Chapter 1

Introduction

1.1 Goals

The following thesis aims to provide insight into threat assessment and mitigation, regarding ambient intelligence systems. In order to achieve this, several main discussion directions are proposed - architectural modifications, threat analysis, correction methods, detection schemes and feedback polynomial study.

Firstly, the proposed architecture for the system is network-based, being more versatile and providing an increased abstraction level to the final system. The network, or intelligent grid, is comprised of intelligent agents, each managing an individual resource, leading to more efficient workloads at the cost of dependability. Given that each resource being being used separately, with minimum oversight, there is no innate guarantee that a given resource is not malicious.

Because of the aforementioned security risk, a thorough threat assessment is done in order to ascertain system reliability. This can be achieved through an analysis of error occurrence, resulting in a threat model that will help understand and prevent future vulnerabilities. At the same time, it will also allow for quick response to any identified malicious resource used by an intelligent agent.

On top of threat detection, a correction step is included, were an error to be introduced into the system. Such errors can occur due to the transceiver nature of the intelligent grid nodes, with the messages sent between agents being corrupted by external factors. Because of this, from an architectural

point of view, the transmission layer should handle authentication and eventual error correction steps.

While threat assessment and error correction deal with the consequences of a potential attack, detection schemes manage threat identification. The proposed method of achieving such a scheme is to identify previously unknown attack patterns, associate it to a specific signature and compile all signatures into a dictionary, for future lookup, thus being able to check for the given signature in the future. In order to construct the detection scheme, a feedback polynomial has to be identified, based on the available metrics.

Finally, the most important aspect of this thesis is promoting the aforementioned ideas concerning the reliability of the intelligent grid, with regard to security and fault tolerance. The end goal is to begin a discussion about the design of a fully secured system, starting from the presented paradigms.

1.2 Design

Due to the presented intelligent grid being emulated on current hardware, it is required that the overall system is reliable, in terms of resource allocation and access, and inter-agent communication protocols. As such, the final design is meant for reliability, meaning that it takes into account deviations that appear between system and specification [1], [2].

Another aspect promoting a reliable system is that of cost, more specifically, the proportional relationship between cost and security - the investment has do double in order for the vulnerability count to halve. At the same time, the security errors and their effects can generate costs measured in billions of dollars [2], [3]. For this reason, the final design is done in accordance to the following principles:

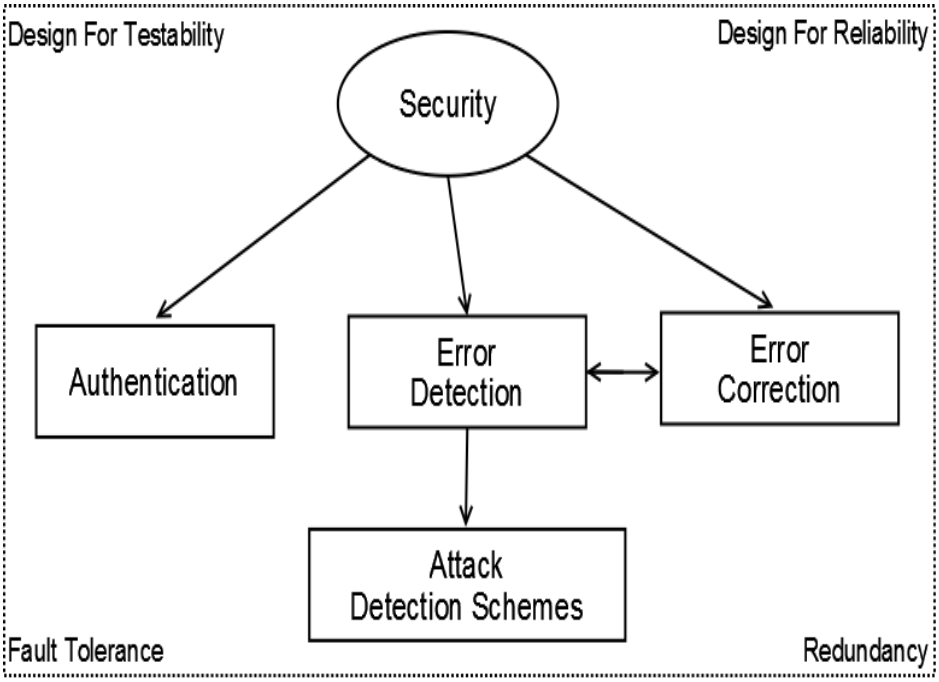

Figure 1.2.1: Security mechanisms and designing principles

As shown in fig. 1.2.1, the goals set for this thesis cover several security directions, and are doubled by design principles coupled with their respective mechanisms. The design aims to measure the improvements caused by different techniques, such that the final system is designed to be tested based on relative risk measures.

