ARCHITECTURAL CHALLENGES FOR A DEPENDABLE INFORMATION SOCIETY

Luca Simoncini
University of Pisa and PDCC, Italy
luca.simoncini@iet.unipi.it

Felicita Di Giandomenico
ISTI-CNR and PDCC, Pisa, Italy
felicita.digiandomenico@isti.cnr.it

Andrea Bondavalli
University of Florence and PDCC, Italy
a.bondavalli@dsi.unifi.it

Silvano Chiaradonna
ISTI-CNR and PDCC, Pisa, Italy
silvano.chiaradonna@isti.cnr.it

Abstract This paper is two-fold. In the first part it tries to raise awareness on the level of complexity of future computer-based interconnected systems/infrastructures, at least as they are envisioned, and on the level of dependability we are today able to justify with confidence. It tries to motivate that fundamental methods and methodologies must be reconsidered, studied, exploited, assessed and applied to move towards an utopia that can be called "ambient dependability", a global view of the concept of dependability [Laprie, 1992], which encompasses not only the technological aspects but includes inter and multi disciplinary fields, which span over ergonomics, usability, education, sociology, law and government. The second part of the paper provides the authors views, based on their experience, on future directions and architectural challenges to be tackled for approaching, as a first step towards ambient dependability, at least an Information Society which we can depend on.

Introduction

Our society is heavily dependent on computerized interconnected systems and services for which the computer plays a central role in controlling communications, databases and infrastructures. In addition, the use of PCs, home appliances, PDAs, wireless phones and all sort of everyday life objects increases of order of magnitudes the number of networked users, who want to use such objects with low, if any, knowledge of the technicalities behind or embedded in them, and worse by blindly relying on the myth of "infallibility" associated to computers and services controlled by computers [Ducatel et al., 2001; ISTAG, 2002; AMSD, 2003].

In Europe, starting with the March 2000 Lisbon Summit [European Presidency, 2000], EU is stressing the strategic relevance of Europe being the most advanced part of the world based on ICT, by making many sensitive components of our society (finance, banking, insurance, health, commerce, business, government, etc.) dependent on computers, computer-controlled services, networks or infrastructures. The statement that European citizens will be able to rely and depend on "Ambient Intelligence" by using dependable computers or computerized systems is true only for a very limited part. Instead it must be clear that an ICT-based society will need a very large societal reorganization, which should be able to manage the global nature of the envisioned services and infrastructures, which are not limited by national borders or legislations.

Despite significant advances have been achieved in the topic of dependable computing systems over the past ten years or so much further research is required when considering the future landscape. In February 2001 the IST Advisory Group published a number of scenarios for Ambient Intelligence (AmI) in 2010 [Ducatel et al., 2001]. In AmI space, and in these scenarios in particular, dependability in general, and privacy and security in particular, emerge as central socio-technical challenges to be addressed. Devices are ubiquitous, and interact with pervasive networked infrastructures; information assets flow over open information and communications infrastructures, exposed to accidental and deliberate threats. The ISTAG scenarios envision new kinds of human relationships and interactions, and powerful technologies to support them. Important perspectives on the dependability of socio-technical systems arise from detailed ethnographic and human-computer interaction analyses of their actual use.

For sake of understanding, one of the ISTAG scenarios is described in the Appendix, with the discussion on dependability implications reported in the AMSD Roadmap [AMSD, 2003]. This description tries to highlight the threats to AmI scenarios, thus pointing out that research in dependability beyond the current state-of-the-art is essential.

The way towards the landscape envisioned by the ISTAG scenarios has to cope with the actual evolution towards more complex services and infrastructures. They are usually based on the layering of different systems (legacy systems), designed in different times, with different technologies and components and difficult to integrate, and a dependable ICT-based society will have to cope not only with accidental and non-malicious human-made faults, but also with malicious faults of different severity up to the possibility of terrorist attacks against infrastructures or through infrastructures. In addition to these faults, taking into account the commercial strategies of large industries - Microsoft has just announced and will soon commercialize very powerful communication gadgets, which will make ubiquitous computing a reality [Microsoft Research, 2003]-, a new type of subtle fault will become evident in the near future. They will be mainly generated by the combination of: 1) a very large number of computer-controlled systems of common usage, and 2) a large number of non-trained users operating such systems. Envisioning a type of "common mode operational fault" with unpredictable social consequences is no more futuristic.

This landscape needs a "global" view of the concept of "dependability", which has to start from the basic intrinsic characteristics of the components (of a computer, of a network, of an infrastructure, of a set of services, of the interested managing and user bodies, of the society) to grow up and reach reliance in "ambient dependability", which encompasses not only the technological aspects but includes inter and multi disciplinary fields, which span over ergonomics, usability, education, sociology, law and government.

A first step towards ambient dependability is achieving a dependable Information Society that requires a harmonized effort from a large set of actors, and need to consider many challenging points:

- New threats have to be analyzed, studied and modeled.
- New fault types have to be analyzed, studied and modeled.
- Design methodologies have to be studied for designing under uncertainty.
- A user-centered design approach, like design for usability, has to be applied.
- The concepts of "architecture" and "system" have to be rethought and redefined.
- Architectural frameworks are needed for adapting functional and non-functional properties while at least providing guarantees on how dependably they are adapting.
- A move is needed towards the definition of extended dependability attributes, like "acceptable availability under attack".

- New modeling and simulation means and tools are needed for complex interdependencies, for system evolution, evaluation of combined measures, evaluation of vulnerabilities related to security to mention some of challenges.

