Optimizing the Delivery Chain in Heterogenous Networks

Xavier Sanchez-Loro[12], Josep Paradells[12], Jordi Casademont[12], and José Luís Ferrer[12]*

[1]Wireless Networks Group - Telematics Department, Universitat Politècnica de Catalunya,
Mod. C3 Campus Nord, C/ Jordi Girona 1-3, 08034 Barcelona, Spain
http://wireless.upc.edu/en/index.html
[2]i2CAT Foundation,
Gran Capita 2-4 (Nexus I building), 08034 Barcelona, Spain
{xsanchez,josep.paradells,jordi.casademont}@entel.upc.edu
jlferrer@i2cat.net
http://www.i2cat.net/en

Abstract. The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be written using the *abstract* environment.

Keywords: We would like to encourage you to list your keywords within the abstract section

1 Introduction

In previous work [9, 8, 7, 10], we explored the possibilities of context-awareness, service discovery and negotiation as the basis to deploy ubiquitous services in heterogenous environments. First, we explored the concept of delivery context in the Ubiquitous Web, seeking ways to acquire context data from user devices and enhancing the available mechanisms for negotiation and context acquisition across the delivery chain. This way we worked with proxies to enhance server and client negotiation in heterogeneous web environments, allowing proxies to transparently detect device capabilities and personalizing context distribution and content negotiation according to user preferences, device capabilities and network environment. Furthermore, we explored the possibilities of using intermediaries to optimize web traffic by means of header reduction. One conclusion of this piece of work is that e-2-e principles of current Internet are not valid in heterogeneous environments. With such different underlying computing and networking technologies, we need more intelligence and autonomy in the network and means to adopt different protocol behaviours depending on underlying

* This work is partially funded by MECD and FEDER project TEC2009-11453 and I2Cat Foundation Projects TARIFA. The authors would like to express their thanks for contributions from the TARIFA project and the participating research groups: NETS, MediaEntel, GXO, ANA, Cols and GTM.

network environment and context. Using a model of autonomic proxies/middleboxes performing local adaptations at required network segments provides better results than plain-old e-2-e negotiation across the delivery chain.

All this work give us with enough insight in the field to conclude that Ubiquitous and Pervasive Computing requires more level of control and adaptation granularity than current networking models offer. We cannot made the network more pervasive and invisible at the Application level, indeed we need that the network stack itself becomes ubiquitous and pervasive. Thus, on further efforts [12, 11, 14] we have worked in designing from scratch a semantic context-aware network architecture focused on the provision of ubiquitous services. This Future Internet (FI) architecture expands the concept of delivery context to support any type of service traffic, not just web traffic. So, we propose that intermediate nodes participate in the discovery, negotiation and adaptation of services. Hence, integrating service discovery, negotiation and adaptation as core features of the architecture. Besides, another important divergence of the architecture from TCP/IP stack is its layer-less modular nature, since it is based on SOA principles, and follows the model of RBA instead of OSI's one. Hence we designed a flow-oriented context-aware network architecture where communications are composed on the fly (using reusable components) according to the needs and requirements of the consumed service [1].

1.1 New Networking Models for the Future Internet

The Internet is facing an architectural crisis as the load and stress on the Network increase with new users and applications. Current Internet architecture is becoming more and more ossified and complex; with lots of patches aimed to amend different issues that have arisen during its almost 40 years of existence. This patching has been done outside the core of the architecture, increasing its complexity and blurring -and sometimes even contradicting- the neat principles behind the original design of the Internet. Furthermore, new services and computing paradigms require new modes of interaction, new features (identification, context-awareness, seamless service discovery and composition, etc.) and clean solutions to known issues (mobility, security, flexibility, etc.); but it is not clear how the current Internet architecture will be able to cope with all these new requirements. In the research community, there are two approaches to solve this crisis: evolutionary and disruptive (clean-slate). Most of the initiatives fall in the category of clean-slate redesign, where the network architecture is designed anew from scratch. These initiatives exhibit some clear lines of research and similar approaches for certain issues. The main common lines of research are:

1. **Redefinition of the layering concept and protocol design**. These lines imply a complete redefinition of the network architecture. Most of the initiatives follow this trend in one-way or the other. Some approaches:
   (a) *Questioning of the end-to-end principles of communication.* e-2-e principles are not taken for granted and many initiatives depart from this communication paradigm.