Chapter 2

Intelligent Grid

2.1 Description

The network design follows the industry tendency of mixing different types of networks, thus extending their functionality and reliability. For this reason, it is targeted towards the "smart" home, an environment that the intelligent system is meant control. The standard approach however, fails to account for an important component in the network - the PC. Formerly neglected by ambient intelligence systems, the PC is currently the linchpin of the multimedia ambient concepts, such as Microsoft Media Center, Apple FrontRow and HP Digital Entertainment Center. When it is in use, the general purpose computing system acts as both a sensor-actuator pair, controlling the network, as well as a source of computational power for the system to use, reducing overall power consumption [4].

The ‘intelligent grid‘ is the nomenclature for a collection of sub-components belonging to a multitude of fields regarding ambient intelligence, complementing each others ??. Those components are the derived from sensor networks, providing the ability to monitor and automate processes, multimedia networks, providing a plethora of I/O, storage and DSP capabilities, and classical computational networks, providing high performance computing from low cost components. All those parts are combined into a grid network, as it derives computational power most efficiently [5]–[12].

Based on the definition of ambient intelligence, it can be derived that the intelligent grid can be characterized by the following:

- the great majority of technology is embedded, hidden in the background
- is sensitive, adaptive, and responsive to the presence of people and objects
- that augments activities through smart non-explicit assistance
- that preserves security, privacy and trustworthiness while utilizing information when needed and appropriate
- it can accept explicit targeted tasks from the users
- it searches for computational power, being able to provide such a feature.

Stemming from those characteristics, the base schematic of the network can be presented:

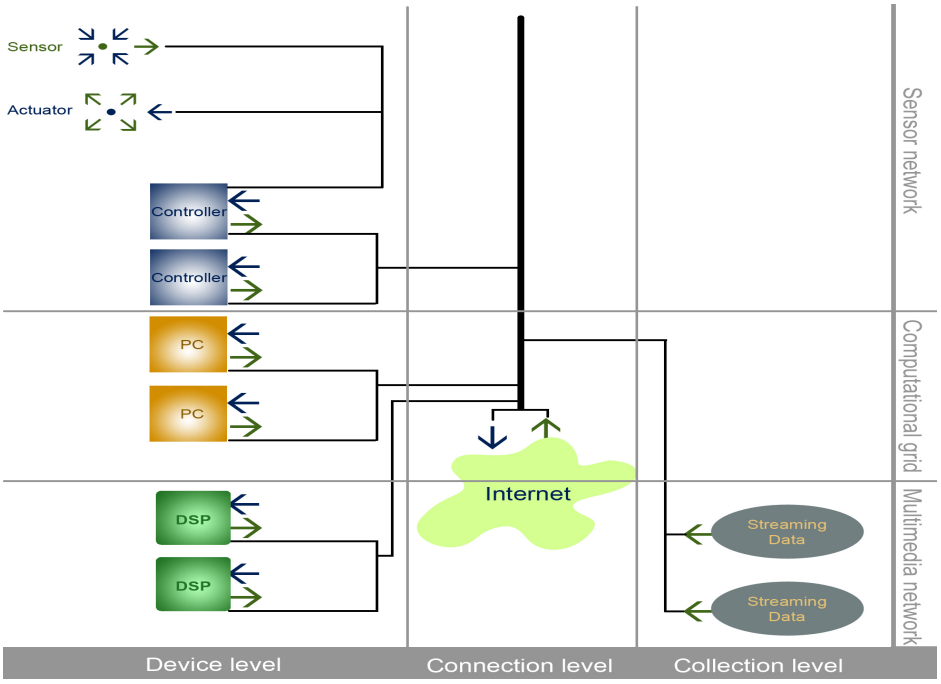

Figure 2.1.1: The intelligent grid. [13]

Each sub-network, abstraction level, and its respective equipment is showcased in fig. 2.1.1. Physical components, such as sensors, actuators, DSPs and PCs are located at device level, with the higher abstraction data storage

and streaming are denoted by the collection level. In between them, handling communication between the devices, as well as linking to the internet, is the connection level.