1.1 Where are we today?

We are facing such a huge problem that it is important to understand what level of trustworthiness we may pose on our present (theoretic) knowledge and industrial practice. Some data:

- In 1995, The Standish Group [The Standish Group] reported that the average US software project overran its budgeted time by 190%, its budgeted costs by 222%, and delivered only 60% of the planned functionality. Only 16% of projects were delivered at the estimated time and cost, and 31% of projects were cancelled before delivery, with larger companies performing much worse than smaller ones. Later Standish Group surveys show an improving trend, but success rates are still low.

- A UK survey, published in the 2001 Annual Review of the British Computer Society [British Computer Society 2001] showed a similar picture. Of more than 500 development projects, only three met the survey's criteria for success. In 2002, the annual cost of poor quality software to the US economy was estimated at $60B [NIST, 2002].

While recognizing that much advancement has been made in dependability methods, tools and processes, still a great gulf remains between what is known and what is done. It appears evident that many industrial engineering designs are still based on best effort processes with limited, if any, application of the theories developed so far. This implies a relevant educational issue, addressed later.

Current research in dependability covers a wide spectrum of critical systems, going from embedded real-time systems, to large open networked architectures. The vision of research and technology development in Dependable Computing has been summarized by the dependability community involved in the IST-2000-25088 CaberNet Network of Excellence (`http://www.newcastle.research.ec.org/cabernet/research/projects`), where a considerable number of pointers to relevant research activities is also provided. A brief recall of this vision document is reported in the following.

Fault Prevention

Fault prevention aims to prevent the occurrence or the introduction of faults. It consists in developing systems in such a way as to prevent the introduction of

design and implementation faults, and to prevent faults from occurring during operation. In this context, any general engineering technique aimed at introducing rigor into the design process can be considered as constituting fault prevention. At level of *specifications,* the assumption that a complex software-based system may have a fixed specification is wrong. Today systems have in their requirements, and therefore in the specifications, assumptions related to human processes, from their interactions to the way they report their results. Such processes change as normal evolutionary processes. Thus software engineering methods must support these changes, but this is not the actual situation. The relevant issues to be considered are related to the *form* of the specifications (read it as the need to use proper abstractions) and to the *notation* of the specification (read it as the need to use formal mathematical notations, able to manage the changes). If in the specification process requirements are dealt with approximate or ambiguous notations, not able to provide manageability of abstractions, it is difficult, if not impossible, to know what it is tried to be achieved. Particularly challenging appear: i) the formal definition of security policies in order to prevent the introduction of vulnerabilities, and ii) the human factors issues in critical "socio-technical" systems.

Fault Tolerance

Fault-tolerance techniques aim to ensure that a system fulfils its function despite faults. Current research is centred on *distributed* fault-tolerance techniques, *wrapping* and *reflection* technologies for facilitating the implementation of fault-tolerance, and the generalization of the tolerance paradigm to include deliberately malicious faults, i.e., *intrusion-tolerance.*

Distributed fault-tolerance techniques aim to implement redundancy techniques using software, usually through a message-passing paradigm. Much of the research in the area is concerned with: i) the definition of distributed algorithms for fault-tolerance; ii) facilities for group communications and consensus; iii) automatic recovering and re-integration of failed replicas; iv) fault-tolerance techniques for embedded systems (tailored to both the synchronous and asynchronous models); v) tolerance to intrusion and intruders.

Other areas of active research concern fault-tolerance in large, complex distributed applications. Of special note in this area are techniques aimed at the coordinated handling of multiple exceptions in environments where multiple concurrent threads of execution act on persistent data; fault-tolerance in peer-to-peer systems; recursive structuring of complex cooperative applications to provide for systematic error confinement and recovery and mechanisms for dealing with errors that arise from architectural mismatches.

The implementation of distributed fault-tolerance techniques is notoriously difficult and error-prone, especially when using COTS (off-the-shelf compo-

nents) that typically (a) have ill-defined failure modes and (b) offer opaque interfaces that do not allow access to internal data without which fault-tolerance cannot be implemented. There is thus considerable interest in addressing these difficulties using wrapping technologies to improve robustness and reflective technologies to allow introspection and intercession.

Fault Removal

Fault removal, through verification and validation techniques such as inspection, model-checking, theorem proving, simulation and testing, aims to reduce the number or the severity of faults. Among the most recent research activities in this field, there are:

i) *probabilistic verification,* an approach with strong links to fault forecasting that aims to provide stochastic guarantees of correctness by neglecting systems states whose probability of occupation is considered negligible;

ii) *statistical testing,* an approach to software testing which is based on the notion of a test quality measured in terms of the coverage of structural or functional criteria;

iii) assessment of the correlation between software complexity, as measured by object-oriented metrics, and fault proneness;

iv) *robustness testing,* aiming to assess how well a (software) component protects itself against erroneous inputs. Fault tolerant mechanisms can be tested for their robustness through classical *fault injection;*

v) testing the use of the reflective technologies considered earlier as a means for simplifying the implementation of fault-tolerance;

vi) support to verification and validation, for analyzing the impact of changes and for ensuring that the design and all its documentation remain consistent.

Fault Forecasting

Fault forecasting is concerned with the estimation of the presence, the creation and the consequences of faults. This is a very active and prolific field of research within the dependability community. Analytical and experimental evaluation techniques are considered, as well as simulation.

Analytical evaluation of system dependability is based on a stochastic model of the system's behaviour in the presence of fault and (possibly) repair events. For realistic systems, two major issues are that of: (a) establishing a faithful and tractable model of the system's behaviour, and (b) analysis procedures that allow the (possibly very large) model to be processed.

Ideally, the analytical evaluation process should start as early as possible during development in order to make motivated design decisions between alternative approaches. Specific areas of research in analytical evaluation include: systems with multiple phases of operation, and large Internet-based applications requiring a hierarchical modelling approach. In the area of software-fault tolerance, specific attention must be paid to modelling dependencies when assessing the dependability achieved by diversification techniques for tolerating design faults.