(b) *Changes from host/interface-centric paradigm to information/service centric paradigms.* Two main approaches:
   i. Content-centric architectures
   ii. SOA/RBA architectures

(c) *Protocol modularization and protocol custom composition.* Some common goals
   i. Flexibility: overcome current protocol stacks rigidness in their design and restricted inter-layer communication.
   ii. Adaptability: to changes in communication environments and application requirements, including different nodes, links and applications
   iii. Loose-coupling and evolution: current protocols are ossified by its tight-coupling design, they can't naturally evolve. Protocols need means to evolve by loose-coupling principles.
   iv. Autonomic: protocols (and networks) must be autonomic and self-*.
   v. Custom protocols for dealing with network heterogeneity and meet new application requirements. Two main trends, composed from scratch (e.g. SONATE, TARIFA) and template-based (e.g. Netlets, ANA, SILO)
   vi. Context-awareness. Current protocols are oblivious of context conditions, being unable to adapt or even detect changes in environment conditions.
   vii. Time of composition: run-time or design-time.

2. **Changes in the routing and addressing paradigm**. These initiatives take into account that most of the transactions in the Internet are for accessing to content not to hosts. Although minor in number compared to initiatives following trend 1, a good deal of FI solutions imply changes in the routing paradigm. In some cases, these lines may coexist and/or work with legacy technologies like IP networking. Some approaches:
   (a) *Redefinition of what to address.* From hosts to content and/or services.
      i. Location/Identifier split
      ii. Self-certifying content identifiers
      iii. Content-centric addressing and naming
      iv. Semantic discovery of services and data: service-oriented paradigms usually involve mechanism for discovering and negotiating services. Similarly, content-centric networking needs some way to discover and locate requested content
   (b) *Changes in forwarding paradigms.* Route by data content not by host.
   (c) *Publish/Subscribe paradigm for locating data.*
3. **Network virtualization as a means to deploy multiple concurrent architectures and networks**. Although not a new solution, research on network virtualization is critical to the deployment of FI technologies since with all probability no one architecture will be the final solution and many of them will coexist in the future thanks to virtualization.

Herein, we will focus on proposing a service taxonomy and classification for RBA/SOA architectures aiming to model a common framework for the descrip-

tion and composition of possible Atomic Services (ASs)[1] or networking functions valid for architectures coping with heterogenous environments. The final objective is the fine-tuning the delivery chain to optimize application and delivery performance and behaviour. The paper is structured as follows: first we will introduce the concept of RBA in section 1.2 and how this model is key to understand the philosophy behind some of the most disruptive FI architectures. In section 2 we will analyze the available service definition in the field and propose a "common" service definition, valid for any type of networking service regardless the granularity, thus including support both for Atomic Services (ASs) and Composed Services (CSs). Once we have a clear definition of what a service is, we can propose a taxonomy in section 3 useful for classifying services for composition and allocation purposes in heterogenous environments. With this taxonomy at hand, we propose a service classification in section 4 which is suitable for our needs[2]. In section 5 we discuss some strategies fro the allocation of services along the delivery chain. Finally, section 6 concludes the paper.

1.2 RBA

From an architectural point of view, the most ground-breaking (and influential for our own research) proposal was, and still is, the RBA [2], a non-layered and modular architecture built around the concept of roles. A role represents a communication building block that performs some specific function relevant to forwarding or processing packets. Therefore, new relations amongst network components are established by defining a set of roles to be played by each component involved. In this way, nodes implement the different roles required for a certain communication (i.e., "packet forwarding", "fragmentation", "flow rate control", "byte-stream packetization", "request web page", or "suppress caching"). The protocol stack in this architecture is replaced by a protocol "heap" in which each node's realm of execution in the protocol heap is restricted to the roles it exercises. Non layering thus implies new rules for modularity, header structure and ordering for the processing of metadata, as well as encapsulation. These new rules and relationships were (and still are) not fully investigated and unexpected synergies and incompatibilities may appear in implementing RBA-based architectures. This kind of micro-modularization, breaking existing protocols into small modules with a definite function, is common practice in most clean slate proposals [4, 6, 13, 5]. While they are all aimed at improving flexibility, their key difference is the way in which modularization is approached (in terms of scope, focus, purpose and module granularity) and how network architecture and protocols are abstracted, defined and built on the basis of these modules, that is, how the architecture and communications are composed with these modules. All these synergies between different initiatives concentrate around the concept of

[1] Other terminology for ASs are roles, building blocks, functions...

[2] Any service classification will be subject of controversy and lack of exhaustiveness. We think the proposed classification covers most of the aspects for optimizing service provisioning and connection performance, but it by no means an exhaustive list

service/role. Nevertheless there is no unified or commonly agreed service definition and taxonomy [**?**,**?**]. There are some timid approaches, but none fully developed.