Due to all of the individual network components involved, the following table presents a comparison between the intelligent grid and previous network systems. It can be noted that it encompasses elements form all other networks, resulting in its ability to perform computations at the same time as it manages ambient intelligence.

Table 2.1: Comparison of networks [14]

Network Grid	Sensor grid	Ambient intelligence	Multimedia grid	Intelligent grid
Sensor Network	X	X		X
Multimedia Network		X	X	X
Computation Network	X		X	X

It is required that the network is designed for reliability, such as to ensure no errors occur when accessing resources or communicating between nodes. For this reason, each individual challenge present in ambient intelligence will be presented in regard to the intelligent grid, with specific solutions put forward in order to ensure a reliable design. Such a mechanism is represented by the “consensus issue“, an approach that is not a subject of this thesis, but is extensively covered in Versavia Ancusa’s PhD report [15].

2.2 Power, cost and size analysis

This approach is taken due to the nature of the hardware being used, such as sensors, actuators, controllers and DSPs, becoming "disappearing electronics", meaning super low-power devices that need to rely on a micro-generator instead of a battery for optimal functionality. Thus the need arises for components that can produce their own energy from ambient sources, called scavenging deceives. Such solutions are already available, most know being the harvesting of solar energy, but other sources are being covered, some relying on temperature, pressure, and even vibration gradients. Currently the

idea of energy harvesting is being considered from a system point of view, pushed by real industrial results, such as the parity between the current electrical grid and solar generated power [1], [16]–[18].

One such developing area is that of nanogenerators, such as nanoscale thermoelectric harvesting (based on the Seebeck effect), and nanopiezotronics (converting mechanical energy into electricity). Such systems are being developed in the field of MEMS (micro-electrico-mechanical systems) in order to converd nanoscale mechanical energy into electrical impulses through the use of zinc nanowires and a series of electrodes, producing electricity when the wires brush against the electrodes under sufficient vibration. [17], [19]

Other products, such as the battery-free wireless switches provided by 'Lightning Switch' and 'Ad Hoc Electronics' convert the button press into electrical potential. At the same time, battery-free tire sensors are being developed, powered by the impact acceleration converted through cymbal transducers. [1]

With all other bases covered, the only grid component left to be analyzed from a power, cost and size perspective is the PC, a component that expected to disappear, in favor of a more streamlined user experience. It is expected that future computers will provide only a basic interface, with the bulk of the computation being done on cloud infrastructure instead. **13**

2.3 Portability, scalability and configurability analysis

Those factors need to bbe accounted for because changing the application involves software changes, meaning that significant logistical errors can occur. The proposed solution is to raise the abstraction level [12]:

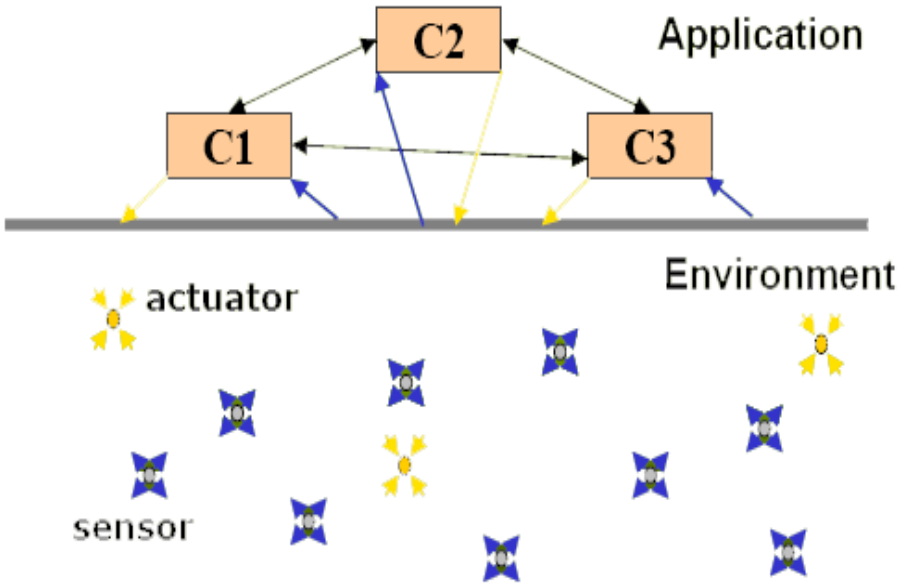

Figure 2.3.1: Raising abstraction level. [12]

Derived from fig. 2.3.1, an increase in abstraction level in the case of, for example, the sensor network, involves the management of individual nodes as a set of computational functions, with the cooperation between them being ensured by a series of sensors and actuators.