Experimental evaluation of system dependability relies on the collection of dependability data on real systems. The data of relevance concerns the times of or between dependability-relevant events such as failures and repairs. Data may be collected either during the test phase or during normal operation. The observation of a system in the presence of faults can be accelerated by means of fault-injection techniques, which constitute a very popular subject for recent and ongoing research. Currently, there has been research into using fault injection techniques to build dependability *benchmarks* for comparing competing systems/solutions on an equitable basis.

1.2 System Dependability and Education

System dependability is "per se" very challenging, but there is the feeling that a methodological approach and a methodic attitude are lacking. No system developer yet knows what is the appropriate blend of methods for fault prevention, fault removal, fault tolerance and fault forecasting. Also, the choices of dependability cases appear to be rather difficult: a meaningful case should contain all the evidence that is needed to argue that the system is adequately dependable. This evidence is made of the dependability targets and related failure rates, hazard analysis, arguments based on the architecture and design just to mention few of them. The lack of such methodic approach is the hardest point to take into account. The situation is even worse considering that a gap exists between theoretical knowledge and what is done in practice.

This open a very relevant educational issue: there is a gap between what is *known* and what is *done*. A point that should be raised, when shaping any system design course is the ratio of fundamentals and principles, which should be taught, and the part of teaching related to the status of the art in techniques and technologies. A student, who in the future will become part of a team of system developers should be taught all fundamentals and principles which will remain invariant in the years to come. This is the only way to reduce the gap between what is *known* and what is *done*.

There is the need of a careful re-visitation of several methodologies and design methods that have been largely studied when computer science had to be credited as a science, and then quickly abandoned when under the pressure

of the market a trend has started towards more and more functionally rich and sophisticated artifacts.

Relevant keywords, which may drive towards new architectural frameworks for a dependable Information Society, are:

- Abstraction
- Composition
- Recursion
- Integration
- Usability

They can be used to get the required level of genericity, openness, adaptability and re-use for different architectural/infrastructural layers:

- For designing dependable architectures/infrastructures at component level.
- For designing architectures/infrastructures for dependability.
- For obtaining dependable architectures/infrastructures from user perspective.

Design of dependable components for architectures/infrastructures

Table 1 identifies requirements, enabling technologies and instruments that are well suited for dependable architectures made of components.

Modelling, designing and using **generic, composable, open source, and reusable components** appear from this table very helpful towards the goal of building systems of systems that can be easily validated and assessed.

Designing architectures/infrastructures for dependability

Another perspective is related to coping with "how to?" (Table 2). Here multiple facets of dependability raise many issues.

We can deduce that **abstraction, recursion, and incremental verification** will definitely help in designing and structuring multi-layers architectures up to the level of complex infrastructures.

Dependable architectures/infrastructures from user perspective

A final perspective is the architectural level that includes the user. Actually the user is the one who has the final word on system dependability. The types of user requests, which can be used for driving this level, are in Table 3.

Table 1.

	Rigorous design (i.e. fault prevention)	*Verification and validation (i.e. fault removal)*	*Fault Tolerance (accidental and malicious faults)*	*System evaluation (i.e. fault forecasting)*
Requirements	- Composable components - Secure components - Separation of concern - Invariance	- Early prototyping - Test cases generation	- Adaptable components	- Testable components - Coverage evaluation - Early prototyping
Enabling technologies	- Formal methods - Design for V&V	- State observability - Testing - Supports to validation and verification - Formal methods	- Redundancy - Functional diversity - Middleware	- Analytical modeling - Fault injection
Instruments	- Specs languages - Modeling	- Tools	- Function placement	- Tools

Table 2.

Rigorous design (i.e. fault prevention)	*Verification and validation (i.e. fault removal)*	*Fault Tolerance (accidental and malicious faults)*	*System evaluation (i.e. fault forecasting)*

- How to compose:
  - Interfaces
  - Legacy systems
- How to guarantee integrity
- How to guarantee security
- How to guarantee survivability
- How to guarantee predictable timing
	- How to assess risks
	- How to trust the tools
	- How to test
		- How to cope with new fault types
		- How to reach survivability
		- How to coordinate adaptability
		- How to get good usability
			- How to deal with uncertainty
			- How to build meaningful models and simulations
			- How to evaluate coverage
			- How to perform experimental verification and testing

Table 3.

Rigorous design (i.e. fault prevention)	*Verification and validation (i.e. fault removal)*	*Fault Tolerance (accidental and malicious faults)*	*System evaluation (i.e. fault forecasting)*
- Is the system compliant with specifications? - Is the system able to adapt to changes	- Do I have the knowledge of possible residual faults?	- Is the system able to provide meaningful service in presence of accidental and malicious faults?	- Has the system sufficient performance to satisfy my needs? - Is system usability sufficiently good to reduce the probability of human errors? - Does the system protect my privacy, integrity of my data and security? - Is the cost/dependability ratio optimal for my needs?

The final rating for a system or infrastructure or service being perceived as dependable comes from a statement like **"I think the system/infrastructure/service has an adequate dependability, obtained in an efficient way!".** And this is the real judgment that counts.

Reaching this level of trust and confidence is a very challenging goal. It needs to consider any system/infrastructure/service from different perspectives and distilling from them the very special aspects that contribute to overall dependability.

1.3 Future Directions

The discussion carried on in the previous sections has analysed and pointed out many requirements for dependability in the future AmI landscape, which encompass different key aspects of the future generation computing systems. Here, some challenging research directions, as envisioned by the authors, are discussed.

Generic, COTS-based architectures for dependable systems

The natural evolution towards more complex services and infrastructures impose an enhancement on how an "architecture" is designed, as the supporting element for the system and the services, and what it is supposed to offer in terms of different x-abilities.

