CASCADAS – Autonomic communication and situated pervasive services

BORBÁLA KATALIN BENKŐ, TAMÁS KATONA, RÓBERT SCHULCZ

Budapest University of Technology and Economics, Department of Telecommunications
{bbenko,tkatona,schulcz}@hit.bme.hu

Keywords: autonomic communication, ACE, pervasive services, self-organization, knowledge network, pervasive supervision

CASCADAS (Component-ware for Autonomic, Situation-aware Communications, And Dynamically Adaptable Services) is one of the four 6th Framework IST-FET projects that aim at providing both theoretical and practical background for a new generation of complex, distributed, pervasive services. The basic building block of the situation-aware, self-organizing, autonomic communication based network is a common abstraction called ACE. ACEs provide and use services, adapt to the situation, create and manage plans, build up knowledge networks, and move and self-organize based on their own autonomic decisions. In CASCADAS, we plan not to categorically rely on the available Layer3 facilities so that to enable the possibility of emerging new-generation protocols/technologies such as utilizing the physical proximity or CPNs. **(In: 2006/12, pp.23–28.)**

1. Introduction

CASCADAS [1] is one of the four EU 6th Framework IST-FET projects (ANA, HAGGLE, Bionets, CASCADAS) that aim at researching situated, autonomic technologies from different viewpoints [2]. The goal of CASCADAS is to reduce the costs (management, communication, development, configuration etc.) of future emerging complex, highly distributed, pervasive services through a self-organizing network managing knowledge and adapting to the situation.

CASCADAS goals can be formulated on different levels. On the theoretical level, we aim to develop a common abstraction (ACE, meaning Autonomic Communication Element), and based on it, identify/provide models, algorithms and general principles in the fields of self-organization, knowledge network, pervasive supervision and security. In practice, we prove the feasibility/operability of the theoretical models employing demonstrational applications. A secondary goal is to elaborate towards new results in the field of communication. One idea is to work out a new communication protocol that also makes use of the physical proximity in the network, running directly above Layer2.[1] Another idea is that ACEs are running over CPN (Conceptual Packet Network) [3], making use of the self-organization and optimization possibilities provided by the CPN.

2. The CASCADAS vision

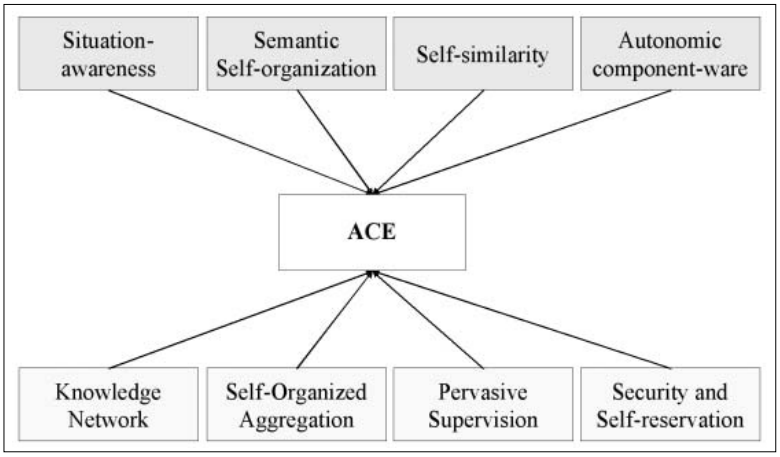
This section elaborates about the motivations and goals of the projects, giving detailed introduction/summary about background principles and the meaning of the keywords.

2.1. Vision

In the CASCADAS vision, the world is proceeding towards pervasive, situation-aware services; the project's goal is to explore emerging problems, elaborate models and provide solutions [4]. The basis of the abstraction is a common lightweight model, the ACE. Future services are envisioned to be available through ACEs *(Figure 1)*. ACEs solve communication and management problems in an automatic and autonomic way, seeking for a kind of optimality.

– ACEs perceive and organize knowledge about the situation in order to understand it, including physical, technological, social, user-specific and problem-specific aspects.
– ACEs are capable of self-configuration and self-adaptation. They modify themselves and re-parameterize the services provided, in order to adapt to the situation.