Since RBA proposes a conceptual model, it can be implemented in different terms: disruptive or evolutionary. On one hand, a pure RBA architecture may be implemented with no layers at all and roles scope ranging from application all the way through to physical layers or perhaps just above a virtualized link layer. This would be the disruptive or clean-slate approaches, and it is the main trend for RBA-based architectures. On the other hand, it can be deployed as an evolutionary enhancement on top of the network layer or even just at the application level. The scope and definition of roles will be directly derived from the implementation scenario. Hence, a common definition of basic roles would be useful both for FI architectures and evolutionary solutions applying the concept at the Application layer.

This kind of micro-modularization, breaking existing protocols into small modules with a definite function, is common practice in most of clean slate proposals. They key difference is how this modularization is approached (in terms of scope, focus, purpose and level of granularity) and how network architecture and protocols are abstracted, defined and built on using these modules as basis; that is how the architecture and communications are composed with these modules.

2 Service Definition: What Constitutes a Service?

If we take a look on the literature of service discovery, Service-Oriented Architectures, protocol modularization and service/funtional composition fields, we can found different definitions of what is a service.

- RFC 2165-SLPv1: The service is a process or system providing a facility to the network. The service itself is accessed using a communication mechanism external to the the Service Location Protocol.
- ATIS: Something of value being provided to and consumed by end-users of communications networks. A function that is performed by software whose output is used by some other entity. It may be, for example, a presence engine, or a routing function, or a number translation capability, or protocol interworking. 'Service' may also refer to the software itself.
- W3C
  - Web Service: A service is an abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of providers entities and requesters entities. To be used, a service must be realized by a concrete provider agent.
  - An application that provides computational or informational resources on request. A service may be provided by several physical servers operating as aunit.
- OSAIS: The performance of work (a function) by one for another. Services are the mechanism by which needs and capabilities are brought together.

Broadly speaking, all these definitions are specifically tailored to certain application domains or computing solutions and are usually defined from the point of view of the applications that will consume them. There is no unified definition in the field of what constitutes a service that is generic enough to be usable to different service-centric approaches in different application domains.

3 Service Taxonomy

The following independent categories define a taxonomy valid for different types of services, regardless their granularity (see figure 1). This taxonomy aims to organize and ease role/service definition in SOA/RBA solutions and is inferred from our initial proposal of AS, published in [11]. Figure 1 illustrates graphically this taxonomy.

- **Granularity**: atomic, molecular, composed.
  - *Atomic Services* are the core of our architecture. They are the building blocks (roles) used to establish communications and to deliver data in a self-adaptable, self-configurable and context-aware way. Each atomic service provides one concrete and well-defined networking function (along with the reverse function, if any). Different algorithms and implementations of an atomic service may exist (i.e. different congestion control algorithms), and co-exist in the same node, using attributes to both describe the different possibilities and to tune/configure the atomic service in order to use it to fulfill specific workflows needs.
  - *Molecular Services* are common compositions for small sets of ASs that usually work together to perform a more complex function than just a single ASs (e.g. sequencing and retransmission may be composed as two independent ASs, but they are often composed together in form of a molecular service). Using this intermediate abstractions can ease and simplify the composition process of bigger composed services.
  - *Composed Services* are network applications with a wider scope than just establishing communications (e.g. sensing service, directory service, file transfer, instant messaging, presence, etc.). Each composed service or application implies consuming different atomic and sometimes other composed services, with possible dependences appearing between them. In addition, they can involve one or more nodes, depending on the complexity of the service.
- **Execution/Distribution**:
  - *Isolated*: local execution of the service
  - *Distributed*: execution distributed between two nodes, regardless their location. It includes support for e-2-e, section and hop-by-hop distribution/allocation of services.
- **Scope**: services can be applied with different scopes or considerations depending on the desired result.
  - *Network*: services are executed to optimize communication according network context.