Most of the large-scale infrastructures have been developed connecting previously stand-alone systems usually developed from proprietary architectures, where ad hoc solutions were chosen and several electronic components were developed independently.

A basic property in such a context was that the components were designed having in mind the entire structure of the system, and this type of approach had pros and cons.

Positive aspects were:

- The design and implementation of ad hoc components makes easier the validation of the system.
- The knowledge of the system is completely under control of the designer and parts do not exist which are protected by third party intellectual property rights, again making easier validation and procuring, which are mandatory for safety critical systems.
- Re-design and updating the system does not depend on third parties.

Negative aspects were:

- Components and implementation technologies changes and evolutes very quickly, so that several of them may be obsolete when a design is completed and the system can be put in operation.
- Upgrading of components may be required if the operational life of the system is rather extended.
- The strict dependence between components and the system (through the design) makes the system rather inflexible and not adaptable to different contexts or to be interfaced to other systems, that is configurability may be very hard if not impossibe.
- Any new system or major revision needs to be revalidated ex-novo.

Moreover, with respect to interactions, for the integration of large-scale infrastructures, or simply aiming at interoperability between different systems negative aspects of such architectural approaches are:

- Systems with even slightly different requirements and specifications cannot reuse components used in previous designs, so that new systems require, in general, a complete redesign, and experience gained from the operation of older systems cannot be used.
- Interoperability is hard to achieve because of different projects specifications (different dependability properties offered, different communication protocols or media etc.), and the integration of two or more different systems must consider this.

For being able to follow the trend which demands the usage of COTS (components off-the-shelf) and to avoid the negative points previously listed, thus reducing development and operational costs, a strategic R&D activity towards the definition, prototyping and partial verification and validation of a generic, dependable, and real-time architecture is required. Such an effort would aim at the definition and construction of an architectural framework such:

- To reduce the design and development costs.
- To reduce the number of components used in the several subsystems.
- To simplify the evolution process of the products and reduce the associated costs.
- To simplify the validation (and certification) of the products through an incremental approach based on reuse.

The proposed infrastructure should have the following characteristics:

- Use of generic components (possibly COTS) which can be substituted, following technological evolution, without redesign or revalidation of the system.
- Reliability/availability and safety properties should be associated to the architectural design and not only to intrinsic properties of its components, so that techniques for error detection, diagnosis and error recovery be as most as possible independent from the specific components, be they hardware or software.
- Use of a hierarchical approach for functional and non functional properties, so to ease validation.
- Use of early evaluation methods to support design refinements. An early validation of the concepts and architectural choices is essential to save on money and on the time to market for a final product. The feedback of such evaluation is highly beneficial to highlight problems within the design, to identify bottlenecks and to allow comparing different solutions so as to select the most suitable one.
- Openness of the system, in the sense that it should be able to interface and communicate with other systems through different communication systems and to adapt itself to the different kind of architectures it has to interact with.

Developing dependable systems requires also an open utility program framework which may be easily and dynamically enriched to cope with new needs

which may arise during system lifetime. On-line support for administration, operation, maintenance and provisioning is needed. Examples of key areas are: error/fault diagnosis, QoS analysis to trigger appropriate reconfiguration actions in faulty situations, on-line software upgrades without loss of service.

A large body of activities has been performed in this direction by the international dependability community. The main contributions of our group are [Bondavalli et al., 2000; Bondavalli et al., 2001a; Porcarelli et al., 2004].

Model-based dynamic reconfiguration in complex critical systems

Information infrastructures, especially as foreseen by the AmI vision, are very complex networked systems where interdependencies among the several components play a relevant role in the quality of services they provide. In such system organization, the failure of a core node may induce either saturation on other parts of the infrastructure or a cascading effect which puts out of work large part of the infrastructure, with consequent loss of service and connectivity with lengthy recovery times. Therefore, adaptivity of the system architecture with respect to unforeseen changes that can occur at run-time becomes one of the most challenging aspects. Apart from natural system's evolution, many other sources of variability are possible, such as the occurrence of fault patterns different from those foreseen at design time, or the change of application's dependability requirements during the operational lifetime. To cope with unpredictability of events, approaches based on on-line system reconfiguration are necessary/desirable.

A simplified solution to cope with such a dynamic framework would be to pre-plan (through off-line activities) the "best" reaction to system and/or environment conditions, and to utilize the appropriate pre-planned answer when a specific event occurs at run-time. However, such a solution would be practically feasible only in presence of a limited and well defined in advance number of situations requiring the application of a new reconfiguration policy in the system. Unfortunately, especially in complex systems, such a complete knowledge is not available, and situations may occur for which a satisfactory reaction has not been foreseen in advance.

Therefore, it raises up the need of dynamically devising an appropriate answer to variations of the system and/or environment characteristics in order to achieve the desired dependability level. To this purpose, a general dependability manager would be very useful, which continuously supervises the controlled system and environment, ready to identify and apply reconfiguration policies at run-time.

The architectural definition of such a general dependability manager includes an evaluation subsystem to provide quantitative comparison among sev-

eral possible reconfiguration strategies. Model-based evaluation approaches are very suited to this purpose. The idea is to build simplified (but still meaningful) models of the system to be controlled. The model simplicity in such a context is dictated by the need to solve the model dynamically as quickly as possible, in order to take appropriate decisions online. Too complex systems, in fact, would require too high computation time, thus defeating the effectiveness of the solution itself. In a logical view, monitoring entities have to be inserted in the framework, able to catch exceptional system/environment conditions and to report appropriate signals to the dependability manager. Issues of distributed observation and control are involved in this process. Simple but yet effective indicators have to be defined, as a synthesis of a set of "alarming symptoms" (such as fault occurrence, different applications' request, traffic conditions, detection of attacks, ...). Based on critical values assumed by such indicators, a reconfiguration action is triggered. Of course, because resources are precious for assuring satisfactory levels of service accomplishments, the triggering of reconfiguration/isolation procedures have to be carefully handled.