Figure 1.
Principles and tools in the CASCADAS vision



1 *Running over Layer2 – besides being an interesting research field – is also a practical consideration, since many concerning operating systems (e.g. TinyOS) support only Layer2.)*

– ACEs make up into self-organized structures (e.g. in order to optimize the service or to create a new composed service).
– ACEs have self-* properties (self-healing, self-similarity, self-configuration, etc.)

The project examines autonomic self-organizing communication elements from several viewpoints:

(1) *The ACE model.* Defining a common abstraction, modelling, creating a framework.
(2) *Semantic self-organization.* Composition of ACEs, mobility.
(3) *Knowledge networks.* Creation of knowledge network and knowledge management.
(4) *Pervasive supervision.* Observing the procedures on the distributed system, and interventions when needed (e.g. self-healing).
(5) *Security.* Besides classical tasks (authorization, right management, cryptography), reputation based trust models.

2.2. Keywords and the meaning behind

CASCADAS – just like the 3 sister projects – uses several keywords and principles that may not be commonly known in the telecom sector. Many of them have similar, partially overlapping meanings, and also there are some which are just new titles for long-known concepts.

Autonomic

The word "autonomic" gained wide attention first in 2001, as IBM started the "autonomic computing" initiative, aiming to create a self-managing artificial system, similar to the human autonomic nervous system. IBM had identified four (plus one) functional elements that are vital for autonomic operation:

– Self-Configuration: automatic configuration of components.
– Self-Healing: automatic discovery, and correction of faults.
– Self-Optimization: automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements.
– Self-Protection: proactive identification and protection from arbitrary attacks.

There is a difference between autonomic and automatic systems. "Automatic" means that the process is executed by itself, without external intervention. Autonomic means more: besides being of course automatic, it also shows self-* aspects (a kind of intelligence).

CASCADAS ACE is an autonomic component, having all four necessary properties.

Autonomic communication

Autonomic communication means that even the communication has self-* properties. First, the communication technology should be fault-tolerant and self-optimizing; on the other hand, communication parties themselves are also making decisions in an autonomic way (message sending and receiving, interpreting the message, and the reaction to it).

Having a more abstract look at the problem, it can be said, that the goal of the IBM autonomic initiative was to adapt the system to a more or less known environment. In case of autonomic communication, the environment is highly variant, but through talking to each other, it is possible to get known with it, moreover, also to affect it on a certain level. Autonomic communication also means that the intelligence of the system is not centrally located, but is spread over the components.

Autonomic communication is the basic principle of CASCADAS.

Pervasive services

Pervasive (ubiquitous, everywhere) service means that the source of the service is rather the environment than a distinct computer. This can be interpreted in several ways. It may mean semantic labelling, a locality concept (that the service is only available for nearby clients), intensive logical mobility (the service is moving freely over the network to find the best environment), or that the environment re-organizes itself according to the actual needs (e.g. creates new instances of popular services). Another interpretation is that services simply surround the user (even the toaster is regarded as a computer).

Typically, pervasiveness is supplemented with intelligent aspects such as self-organization or autonomy.

CASCADAS focuses on pervasive services; it's an important element of the vision.

Situation-aware, situated

Situation-awareness (context-awareness, environment-awareness) is a kind of re-consideration of principles that are present in numerous fields (e.g. agent models, control theory). The meaning is that the element observes the context, and reacts accordingly by a four-step process: observation of the context, interpretation (understanding) of the acquired information, calculating the reaction, and response/intervention.

Soon after the introduction of context-awareness models, two problems were identified. Communication boom means that increasing the size of the system, the number of propagated messages (context/state information) increase exponentially; resulting that protocols that worked well for a small system are often not applicable for a real-life, large system. The other problem is about the understanding: can we assume a common ontology being present in the background that guarantees that each ACE will understand the received message correctly (in case it doesn't drop it as unknown)? A possible solution is to have a common environment model that includes the message ontology (which may change or refine in a flexible way with the environment model).

ACE is a situation-aware component.

Locality concept

The concept of locality is closely related to pervasiveness. Locality means that – due to self-organization, mobility, etc. – the service provider and its user are near to each other in the system, so as a result, only local communication is needed (messages are to be propagated to nearby parts of the system as those interested in it are there anyway).

Another interpretation is that the value of an information atom is the highest around its origin place, as we go farther in terms of time and space, its importance/accuracy/correctness decreases (for example, a statement like "it's 5 o'clock" will be less and less accurate as time goes on, and the truth of "I'm in London" decreases as I'm flying back to Hungary).