2.3.1 Middleware

Because of the myriad of computer systems running different operating systems and software tools, middleware is a required abstraction layer meant to provide a common end-point for a distributed heterogeneous system, in order to manage any increase in complexity.

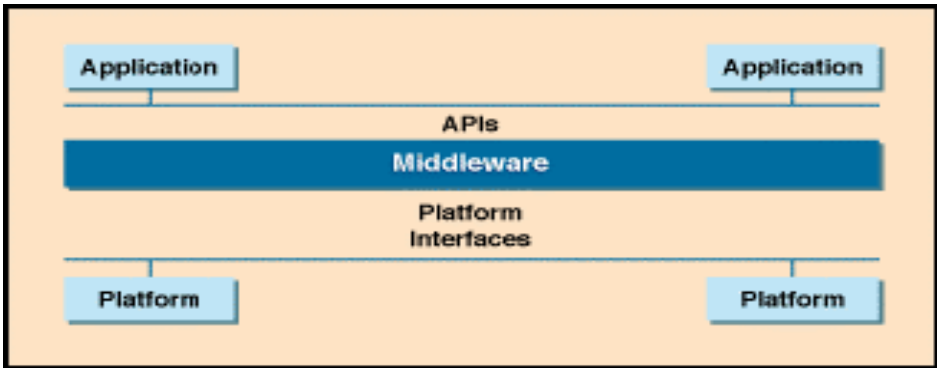

Figure 2.3.2: Middlewre layer. [20]

Middleware acts as a bridge between individual platform interfaces and higher level APIs, as presented in fig. 2.3.2, thus it needs to support compatibility with both neighboring layers. With the intelligent grid being comprised of three sub-networks, each of their requirements must be handled specifically:

Table 2.2: Overview of the programming requirements [14]

Programming requirements	Networks based on		
	Sensor	Multimedia	Computation
Concealed issues	hardware and distribution	hardware	distribution
Restricted Resources			
Energy	X		
computing power	X	X	X
communication bandwidth	X	X	X
Network Dynamics	high	medium	low
Scale of Deployments	N*(100...1000)	N*(10...100)	N*(10...100)
Real-world Integration			
Time scale	X	X	X
Location scale	X	X	
Collection and Processing of Data			
Preprocessing	X	X	
Aggregating data	X	X	X
Local processing		X	X

With table 2.2 presenting the requirements for each type of network, the following table showcases how existing types of middleware can be used for each type of sub-network in the system:

Table 2.3: Types of middleware and their usage [21]

Type of aproach	Networks based on		
	Sensor	Multimedia	Computation
Events	X		
Remote Procedure Call		X	X
Object Request Broker		X	X
Message-oriented			X
Databases	X	X	
Mobile (Intelligent) Agents	X	X	X

As seen in table 2.3, the most versatile type of middleware is that of "intelligent agents", independent system components that communicate with each other on a peer to peer basis. The concept of agents stems from the application of artificial intelligence to the field of distributed systems - Agent-Oriented Programming (AOP). The most widespread agent oriented middleware is the Java Agent Development framework (JADE), a completely distributed middleware system that supports easy extension via additional modules. The framework facilitates the development of complete agent-based applications by means of a run-time environment implementing the life-cycle support features required by agents, the core logic of agents themselves, and a rich suite of graphical tools. [22]

Being build in Java, JADE benefits form a myriad of third-party libraries, as well as basic language features, making it easily extensible, while also allowing for other abstraction layers to be build on top of it. Another significant advantage is that JADE is a fully open source project, adhering to FIPA specifications and IEEE standards.