As soon as a system reconfiguration is required, the model solution helps to devise the most appropriate configuration and behavior to face the actual situation. For example, through the model solution it can be evaluated the dependability of a new architecture of the system obtained by rearranging the remaining resources after a fault, or some performability indicator to carry out cost-benefit tradeoff choices. Therefore, the output provided by the dependability manager is a new system configuration; of course, it is expected to be the best reconfiguration in order to satisfy the dependability requirements. Different modeling techniques and models solution can be considered and integrated to reach the goal. Already evaluated reconfiguration actions can be maintained in a database accessible by the dependability manager, to be easily retrieved when the same "alarming pattern" will be subsequently raised in the system.

A general framework should be pursued, not tied to a specific application but flexible enough to be easily adapted to different problems. In particular, a methodology has to be defined, allowing to identify systematically the input parameters of the manager, the metrics of interest and the criteria to base the decision on. These are very challenging issues, especially the last one, which is an instance of the well known and long studied problem of multiple-criteria decision making.

Also in this area there is a noteworthy body of research. The main contributions of our group are [Porcarelli et al., 2004; Bondavalli et al., 1999].

Enhancing methods for dependability evaluation

System evaluation is a key activity of fault forecasting, aimed at providing statistically well-founded quantitative measures of how much we can rely on a system. In particular, system evaluation achieved through modelling supports the prediction of how much we will be able to rely on a system before incurring the costs of building it. It is therefore a very profitable evaluation approach to be employed since the very beginning of a system development activity.

However, a number of new issues are raised by the relevant characteristics of the future systems, that are not satisfactorily dealt with by current modelling methodologies. Most of the new challenges in dependability modelling are connected with the increasing complexity and dynamicity of the systems under analysis. Such complexity need to be attacked both from the point of view of system representation and of the underlying model solution. A few issues and directions to go are discussed in the following.

State-space explosion and ways to cope with it. The state space explosion is a well known problem in model-based dependability analysis, which strongly limits the applicability of this method to large complex systems, or heavily impacts on the accuracy of the evaluation results when simplifying assumptions are made as a remedy to this problem. Modular and hierarchical approaches have been identified as effective directions; however, modularity of the modelling approach alone cannot be truly effective without a *modular solution* of the defined models.

Hierarchical approaches. Resorting to a hierarchical approach brings benefits under several aspects, among which: i) facilitating the construction of models; ii) speeding up their solution; iii) favoring scalability; iv) mastering complexity (by handling smaller models through hiding, at one hierarchical level, some modeling details of the lower one).

At each level, details of the architecture and of the status of lower level components are not meaningfull, and only aggregated information should be used. Therefore, information of the detailed models at one level should be aggregated in an abstract model at a higher level. Important issues are how to abstract all the relevant information of one level to the upper one and how to compose the derived abstract models.

Composability. To be as general as possible, the overall model (at each level of the hierarchy) is achieved as the integration of small pieces of models (building blocks) to favour their composability. We define composability as the capability to select and assemble models of components in various combinations into a model of the whole system to satisfy specific application requirements.

For the sake of model composability, we are pursuing the following goals:

- to have different building block models for the different types of components in the system. All these building blocks can be used as a pool of templates,
- to automatically instantiate an appropriate model, one for each component, from these templates, and
- at a given hierarchical level, to automatically link them together (by means of a set of rules which are application dependent), thus defining the overall model.

The international dependability community is very active on topics related with methods and tools for dependability evaluation and a massive production exists on the several involved aspects. The main contributions of our group on the above discussed issues are [Mura and Bondavalli, 1999; Bondavalli et al., 2001b; Mura and Bondavalli, 2001].

On-line evaluation as a component/mechanism for dynamic architectures. Dependability evaluation is typically an off-line activity. However, because of the unpredictability of events, both external and internal to the system, on-line evaluation would be desirable in a number of circumstances. In fact, when the topology of the architecture changes significantly along time, accounting for too many configurations may become a prohibitive activity when done off-line. Of course, although appealing, the online solution shows a number of challenging problems requiring substantial investigations. Models of components have to be derived online and combined to get the model of the whole system. Thus, compositional rules and the resulting complexity of the combined model solution appear to be the most critical problems to be properly tackled to promote the applicability of this dynamic approach to reconfiguration.

The main contributions of our group on this research direction are [Porcarelli et al., 2004; Bondavalli et al., 1999; Chohra et al., 2001].

Integration of experimental and model-based evaluation. Moreover, synergistic collaboration between model-based and measurement approaches throughout the system life cycle is more and more pressed by the future landscape. This calls for *benchmarking for dependability,* to provide a uniform, repeatable, and cost-effective way of performing the evaluation of dependability and security attributes, either as stand-alone assessment or, more often, for comparative evaluation across systems and components. The shift from system evaluation techniques based on measurements to the standardized approaches required by benchmarking touches all the fundamental problems of current

measurement approaches (representativeness, portability, intrusion, scalability, cost, etc.) and needs a comprehensive research approach, with a special attention to COTS and middleware aspects.

The main contributions of our group in this area are [Bondavalli et al., 2002; Bondavalli et al., 2003].