In pervasive systems, the locality concept – meaning either logical or physical locations – is intrinsic, as the source of the service is in the direct environment of the user. In CASCADAS, locality concept is intrinsic in the service access model (pervasive services), besides, the knowledge network also supports a kind of local behaviour.

Mobility

In the CASCADAS vision, mobility is a basic property of the ACE. In order to avoid misunderstandings, by "mobility" we do not mean physical mobility (that the ACE leaves the WLAN covered area) but logical mobility (that the ACE is able to move freely among the places of the network that are able to accept it).

3. Tools

Let us discuss in more detail, how CASCADAS is aiming to realize its goals. The tools are: knowledge network building, self-organization, pervasive supervision and a distributed security system.

3.1. Knowledge network

Approaching the problem from the low level, the ACE acquires environment-descriptive information from the knowledge network. But the knowledge network is more than a pure information store or environment abstraction, it is able to organize the knowledge and to optimize itself. To follow the self-similarity paradigm, the knowledge network is built up from the ACEs; and if an ACE has any information that is worth to put into the knowledge network, it can place it there.

Knowledge network supports the concept of locality, so that stored information is important first of all for the neighbourhood, and its value/usefulness decreases with time/distance. Locality can be of logical or physical nature. The difference comes to the surface when ACEs are moving: in physical mobility, the originating location is important, while in logical mobility it is the originator ACE that counts. So, in logical mobility, the information should follow the moving ACE (which is the easiest to implement if the ACE stores the concerning information in itself).

Knowledge network also supports non-local, self-organizing operation, which is drafted through hierarchical, self-organizing overlays. As for now, intra-overlay communication is local, and non-local communication can be achieved via inter-overlay flows.

The knowledge network consists of knowledge atoms which are the basic building blocks of the knowledge self-organization.

3.2. Self-organization

Self-organization may have several goals, e.g. to help the service and the client to find each other (moving them physically closer), to increase service quality (e.g. by replication of the service, by grouping similar services and using load balancing), or to create new complex services. The basic operation of self-organization is the aggregation which may be weak or strong. Strong aggregation is exclusive (the component is forbidden to take part in other compositions when it participates in a strong composition).

Self-organization may result in structures that are technically overlays. The big difference to common overlay networks is that in the CASCADAS ACE network, there are autonomic elements in the higher levels as well (and as such elements, they're not guaranteed to work always deterministically from the external point of view).

3.3. Pervasive supervision

Pervasive supervision assures self-healing and self-optimization abilities. The supervisor authority observes the context and the operation of the system (or rather just those ACEs which agreed to be supervised), looking for errors, misbehaviours or processes that can be optimized. If it is needed, the supervisor can also interfere: it may instruct to review a possibly wrong self-organization, or ask an ACE to move, or initiate the healing of a faulty environment-model etc. Supervision is based on a contract, where the supervised ACE binds itself to unconditionally execute the supervisor's commands (so in some way, its autonomy is limited in the supervised period). Why do we call the supervision "pervasive"? It is, because the pervasive is ubiquitous, it's present on all levels at the same time (network, communication, content, self-organization, social aspects, etc.).

From the theoretical point of view, supervision is one of the most important tools; it enables ACEs to get out from a quasi stochastically self-organized, error sensitive, somewhat inflexible state and gain chance for self-healing and self-optimization.

3.4. Security

The primary goal of the security subsystem in CASCADAS is to add protection to the system (authentication, authorization, message integrity, DoS protection).

As secondary goal, it contributes to the ACE environment model, through a reputation based trust model. A reputation registry is maintained about the past of ACEs. Knowing the past may help in the current cooperation (good reputation means trust, bad reputation warns).

4. The ACE model

ACE is the common abstraction which the four tools are organized around. ACE is a component model and its mapping to an architecture (architecture is not covered in this paper).

4.1. The basic ACE model

ACE can be examined on different abstraction levels. Let's start from the abstract ones and move towards the direct models [5].

Common part, specific part

CASCADAS ACE consists of two parts: a Common part and a Specific part *(Figure 2)*. The Common part contains those functionalities that are present in all ACE instances. The Specific part contains everything behind that, e.g. service providing ability. The functionality of the Specific part can be explored through communication with the Common part.