    - *Application*: services are executed to optimize application behaviour and interconnection to meet application requirements according to context characteristics.
- **Usage**: rules governing the service usage.
    - *Mandatory*: usage of this service is mandatory as it i basic for establishing a communication (e.g. forwarding).
    - *Optional*: usage of this service is optional, its usage will depend on application requirements and context characteristics
- **Purpose**: which is the purpose of the service.
    - *Delivery*: to deliver data between two different entities involved in the delivery chain (they can be adjacent or non-adjacent nodes or they can be two end applications, depending on the scope of the service)
    - *Mobility*: services related to application, user and node mobility.
    - *Storage*: services dealing with the storage of data.
    - *Security*: services dealing with security issues.
    - *Data Adaptation*: services dealing with adapting and transforming data for different objectives, interoperability, customization and optimization of data.
    - *Addressing*: services dealing with the identification and labeling of resources.
    - *Management*: services dealing with the management of the different entities in the network (nodes, applications, services, etc.).
    - *Signaling*: services dealing with interchange of signaling and control data.
    - *Presentation*[3]: services dealing with the presentation of contents and user/application interfaces
- **Order**: order/existence of the AS in workflow composition may be dependent of another service.
    - Dependent.
    - Independent.

4 Service Classification

The following tables (see table 1 and table 2) show the different services that can be applied according to different purposes: Application and Network.[4][5]

5 Allocation Strategies

One critical process during the negotiation of a composed service is the composition and allocation of AS to obtain a feasible workflow for each node involved in the communication; one that allows the fulfillment of the service requirements according to the context constraints. Thus, this process must perform the following analysis:

[3] This category is only available for services with application scope.

[4] Each service includes the reverse function if any.

[5] Transcoding is a coupled pair of Decoding/Encoding. Includes any data transformations, i.e. data transcoding, cyphering, compression, etc.

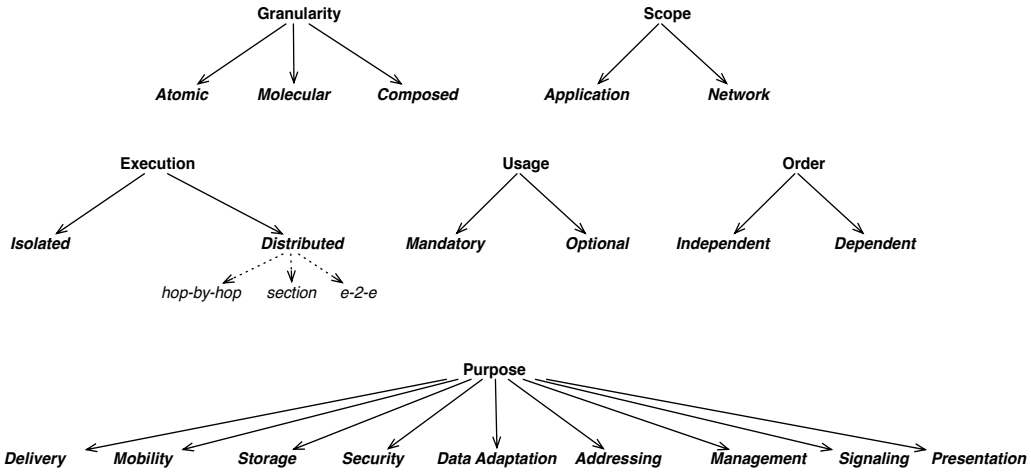

Fig. 1. Service taxonomy

- Analysis of context constraints Vs QoS Requirements. Requirements should be checked against context constraints in order to obtain sets of possible AS.
- Analysis of possible feasible AS configurations for the whole path (not for just isolated individual nodes) to fulfill service requirements with these context constraints. Obviously, this analysis implies knowing where the different AS are allocated in each configuration. For this analysis, we can adopt some techniques proposed in the field (see section 5).

Hence, as we can see, service composition and allocation are two interleaved processes that cannot be feasibly decoupled.

Implementing this process implies knowing two variables for each AS/AM: the cost and the benefit/effect of using it. Costs should be checked against context constraints, whilst effects should achieve QoS requirements. Therefore, metrics should be researched to measure the cost of executing an AS in different terms (e.g. CPU/RAM utilization, affected bandwidth, delay, etc.), although in the end all these metrics should be unified/normalized to use the same units in order to be suitable for expressing a cost function. Even it must be thoroughly investigated, at a first glance expressing costs in terms of delay with time units seems a good starting approach. However, our first impressions is that the benefits of using a certain AS have not a straightforward translation to a common unit, and it should be investigated even if it is possible.

When deciding which is the best workflow configuration for a composed service path, it is needed some tools to analyze the effect and the cost of using each configuration. After some research on the field, we have found some techniques and references that may be suitable for our purposes. This is the case of Wolf's layered graph approach [3], this technique allows to compute the cost of service

Table 1. Service Classification for Application domain.