Appendix: ISTAG scenario: Maria - The Road Warrior

After a tiring long haul flight Maria passes through the arrivals hall of an airport in a Far Eastern country. She is traveling light, hand baggage only. When she comes to this particular country she knows that she can travel much lighter than less than a decade ago, when she had to carry a collection of different so-called personal computing devices (laptop PC, mobile phone, electronic organizers and sometimes beamers and printers). Her computing system for this trip is reduced to one highly personalized communications device, her 'P-Com' that she wears on her wrist. A particular feature of this trip is that the country that Maria is visiting has since the previous year embarked on an ambitious ambient intelligence infrastructure program. Thus her visa for the trip was self-arranged and she is able to stroll through immigration without stopping because her P-Com is dealing with the ID checks as she walks. A rented car has been reserved for her and is waiting in an earmarked bay. The car opens as she approaches. It starts at the press of a button: she doesn't need a key. She still has to drive the car but she is supported in her journey downtown to the conference center-hotel by the traffic guidance system that had been launched by the city government as part of the 'AmI-Nation' initiative two years earlier. Downtown traffic has been a legendary nightmare in this city for many years, and draconian steps were taken to limit access to the city center. But Maria has priority access rights into the central cordon because she has a reservation in the car park of the hotel. Central access however comes at a premium price; in Maria's case it is embedded in a deal negotiated between her personal agent and the transaction agents of the car-rental and hotel chains. Her firm operates centralized billing for these expenses and uses its purchasing power to gain access at attractive rates. Such preferential treatment for affluent foreigners was highly contentious at the time of the introduction of the route pricing system and the government was forced to hypothecate funds from the tolling system to the public transport infrastructure in return. In the car Maria's teenage daughter comes through on the audio system. Amanda has detected from 'En Casa' system at home that her mother is in a place that supports direct voice contact. However, even with all the route guidance support Maria wants to concentrate on her driving and says that she will call back from the hotel. Maria is directed to a parking slot in the underground garage of the newly constructed building of the Smar-tel Chain. The porter - the first contact with a real human so far! - meets her in the garage. He helps her with her luggage to her room. Her room adopts her 'personality' as she enters. The room temperature, default lighting and a range of video and music choices are displayed on the video wall. She needs to make some changes to her presentation - a sales pitch that will be used as the basis for a negotiation later in the day. Using voice commands she adjusts the light levels and commands a bath. Then she calls up her daughter on the video wall, while talking she uses a traditional remote control system to browse through a set of webcast local news bulletins from back home that her daughter tells her about. They watch them together. Later on she 'localizes' her presentation with the help of an agent that is specialized in advising on local preferences (color schemes, the use of language). She stores the presentation on the secure server at headquarters back in Europe. In the hotel's seminar room where the sales pitch is take place, she will be able to call down an encrypted version of the presentation and give it a post presentation decrypt life of 1.5 minutes. She goes downstairs to make her presentation... this for her is a high stress event. Not only is she performing alone

for the first time, the clients concerned are well known to be tough players. Still, she doesn't actually have to close the deal this time. As she enters the meeting she raises communications access thresholds to block out anything but red-level 'emergency' messages. The meeting is rough, but she feels it was a success. Coming out of the meeting she lowers the communication barriers again and picks up a number of amber level communications including one from her cardio-monitor warning her to take some rest now. The day has been long and stressing. She needs to chill out with a little meditation and medication. For Maria the meditation is a concert on the video wall and the medication... a large gin and tonic from her room's minibar.

Plots elements

WHO: Maria, Devices: P-Com, Cardio monitor; AmI service providers: Immigration control system, Rent-a-car, Car, Traffic management, Hotel Smart-tel, Hotel room, Seminar room, Maria's Company remote access, "En casa" system.

WHERE: Airport, City roads, Hotel, Home.

WHEN: Arrival to airport and immigration control; Car rental and trip through city; Arrival to hotel; Communication to/from "En casa" system; Access to room and adaptation; Communication with company; Presentation delivery; Relaxing.

WHAT:

- Maria: Interacts with P-Com through screen/voice interfaces. P-Com works as Personal data holder, Communications device, Negotiation tool; Holds a Cardio monitor that monitors and transmits signals
- Immigration control system: Carries out an ID check.
- Rent-a-car system: Automates the renting process by means of transaction agents.
- Car: Manages the access to the car, supports driving in the city traffic, offers telecommunications.
- Traffic management system: Manages access rights to the city streets.
- Hotel (Smart-tel): Manages parking reservation, provides local information, activates transaction agents and local preferences agents.
- Hotel Room: Manages temperature/light controls, Video/music adaptive systems, commanded bath, communications systems, provides facilitates for editing documents.
- Seminar room: Provides communications, and document life manager.
- Company remote access: Provides remote billing, negotiation agents, Remote access to servers.
- "En casa" communications system: Provides Multimedia communication capabilities.

ASSETS: Personal data, business data, infrastructure.

Scenario characteristic: Degrees of Comfort

Maria in this scenario moves through different public and private spaces, which she characterizes according to specific values that determine the information she will exchange and the type of interactions she will engage in. She starts her journey at the airport, she moves in a public space and a public invidualized profile seems convenient and comfortable at this stage as her P-Com helps her clear passport controls quickly and to be instantly contactable and recognizable. In the car from the airport to the hotel, she is in a semi-public space and maybe she desires to concentrate on her thoughts and be disturbed only by a select group of people, maybe

her family and work associates - En Casa is recognized as a cyberspace object in her select group. In the Smar-tel hotel she is in semi-private space, as it is offered in temporary privacy to successive individuals. She may want the room therefore to behave like a trusted private space or she may feel more comfortable switching her AmI space off.

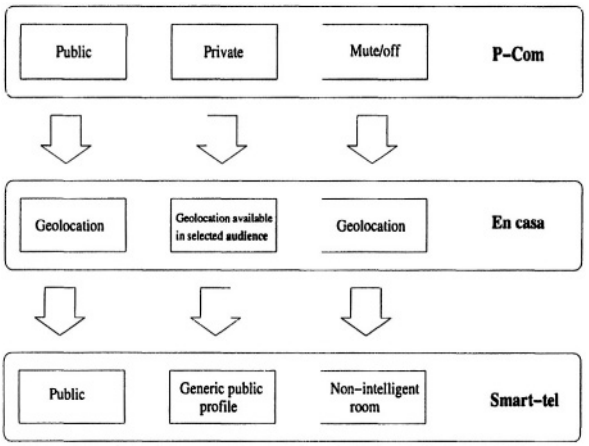

Figure A.1. Maria - Degrees of comfort.

