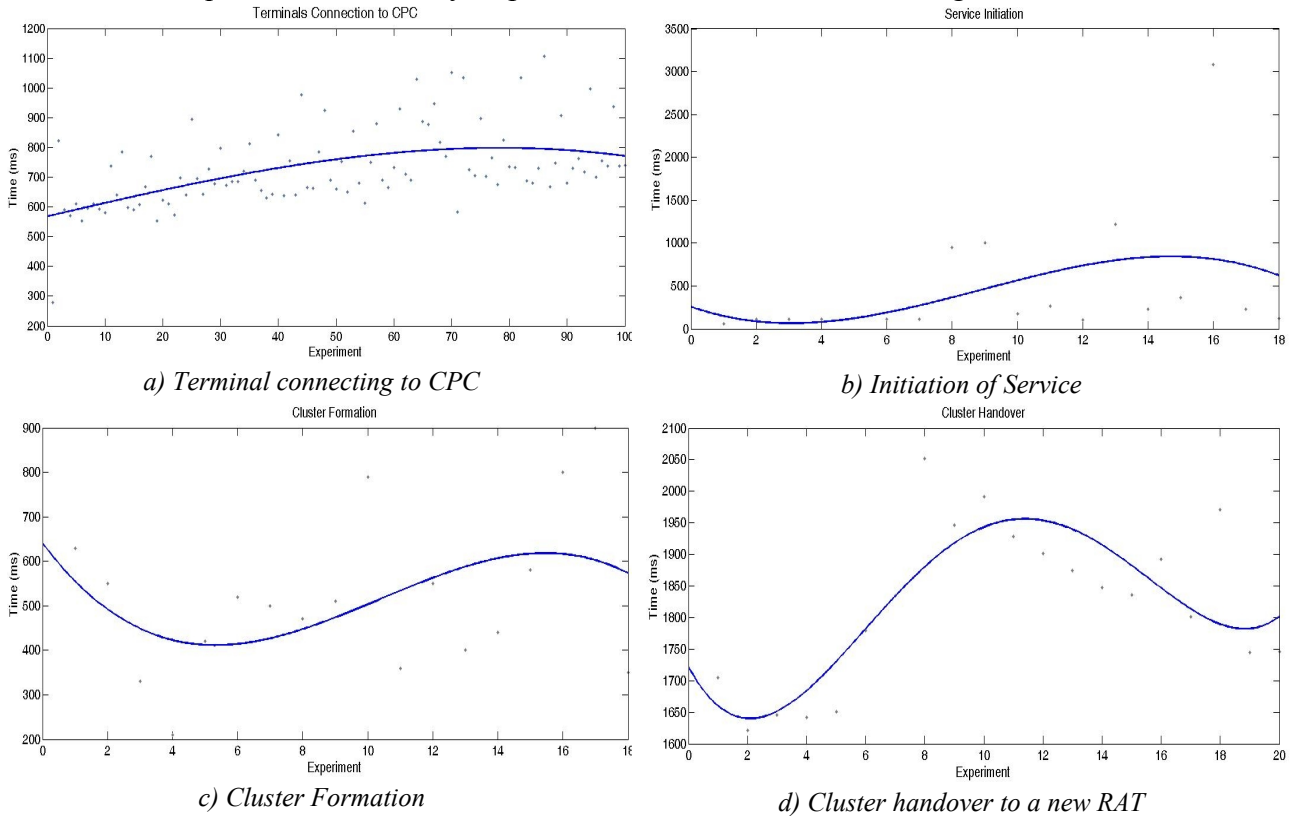


Figure 8: Results for the test cases (a) – (d)

In the third experiment we measured the requirements posed for the formation of an ad-hoc cluster between two mobile devices while in parallel guaranteeing service continuation. The terminals, initially exchange information through the CCR and after its validation decide to form a cluster, where one terminal (usually the one with the highest hardware and software capabilities) acts as a gateway and the other one connects to it so as to gain access to the streaming service. The gateway maintains the connection with the application server and re-transmits the content to the other members of the cluster. Due to space limitations

we omit details related to the decision algorithm as well as the implementation of the CCR protocol. For more details, the reader is encouraged to proceed in . Each time, we measured on the client terminal side the time requirements for performing the afore described operation. We report a mean value of 503.15ms with a standard deviation of 517.20ms over 20 execution. The overall results are presented in Figure 8 (c). The produced results unfortunately exhibit a high variance. The reason for this is actually threefold. The use of HTTP together with Java Web Services and JDK internal web server heavily influences the performance of the procedure since the latter takes place over an ad-hoc WLAN connection between two mobile devices. Therefore, the problem lays in the employed technologies which on one hand enable fast prototyping but on the other are not efficient. Despite this, the time requirements are tolerable and thus can be perceived as a viable case.

In the final experiment we measured the time requirements posed to a gateway terminal in order to handover from one RAT to another while of course maintaining the cluster structure. Each time, we measured on the gateway terminal the time requirements for performing the decision operation and implementing the corresponding action. We report a mean value of 1820ms with a standard deviation of 123.81ms over 20 execution. The overall results are presented in Figure 8 (d). The procedure requires almost two seconds however it has to be taken into account that this is due to the fact that the terminal has to ensure that all devices are informed about the subsequent action and additionally maintain all service streams. The latter is translated to additional computational load and resources utilization thus increased time requirements. Despite this, the success of this test case highlights that in the near future this scenario will probably comprise an integral part of many every day applications.

Conclusions

Based on the functional architecture of the E3 project we presented the implementation and validation of a proof of concept platform for self-x and cognitive functionalities which constitutes an integral part of the E3 prototyping environment. The performance analysis and the experimental results, signify that the platform integrates advanced functions and mechanisms for the support of cognitive and self-x systems, thereby increasing the overall system flexibility and leveraging the E3 approach as an attractive solution for future mobile telecommunication systems.

Acknowledgement

This work was performed in project E3, which has received research funding from the European Community's Seventh Framework programme. 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 E3 consortium is hereby acknowledged

References

- [1]. ICT-2007-216248 E³ Project, <http://www.ict-e3.eu/>
- [2]. 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP): Information Service (IS)".
- [3]. 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP): Requirements".
- [4]. 3GPP TS 32.362: "Entry Point (EP) Integration Reference Point (IRP): Information Service (IS)".
- [5]. "The E3 architecture for future cognitive mobile networks", Alexandros Kaloxylos, [Thomas Rosowski](#), Kostas Tsagkaris, Jens Gebert, Eckard Bogenfeld, Panagis Magdalinos, Aristi Galani, Klaus Nolte, PIMRC 2009
- [6]. E3 Deliverable D6.3 "Final version of the prototyping environment; Scenarios, interfaces for integration of cognitive systems"

- [7]. E3 Deliverable D6.4 “Results from validation of cognitive systems on the E3 prototyping environment”
- [8]. End to end efficiency, <https://www.ict-e3.eu/>
- [9]. <http://java.sun.com>, the source for java developers
- [10]. VLC media player, <http://www.videolan.org/vlc/>