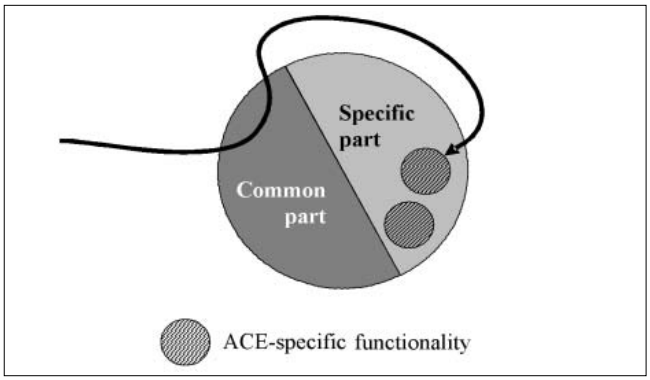


Figure 2. Two parts of the ACE

How does an ACE work?

The operation of ACE is based on its two models: the Self-Model and the Environment Model *(Figure 3)*. Self-Model describes its own operation and goals; while Environment Model models the environment. Both can vary in time: adapt, refine, and get reviewed. As the ACE knows its own possibilities and goal, it is able to create a plan (or more than one plans and choose the best one) and behave as it prescribes. Actions may be reactive (answer for an incoming request/signal) or proactive (there is no external trigger to it). The internal intelligent part of the ACE determines the actions based on the Environment Model and the Self Model (which – besides message sending – can also be the review/altering of a model).

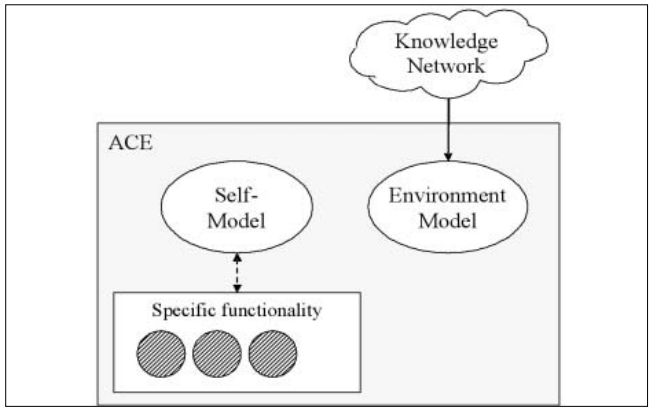


Figure 3.
ACE operates based on the Self- and Environment model

Conceptual model

The conceptual ACE model summarizes the background as a set of short, clear statements (as visualized in an UML diagram as well), see *Figure 4*.

- ACE provides service towards other ACEs.
- Each ACE resides on one location at the same time (of course, the location may change as the ACE moves).
- ACEs use message based communication. Although the communication is theoretically a 3-step process: discovery (the parties locate each other), contracting (agreeing on the interaction conditions), interaction. In simple cases the steps can be implicit and overlapping (e.g. in case of a broadcasted question and a returned answer, we can say that the question contained a default contract which was accepted as the responder returned the answer), in complex cases, phases may become explicit. There may be intrinsic contracts that are valid from the time of instantiation (e.g. allowing for the usage of SEE ACE services, see later).
- The ACE has two models: a Self-Model and an Environment Model.
- ACE is self-similar in the meaning of a possible aggregation (the aggregate is an ACE, too).
- The ACE creates and manages plans in order to realize its goals.

4.2. Execution Environment

For security and other considerations, a special ACE type was defined: the SEE ACE (Service Execution Environment ACE). The SEE is obligatory to be the first ACE on the location (e.g. on the computer); it is instantiated explicitly, and unable to move. All non-SEE ACEs are free to move with the limitation that on the destination location there must be already at least one ACE (which condition is trivially satisfied by the SEE ACE). So, in other words, ACEs can freely move amongst SEEs.

As it is guaranteed to have an SEE ACE on each location where ACEs may occur, there's a possibility to place a kind of "platform functionality" in the specific

part of it. The mobile ACE first explores the new SEE, and then uses the platform functionality through it.

4.3. The ACE model and the CASCADAS tasks

Knowledge network

The ACE model gains information about its environment (context) through the knowledge network (KN), this is the source of the environment model. KN may answer concrete questions or may provide a subscription based notification service.