The scenario continues: what goes wrong and how it works out...

The P-com (support for critical personal information) that Maria wears around her wrist sometimes fails causing her to get stranded at airports and wherever she needs to identify herself. A broken P-com is reason enough for the security system at her office to alert the authorities and refuse access to the office building that she wants to enter. To minimize delays in case of trouble with her P-com, Maria always carries her passport and some other documents describing her identity and containing passwords, user ID's and other kinds of codes. Malfunctioning of her P-com not only makes Maria lose her digital identity (single point of failure), it can also make her invisible to the tracking systems that keep an eye on her whereabouts for her daughter, the company she works for and her friends. For this reason Maria still carries a mobile phone. P-Com's rarely get stolen. However, high tech devices are frequently the target of "crackers" that try to gain access to the digital identity of the owner and their assets. For this reason Maria demonstrates some lack of trust in the hotel system although it claims to deploy strong encryption. This sometimes motivates a constant level of alert, as prevention of nuisances and inconvenience: some while ago Maria was stopped while trying to pass through an airport as suspected owner of a stolen identity.

The role of AmI and potential threats

Privacy. AmI understands context, and its spatial awareness adapts to the socially accepted natural modes of interaction between people. In Maria's case, her P-Com enables her to be

public to the authorities but not to the crowd in general, although some degree of surprise and unexpected meeting might be welcome by her. In the taxi her P-Com enables her to retain a moment of reflection and contemplation by respecting some privacy, which fits well to the occasion. At the hotel, she would feel better to switch it off altogether, or to make it selectively public depending on her mood. Digital ID theft is a major concern, and the main menace to trust in AmI. As Maria comes to rely more on the P-Com and its support to her digital identity, its loss is likely to be increasingly traumatic. Tolerability to this risk will be similar to the fervent social debates society has conducted concerning risks in the area of personal safety. Countering ID theft requires a combination of different social engineering and technical counter-mechanisms. Prevention is likely to be a continuous competition between defense and attack techniques in a form of "arms race". The attitude to risk is likely to vary widely between social and national groups. There may be P-Com product families that allow the user to choose on the basis of (informal) risk assessments, in a trade off among style, functionality and robustness. To this end there would need to be clear methods for assessing the probability of digital ID loss, its duration, and the potential consequences and liabilities. In addition to the violation of the confidentiality of the digital ID, there is also the problem of partial loss of personal data (which could have effects such as erosion of privileges, counterfeiting of personal security policies, ...). As users will manage multiple identifies for their different social roles (citizen, employee, mother, ...), this could lead to inappropriate behavior of P-Com in certain contexts - a failure that could be much more difficult to detect.

Trust and confidence. AmI acts as Maria's invisible assistant, It efficiently provides Maria the necessary resources to enable her to successfully complete her mission. AmI is unobtrusive; it silently and discreetly acts for her in the background. It has to be robust, pro-actively correcting any failures that may occur. AmI increases Maria's capabilities: she can act, while her environment takes care of her needs. The other side of Maria's trust and confidence on the P-Com is the need to make provision for when it fails. The severity of the failure and its duration depend on the extent to which there are methods for rapid recovery and what alternative methods are in place e.g. how practicable is it to revert to a passport and a mobile phone to replace P-Com. The medical advice given by the cardio-monitor has serious safety implications. If it really behaves as an active e-health device integrated into AmI, there should be serious evidence of its dependability. Most users would feel comfortable with isolated devices that provide some assistance, without interacting with public open systems. Trust and confidence have to be also developed on AmI infrastructure support systems. Maria has also done some type of informal risk assessment to judge that the hotel digital security may not be adequate for the sensitive content of her presentation. In a future she might ask for some security certification and third-party evaluation. Maria seems to have placed complete trust in the navigation and travel support that she accesses through the P-com. The dependability requirements of AmI are very influenced by the performance, capability and dependability offered by alternative applications. Is the ordinary taxi system still functioning so that it will be only a mild inconvenience if the automatic traffic control system is not available? What would happen should she get lost? There may be longer-term negative consequences of AmI related to the development of strong dependence on AmI. Any unavailability or incorrect response by AmI might provoke the blockage of Maria's personal, social and professional lives. She has incorporated AmI into her existence, and now she cannot survive without it.

Interdependencies. This AmI scenario makes very apparent the interdependence of systems deriving from the pervasive deployment of ICT-based systems. Studying each dependability attribute in isolation might be misleading. The propagation and cascading effects that

can derive from the failure of single elements might cause effects that are unforeseeable at first sight. The interconnectivity among systems requires different levels for the analysis of the potential threats to dependability. Faults that might not affect a local system might distress the more general application, and trigger a distant failure. For instance, a rather small diminution in the bandwidth of the network supporting the traffic system might in the long run cause an accumulation of delays and problems in the management of parking places. A significant problem is the widespread presence of single failure points. Any disturbance of the immigration control could affect a great quantity of people and of other services. It is also important to emphasize that most of the services depend on mobile code and agents, who are in charge of representing the different actors, perform negotiations and transactions, and to take crucial decisions for their owners. The dependability of these agents appears as one of the weakest points of AmI. In the Road Warrior scenario privacy can be managed in a linear way, by establishing profiles that are pre-configured by the user, by context, audience/communication group, time or day, location, etc. Private, semi-private, public, semi-public and mute profiles with all the possible nuances in-between may be identified or 'learnt' by use and pattern. Maria would feel more comfortable setting her P-com intentionally in the degree of comfort that she would require in each situation or teaching it to understand her personal social protocol of what is comfortably public and what is safely private.