2.3.2 Intelligent Agents

As a newly emerging technology, agent-based software solutions are at the forefront of scientific research, with significant effort put into commercial applications of the concept. Agents are, in essence, software components that can operate independently and can be linked together in order to increase each others capabilities. While single-agent systems can exist, the strength of the system derives from several agents exchanging messages, through specified protocols, in a collaborative manner. This communication enables coherency between the actions of all agents in the system, leading to more efficient workflow across the entire network [22], [23].

Multi-agent systems are already being used in a myriad of applications, from personal use to mission-critical industrial applications, such as process control, system diagnostics, manufacturing, logistics and network management. All agents are designed to be autonomous and reactive, meaning that they can operate without human intervention and adapt to changes in their environment. Other characteristics include proactivity, for agents that take goal-directed initiative in order to fulfill its function, and mobility, in the case of agents that are able to travel to different nodes in the network. Each agent can function as standalone applications, resource managers, network services and even APIs for lower level components. Those characteristics allow for the grid components, such as the controllers, DSPs and PCs, to be considered agents. Applying the concept of intelligent agents to the design from fig. 2.1.1 leads to the following schematic:

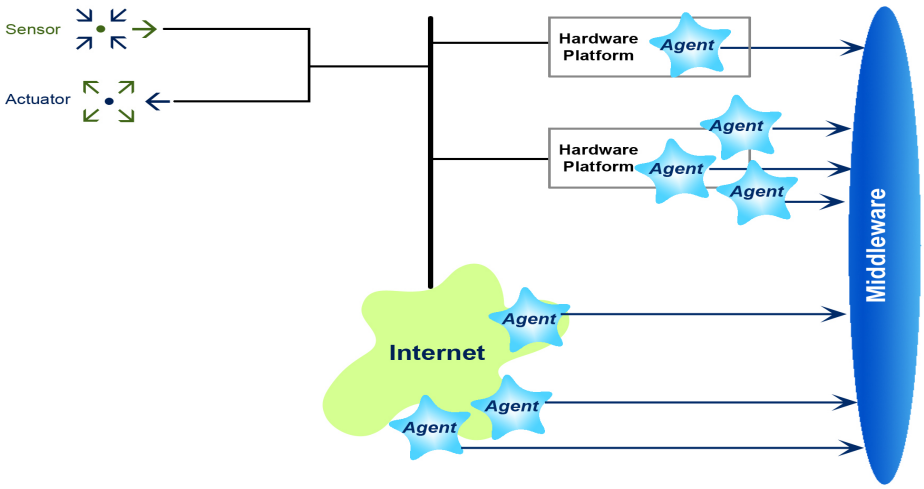

Figure 2.3.3: The intelligent grid invested with agents. [13]

Based on fig. 2.3.3, the device level contains only sensors and actuators, the connection level contains the middleware, and the collection level is represented by the agents. With the raise in abstraction level, the hardware aspects lose relevance - the only relevant aspect are the agents, with the mobility characteristic allowing for the physical components to be changed at any time in order to suit the problem requirements.

2.3.3 Middleware-based simulation for intelligent grid

The first relevant metric to be measured through simultaneous is the comparison between JADE and conventional middleware. This is done by modeling a simple problem to be solved in polynomial time by the same number of nodes in each context. The goal is to measure the time it takes for the application to complete, using both middlewares. The following graphs showcase the comparison between JADE and typical middleware (MIP).

?? Fig3.5 ??

?? Fig3.6 ??

From ?? Fig3.5 ?? it can be derived that the time increase in the case of standard middleware follows an exponential curve, due to the increase of problem inputs. At the same time, ?? Fig3.6 ?? presents the comparison between MIP and JADE on a constant number of inputs. The edge gained by the JADE system comes from the implementation using calls for proposal (CFP) - messages sent by agents every time they require access to a sensor, in order to get its recorded value.

It can be noted that MPI systems are ahead when the node count is low, however they are bottlenecked by the communication overhead after 25 nodes, the point at which the JADE system becomes more efficient. Another significant advantage for JADE is the portability of the code, allowing for more working environments.

2.4 Reliability

Due to the innate unreliability of disappearing electronics - caused by the ability of grid nodes to emerge, move, fail or running out of power - designing a reliable system becomes an unavoidable requirement in order to guarantee a secure system. This thesis will address different means of improving system security and dependability, foremost being the notion of redundancy.