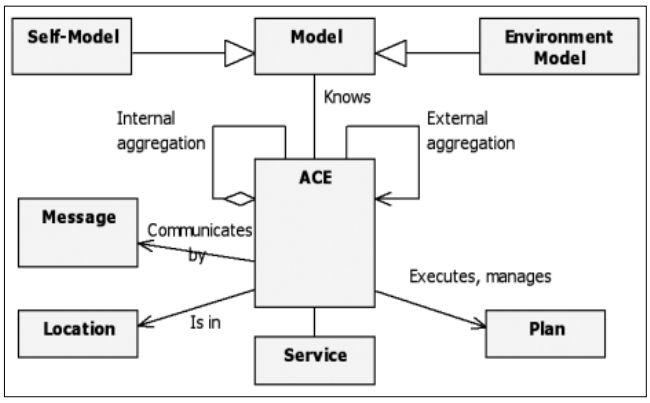
From the technical point of view, two models are possible for the relationship of ACEs and the KN: either all ACEs belong to the KN, or there are ACEs that are outside of the KN (but may use it). Our current opinion is that the obligatory KN membership could make the component model too heavy (while a lightweight model is intended). So, in our actual model, the specific part of KN-member ACEs contains the functionality to put knowledge into the KN, to organize it, and to query and delete information.

Self-similarity

ACEs are able to create weak and strong aggregations. In case of weak (or lazy) aggregation, the cooperating parties don't stop autonomous elements, they just bound themselves to a kind of cooperation. According to the cooperation contract, parties may have confidential information about each other (e.g. they know the abstraction of the others' Self-Models), in order to achieve a more successful cooperation. The same ACE can participate in more than one weak aggregation at the same time.

In case of strong aggregation, one can differentiate between the container ACE and the contained ACEs. The concept of the cooperation is that the container ACE has full access rights and control on the contained elements; it is able to access the specific parts of them. Of course, in order to make use of the contained specific functionalities, the contained Self-Models (and Environment Models) need to be integrated into the container ACE. Please note that in case of strong cooperation, the autonomicity of the contained ACEs is basically lost, as all decisions are made by the container, and the contained ACEs are not directly accessible anymore. On the other hand, strong aggregation makes it possible to achieve formerly unreachable things, e.g. the container ACE can freely combine the contained specific part functionalities. Participating in a strong aggregation requires exclusivity.

Figure 4. Conceptual ACE model



4.4. Pervasive supervision

Supervision needs to access more information than any other member of the system. It may monitor everything, not only messages, but also the internal parts and processes of the ACE, the Self-Model, the Environment Model, the flow of decision making, and the interaction (output). The supervised ACE obliges itself to make the concerning information available for the supervisor. Theoretically, supervision is able to supervise the common part only, as there are no preliminary assumptions on the specific part (it may be anything: a Prolog engine, some machine level code, or even a disguised human future teller). So, the specific part can be monitored through the input/output channels only, and the observed things can be compared with the abstract description of the functionality (that is part of the Self-Model). It is also possible that the ACE doesn't publish its complete self-model to the supervisor, but only an abstraction of it. In this case, the possibilities of the supervisor are limited (a deterministic error may be observed as non-deterministic[2]).

Naturally, in order to assure self-similarity, the supervisor is also an ACE – it just meets stricter requirements than a "normal" one (e.g. security).

5. Sample scenarios

Results will be demonstrated through sample scenarios.

5.1. Pervasive content sharing

Pervasive content sharing consists of several subscenarios: pervasive advertisement, friend search and a pervasive (tourist/museum) information system.

Maybe the most interesting one – and the most different from other project – is the pervasive advertisement application scenario. ACE-controlled, adaptive advertisement surfaces are spread over the city (e.g. displays), that observe nearby people's preferences (e.g. based on the user preference descriptor ACE running on the mobile phone), and display the most fitting advertisement on the surface.

For example, for a group of young people, an ad of a rock concert is displayed; while in front of managers, the poster of the newest gold watch is shown. Ethic

2 Let's take a very simple example of cooperation. If John gives Jill an apple, Jill will be happy; but if in the meanwhile, John pulls Jill's hair, Jill won't be happy. If the abstract model doesn't contain information about the way of delivering the apple (with or without pulling the hair) the supervisor won't be able to find out the cause of the error (unhappiness).

problems should be taken into account as well (e.g. even if the majority prefers the trailer of a new horror movie it mustn't be displayed in the case even one young kid is present).