Additional dependability issues. The assets at risk are of very different nature. They range from Maria's personal data to business data of Maria's company and of the hotel, and even other infrastructures such as the city traffic management system. Personal data can be the object of different critical faults:

- If P-Com has an internal accidental failure in the airport or when accessing the car or when entering into the hotel (for instance, availability of data, or integrity of data) Maria will be unable to benefit from these AmI services.
- More importantly, P-Com can be subject to malicious attack when acting in open environments (e.g. in the airport, in the car, in the hotel), putting the confidentiality of her personal data at risk. Business data is exposed at each dialogue between AmI business systems and personal and social-wide systems. When dealing with potential customers, any AmI has to send and accept information that can be the source of faults.
- Accidental faults can affect mainly the availability of the service and the integrity of the data managed.
- Malicious failures can provoke an attack to the confidentiality of data, and to the availability of the service. A third type of risks is related to AmI services that are deployed as societal infrastructures, for instance the traffic management system. Here, any fault might cause not just nuisances but potentially critical accidents. The dependability attributes of all hardware and software components, and of the data processed and stored by the system are relevant for the whole of society, giving rise to a new type of Critical Infrastructure.

References

AMSD (2003). A dependability roadmap for the information society in Europe. Deliverable D1.l, Accompanying Measure on System Dependability (AMSD). IST-2001-37553 - Workpackage 1: Overall Dependability Road-mapping, `http://www.am-sd.org`.

Bondavalli, A., Chiaradonna, S., Cotroneo, D., and Romano, L. (2003). A fault-tolerant distributed legacy-based system and its evaluation. In *LADC2003 - First Latin-American Symposium on Dependable Computing,* São Paulo, Brazil. to be published.

Bondavalli, A., Chiaradonna, S., Di Giandomenico, F., and Grandoni, F. (2000). Threshold-based mechanisms to discriminate transient from intermittent faults. *IEEE Transactions on Computers,* 49(3):230–245.

Bondavalli, A., Chiaradonna, S., Di Giandomenico, F., Grandoni, F., Powell, D., and Rabejac, C. (2001a). Error processing and fault treatment. In Powell, D., editor, *A Generic Fault-Tolerant Architecture for Real-Time Dependable Systems,* pages 71–86. ISBN 0-7923-7295-6, Kluwer Academic Publishers, Boston.

Bondavalli, A., Coccoli, A., and Di Giandomenico, F. (2002). Qos analysis of group communication protocols in wireless environment. In Ezhilchelvan, P. and Romanovsky, A., editors, *Concurrency in Dependable Computing,* pages 169–188. Kluwer Academic Publishers.

Bondavalli, A., Di Giandomenico, F., and Mura, I. (1999). An optimal value-based admission policy and its reflective use in real-time dependable systems. *Real-Time Systems Journal, Kluwer Academic Publishers,* 16(1):5–30.

Bondavalli, A., Nelli, M., Simoncini, L., and Mongardi, G. (2001b). Hierarchical modelling of complex control systems: Dependability analysis of a railway interlocking. *Journal of Computer Systems Science and Engineering, CRL Publishing,* 16(4):249–261.

British Computer Society 2001. `http://www.bcs.org.uk`.

Chohra, A., Di Giandomenico, F., Porcarelli, S., and Bondavalli, A. (2001). Towards optimal database maintenance in wireless communication systems. In Callaos, N., Da Silva, N. I., and Molero, J., editors, *The 5th World Multi-Conference on Systemics, Cybernetics and Informatics, ISAS-SCI 2001, Volume I: Information Systems Development,* pages 571–576, Orlando, Florida, USA. IIIS.

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., and Burgelman, J-C. (2001). Scenarios for ambient intelligence in 2010. Final report, IST Advisory Group (ISTAG), European Commission, IST. European Communities. `http://www.cordis.lu/ist/istag-reports.htm`.

European Presidency (2000). Presidency conclusions. Lisbon European Council. 23 and 24 March 2000. `http://ue.eu.int/en/Info/eurocouncil/index.htm`, Council of the European Union.

ISTAG (2002). Trust, dependability, security and privacy for IST in FP6. Ist advisory group report, IST Programme, EC. `http://www.cordis.lu/ist/istag-reports.htm`.

Laprie, J. C., editor (1992). *Dependability: Basic Concepts and Associated Terminology,* volume 5 of *Dependable Computing and Fault-Tolerant Systems.* Springer-Verlag.

Microsoft Research (2003). Microsoft research faculty summit 2003, Cambridge, UK. `http://research.microsoft.com/collaboration/university/europe/Events/FacultySummit/2003`.

Mura, I. and Bondavalli, A. (1999). Hierarchical modelling and evaluation of phased-mission systems. *IEEE Transactions on Reliability,* 48(4):360–368.

Mura, I. and Bondavalli, A. (2001). Markov regenerative stochastic Petri nets to model and evaluate the dependability of phased missions. *IEEE Transactions on Computers,* 50(12): 1337–1351.

NIST (2002). The economic impacts of inadequate infrastructure for software testing. Final report RTI Project Number 7007.011, US National Institute of Standards and Technology.

Porcarelli, S., Castaldi, M., Di Giandomenico, F., Bondavalli, A., and Inverardi, P. (2004). A framework for reconfiguration-based fault-tolerance in distributed systems. In De Lemos, R., Gacek, c., and Romanovsky, A., editors, *Architecting Dependable Systems,* LNCS. Springer-Verlag. To appear, also ICSE-WADS2003, Post-Proceeding of ICSE-WADS2003.

The Standish Group. `http://www.standishgroup.com/chaos`.