2.4.1 Redundancy

At the connection level, the main concern is the energy consumption of the various sensors and actuators. Although they can be used in a myriad of low-power applications, increasing sensor density severely increases the power

consumption. An efficient method of combating this issue is to create node islands that are powered individually, which are turned off when not in use. While this approach is widely used in order to reduce energy consumption, it creates the challenge of introducing complexity into the system - one side-effect could be the creation of blind-spots that need to be accounted for. This creates the need for reliable heuristics, meant to maximize area coverage while also minimizing power draw [24].

Redundancy at link level is more challenging, as connection failures cannot be decoupled from component failures. In order to overcome this particular conundrum, a given component can be linked to several others, thus giving the possibility of determining whether a certain error was caused by a faulty connection or damaged equipment, while also serving as a mechanism for future fault tolerance. This approach, however, leads to increased costs, both in terms of wires and communication complexity, but is still a redundancy feature implemented into the intelligent grid [25].

An implementation of link level redundancy can be achieved if the controllers and sensors are coupled such that sensors in the same area are connected to the same controller, with a given amount of overlap between their coverage zone. Such a design is portrayed in the following:

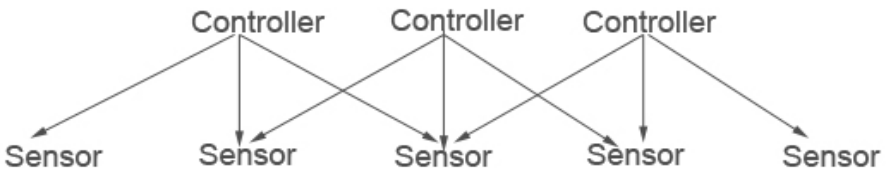

Figure 2.4.1: Redundancy at link level. [13]

The type of connection presented in fig. 2.4.1 leads to the marginal sensor areas being secluded, however it also causes an optimal proportion of redundancy to coverage in the middle section.

Built using the JADE middleware, each controller is managed by an agent that periodically requests the sensor values of all other agents, using those values to determine an action - in this particular case, setting a variable. The connections were made to be fail-stop, meaning that if an error occurs, no value is sent. Errors were then introduced, such that they would affect the maximum amount of lines possible, with the maximum number of errors in which a value from the sensor was received by the controller is given by:

$$max(Errors) = (Full\ covered\ sensonrs) * (Coverage - 1) \tag{2.1}$$

This architecture implies three independent variables, namely the number of controllers, the number of errors and the sensor redundancy to coverage proportion. In order to measure performance, the chosen metric was, again, computation time:

?? Fig3.8 ??

In ?? Fig3.8 ??, the variation of compute time is presented in relation to sensor redundancy, with the controller count remaining constant. It can be noted that the amount of errors introduced is irrelevant for the overall execution time, with the variation being described by:

$$y = \left(a + b * \frac{ln(x)}{x^2}\right)^{-1} \tag{2.2}$$

Relation eq. (2.2) is given by the fact that the measure of diversification, r^2, had a maximum value of 0.9992 and a minimum of 0.997, with the error count going from 0 to 7.

The following figure showcases the results obtained when only a fixed single error is introduced, but the sensor coverage is variable:

?? Fig3.10 ??

As seen in ?? Fig3.10 ??, compute time increases with sensor coverage, with the variation being best described by:

$$y = (a + b * x^2) \tag{2.3}$$

The final variable to be studied is the controller count, with the following figure showcasing this particular case:

?? Fig3.11 ??

Although incomplete, with portions in which no coverage could be obtained, due to eq. (2.1), from ?? Fig3.11 ?? can be derived that the closest approximation is the power function, with r^2 ranging from 0.993 to 0.9999:

$$y = (a + b * x^c) \tag{2.4}$$

An obvious weakness of this approach appears when attempting to use a wireless network, due to the lack of wires that would provide redundancy at the link level. A possible solution would be to create message redundancy, such as creating a generic and systematic way to transform various agreement problems into consensus, thus creating a unified framework to develop fault-tolerant agreement protocols in a modular, correct, and efficient way. An implementation of this principle on the intelligent grid would involve a starting value for each agent, and a common value that the agents have to agree on. More information can be fount in Versavia Ancusa's PhD report [15], [26].