This scenario is intended to demonstrate pervasive, situation-aware elements.

5.2. Distributed auctions

In the distributed auction scenario, users intend buying or selling goods, through creating and parameterizing buyer/seller ACEs and sending them out to the network. As ACEs move on over the network, they can join auctions. As the system assumes to have a very big number of participants, auctions are always operated locally (only nearby elements can participate, and only local communication is needed). So, the goal of an ACE is to get to the most optimal location, both from the network point of view (fast network connection, small communication delay) and content point of view (near to the semantically matching partners).

This scenario is to demonstrate self-organization, the usage of the KN (for self-organization or just to look up how much did a product cost last time), the utilization of the reputation information ("is the partner reliable?", "has it ever cheated?"), and pervasive supervision, accordingly.

6. Towards a new communication model

As the ACE communication is under development, this section gives only a small insight to the ongoing work.

The ACE communication is message-based and ACEs know about themselves which message types they understand. There are common message types understood by all ACEs (e.g. service discovery message, heartbeat towards the supervisor).

At least the following addressing schemes are considered:

(1) *Broadcasting.* The message is addressed to everyone/anyone. The trick in the propagation of such a message is that as the propagation is carried out by ACEs – so autonomic elements –, it is possible that the message won't really reach all network members (e.g. in order to avoid the flood overloading).

(2) *Recipient(s).* The message is addressed to those ACEs where the ACE ID matches with at least one of the recipient IDs. IDs are not required to be unique, so it is possible to use group addressing or property based addressing (if ID is a set of properties).

(3) *Nearby.* The message is propagated to nearby ACEs only (direct neighbours or a few hops away). The sender doesn't need to know the recipients (not even via properties), they are specified through the structure of the network. This addressing scheme has interesting side effects: e.g. if ACE X sends out a message to the nearby ACEs, and one neighbour (ACE Y) replies with a nearby type message, the recipients of the reply may differ from the recipients of the original message. A possible solution is to specify the centre (e.g. nearby(ACE X)). The nearby addressing significantly differs from usual "IP world" addressing schemes and it seems to fit well to the requirements of pervasiveness.

(4) *OneOf(list).* The message should be delivered to at least one of the recipients.

7. Summary

This paper had two goals: to promote the CASCADAS project; and, through this project, to give a draft introduction to today's important concepts, principles and keywords in the field of ambient intelligence.

We gave a general summary about the motivations and goals of the CASCADAS project; and besides discussing the general project vision, we also offered insight into the ongoing work (conceptual ACE model, demonstration scenarios).

What is ambient intelligence in CASCADAS? The project goal is to provide a general ambient intelligence model: there are intelligent elements at all points of the network (even at higher levels) resulting in a fully distributed intelligent system. The system is lead by the autonomic decisions, cooperation and aggregation of intelligent elements; resulting in a multi-level autonomic system with self-healing, self-optimizing and self-configuring abilities.

Acknowledgment

Besides all CASCADAS project members we would like to express our personal thanks to Edzard Hoefig and Fabrice Saffre for the discussions that helped us to better understand the project.

References

[1] CASCADAS website: http://www.cascadas-project.org
[2] F. Sestini,
Situated and Autonomic Communication an EC FET European initiative, ACM SIGCOMM Comp. Com. Rev., Vol. 36, Issue 2, pp.17–20., April 2006.
[3] E. Gelenbe, R. Lent, A. Montuori, Z. Xu,
Cognitive packet networks: QoS and performance.
In Proc. of the IEEE MASCOTS Conference, Ft. Worth, Opening Keynote Paper, pp.3–12., October 2002.
[4] A. Manzalini, F. Zambonelli,
Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision,
In Proc. of IEEE Workshop on Distributed Intelligent Systems, Prague, 2006.
[5] E. Hoefig, B. Wuest, B. K. Benko, A. Mannella, M. Mamei, E. Di Nitto,
On Concepts for Autonomic Communication Elements,
In Proc. of IEEE International Workshop on Modelling Autonomic Communications Environments, Dublin, 2006.