Purpose	Service
Delivery	Flow control/Rate control
	Error Control
	Object Control
	Flow Multiplex/Joining
	Monitoring
Mobility	Portability
	Nomadism
Storage	Cache
	Directory
	Store
Security	Authorization
	Authentication
	Access Control
	Token Exchange
Data Adaptation	Encoding/Transcoding
Addressing	Labeling
Management	Configuration
Signaling	Session control
	Service Discovery
	Service Announcement
	Negotiation
Presentation	Rendering
	Adaptation
	Encoding/Format
	Content Negotiation
	Interaction (Modality)

allocation inside a network, by transforming it in a shortest-path problem, which can be easily computed using Dijkstra's algorithm. The key idea is to transform the network in a graph with multiple layers, where each layer represents the execution of a service. Then, the optimal path is found executing Dijkstra's algorithm for finding the shortest path in the transformed graph. Finally, the multi-layered graph is mapped back to the original network. This approach has some real good pros and some cons we discuss:

- ***It is suitable for computing the placement of multiple services***. But services are allocated in sequential order, thus not suitable for parallel service execution. In our virtual circuit model, for the purposes of allocation, services may be deployed sequentially.
- ***Session-oriented***. The algorithm assumes that the intermediate nodes and the network links used for communication are selected. This is the case of our architecture, since we use a virtual circuit model with a fixed path.
- ***Bandwidth alteration awareness***. The algorithm can be extended to be aware of the influence on bandwidth of using one service. For instance some

Table 2. Service Classification for Network domain

Purpose	Service
Delivery	Congestion control
	Flow control/Rate control
	Forwarding
	Joining/Multiplex
	Forking/Splitting
	Aggregation
	Framing/Encapsulation
	Fragmentation
	Medium Access Control
	Acknowledgement
	Retransmission
	Sequencing
	Monitoring
	Negotiation
Mobility	Inter-domain
	Intra-domain
	Handover
	Point of Attachment (Multi-path)
Storage	Cache
	Buffering
Security	Authorization
	Authentication
	Access Control
	Token Exchange
Data Adaptation	Encoding/Transcoding
	Rate adaptation
	Channel adaptation
Addressing	Labeling (flow, packets, etc.)
Management	Configuration
	Network management
	Node management
Signaling	Session control
	Route discovery
	Route setup
	Circuit setup
	Session Negotiation

services like compression have a cost, but decrease the required bandwidth. In the opposite site, FEC increases overhead and, thus, the required bandwidth. This effect is reflected with bandwidth scaling factor. It should be investigated, but probably this extension of the algorithm (or a very similar one) may be applied for services affecting other network metrics like delay, probability error, etc.

- ***Optional processing***. The algorithm can be used to incorporate the possibility of optional services, services that are not necessary for the correct data communication but can enhance the quality of the connection.
- ***Multicast sessions***. Multicast sessions may also be computed using this approach, but instead of a shortest-path problem, the resulting problem is a standard Steiner tree problem.
- ***Capacity constraints***. Although taking into account capacity constraints is clearly a NP-hard Hamiltonian path problem, which cannot be assured to find an efficient solution, least the lowest-cost path; the authors have proposed modification to Dijkstra's algorithm to incorporate capacity constraints, so that in most situations they can find low cost session configurations with the required resources.
- ***Decided and ordered set of services***. In this technique, services to be placed are decided beforehand. For purposes of adapting it to our proposal, we must decide first which sets of AS will be tested, so we can check the feasibility of a certain set of AS; but the algorithm cannot decide for us which are the services to be allocated.
- ***Knowing costs and benefits***. In order to be able to use this technique, we must envision a method to obtain the correct values for costs and effects of using the services. However, this is not a problem of this technique alone, whatever method we use to compute the allocation of services will require to know the cost and the benefit of using a service.

It seems quite obvious that this technique can help to compute the cost of a service allocation with a reasonable computation cost, but it is not straightforward how to use it for expressing effects. It must be investigated but one possibility to be researched is integrating the effects of services as a means to modify the cost, and thus, just checking if the modified cost complies with the context constraints and the service requirements. This can be feasible for those services where the cost and the benefit can be easily related; but this is not always the case (e.g. FEC increases delay and required bandwidth, but decreases PER). For those metrics that cannot be incorporated in the graph as a scaling factor, the benefit may be calculated with a new layered-graph where the benefit is computed as a cost, but inverted in order to keep the problem as a Dijkstra's shortest-path problem.

Some of the shortcomings of these approaches for our case study is that they cannot take into account capacity constrains [28]; the exact services that will be used and the ordering of execution of these services are known before-hand, since they work with high-level and application level services and not with basic network services like ASs. So, in order to make these solutions work with our proposal we must devise ways to:

1. Compute the cost of using each AS in terms of computing and network usage
2. Decide what are the most appropriate ASs (and their configuration) to obtain a certain performance and behavior. This implies finding ways to infer the benefits of using a certain service. This may be a simple problem when dealing with independent services, but it can become complex when working

with inter-dependent services (i.e. ACK and CRC). Furthermore, many ASs imply allocating two services, one at the input node and one at the output node, which further complicates the problem.
3. Determine the correct ordering of execution of ASs.

We are currently working on simple scenarios with a small subset of ASs, researching for interdependences and the cost and the benefit of using these services, both alone and in combination. We expect to obtain results in the following months.

6 Conclusion

TODO

Acknowledgments. This work is partially funded by MECD and FEDER project TEC2009-11453 and I2Cat Foundation Project TARIFA. The authors would like to express their thanks for contributions from the TARIFA project[1] and the participating research groups: NETS, MediaEntel, GXO, ANA, Cols and GMT.

References

1. TARIFA: The Atomic Redesign of the Internet Future Architecture, `http://www.i2cat.net/en/projecte/tarifa-1`
2. Braden, R., Faber, T., Handley, M.: From protocol stack to protocol heap: role-based architecture. SIGCOMM Comput. Commun. Rev. 33, 17–22 (January 2003), `http://doi.acm.org/10.1145/774763.774765`
3. Choi, S., Turner, J., Wolf, T.: Configuring sessions in programmable networks. Comput. Netw. 41, 269–284 (February 2003)
4. Dutta, R., Rouskas, G.N., Baldine, I., Bragg, A., Stevenson, D.: The SILO architecture for services integration, control, and optimization for the future internet. In: IEEE ICC. pp. 24–27 (2007)
5. Jelger, C., Tschudin, C., Schmid, S., Leduc, G.: Basic Abstractions for an Autonomic Network Architecture. A World of Wireless, Mobile and Multimedia Networks, International Symposium on 0, 1–6 (2007)
6. Müller, P., Reuther, B.: Future Internet Architecture - A Service Oriented Approach (Future Internet Architecture - Ein serviceorientierter Ansatz). it - Information Technology 50(6), 383–389 (2008)
7. Sanchez-Loro, X., Beltran, V., Casademont, J., Catalan, M.: Ubiquitous web access and application layer optimization: Dynamic content negotiation over cellular links. Grid and Pervasive Computing, International Conference on 0, 269–274 (2008)
8. Sanchez-Loro, X., Beltran, V., Casademont, J., Catalan, M.: Ubiquitous web access: Collaborative optimization and dynamic content negotiation. Multimedia and Ubiquitous Engineering, International Conference on 0, 558–563 (2008)

9. Sanchez-Loro, X., Casademont, J., Ferrer, J.L., Beltran, V., Catalan, M., Paradells, J.: Dynamic content negotiation in web environments. In: Context-Aware Computing and Self-Managing Systems. Chapman and Hall/CRC Studies in Informatics Series (March 25, 2009), http://www.crcpress.com/product/isbn/9781420077711
10. Sanchez-Loro, X., Casademont, J., Ferrer, J.L., Paradells, J.: A proxy-based solution for device capabilities detection. In: IASTED European Conference on Proceedings of the IASTED European Conference: internet and multimedia systems and applications. pp. 28–34. ACTA Press, Anaheim, CA, USA (2007)
11. Sanchez-Loro, X., Ferrer, J.L., Casademont, J., Paradells, J., Vidal, A.: Proposal of a Clean Slate Network Architecture for Ubiquitous Services Provisioning. In: IEEE First International Conference on Future Information Networks ICFIN 09) (14-17 October 2009)
12. Sanchez-Loro, X., Ferrer, J.L., Gomez, C., Casademont, J., Paradells, J.: Can Future Internet be based on constrained networks design principles? Computer Networks In Press, Corrected Proof, – (2010), http://www.sciencedirect.com/science/article/B6VRG-51S25P9-5/2/ad39433134291a948e7f5555d583d3df
13. Touch, J., Wang, Y., Pingali, V.: A Recursive Network Architecture. In: ISI Technical Report ISI-TR-2006-626 (October 2006)
14. X.Sanchez-Loro, Gonzalez, A., de Pozuelo., R.M.: A Semantic Context-Aware Network Architecture. In: Future Network and Mobile Summit 2010. Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation (Florence, Italy, June 2010)