To conclude, redundancy denotes a correlation between the maximum amount of errors and the minimum amount off sensors for sustainable system functionality. For this, it is proven that the implementation of the aforementioned redundancy paradigm is a viable solution in order to maintain a functional system, regardless of possible errors. In summary, sensor area redundancy relates to the devices, the proposed intelligent grid model implies inter-agent communication, the consensus-based protocol relates to the messages, and the proposed architecture ensures fault tolerant behavior.

References

[1] J. Rabaey, F. Burghardt, D. Steingart, M. Seeman, P. Wright, "Energy harvesting - a systems perspective," in *International Electron Devices Meeting (IEDM 2007)*, USA, 2007, 363–366.

[2] D. E. Geer Jr., D. G. Conway, "Security is a subset of reliability," *IEEE Security & Privacy*, vol. 6, no. 6, 2008.

[3] M. Zhivich, R. K. Cunningham, "The real cost of software errors," *IEEE Security & Privacy*, vol. 7, no. 2, 87–90, 2009.

[4] W. Weber, J. M. Rabaey, E. Aerts, *Ambient Intelligence*. Springer-Verlag, 2007, ISBN: 978-3-540-23867-6.

[5] A. Abbas, *Grid Computing: A Practical Guide to Technology and Applications*. Charles River Media, 2004, ISBN: 1584502762.

[6] Anirban Chakrabarti, A. Damodarani, S. Sengupta, "Grid computing security. a taxonomy," in *IEEE Security & Privacy*, vol. 6, 2008.

[7] A. Chakrabarti, *Grid Computing Security*. Springer-Verlag, 2007, ISBN: 978-3-540-44492-3.

[8] D. Petcu, *Arhitecturi si tehnologii grid*. Eubeea, 2006, 118–165.

[9] Ian Foster, C. Kesselman, *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, 1998, ISBN: 1-55860-475-8.

[10] Ian Foster, *What is the Grid? A Three Point Checklist*. Argonne National Laboratory & University of Chicago, 2002.

[11] Ian Foster, C. Kesselman, S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," in *International J. Supercomputer Applications*, vol. 15, 2001.

[12] J. Rabaey, "Wireless sensor and consumer multimedia networks – a story of converging trajectories?" In *IEEE Consumer Communications & Networking Conference (CCNC 2005)*, 2005.

[13] V. Ancusa, R. Bogdan, M. Vladutiu, "Discussing redundancy issues in intelligent agent-based non-traditional grids," in *Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES08)*, vol. LNAI 5178, Croatia, 2008, 297–395.

[14] V. Ancusa, R. Bogdan, M. Vladutiu, "Discussing the intelligent agent approach in non-traditional grids," in *Proceedings of the International Multi-Conference on Engineering and Technological Innovation*, vol. I, Florida, USA, 2008, 87–92, ISBN: 978-1-934272-46-6.

[15] V. Ancusa, *Problema consensului in calcul tolerant la erori*. Politehnica University of Timisoara, 2009, PhD Thesis.

[16] R. Swanson, "Future developments in silicon solar cells," in *International Electron Devices Meeting (IEDM 2007)*, USA, 2007, 359–362.

[17] R.Venkatasubramanian, Cynthia Watkins, David Stokes, John Posthill and Chris Caylor, "Energy harvesting for electronics with thermoelectric devices using nanoscale materials," in *International Electron Devices Meeting (IEDM 2007)*, USA, 2007, 367–370.

[18] W. Weber, C. Braun, J. Dienstuhl, R. Glaser, Y. Gsottberger, B. Knoll, C. Lauterbach, D. Leitner, M. X. Shi, M. Schnell, D. Savio, G. Stromberg and M. Verbeck, "Disappearing electronics and the return of the physical world," in *IEEE International Symposium on VLSI Technology (VLSI-TSA-Tech)*, 2005, 45–48.

[19] Eric M. Yeatman, Paul D. Mitcheson, Andrew S. Holmes, "Micro-engineered devices for motion energy harvesting," in *International Electron Devices Meeting (IEDM 2007)*, USA, 2007, 375–378.

[20] Rodica Tirtea, *Integration at middleware level of fault tolerance for distributed systems*. Katholieke Universiteit Leuven, 2005, PhD Thesis.

[21] V. Ancusa, R. Bogdan, M. Vladutiu, "Redundancy at link level for non-traditional grids implemented with intelligent agents," in *Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management (NCM08)*, vol. 1, South Korea, 2008, 597–603, ISBN: 978-0-7695-3322-3.

[22] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, *Developing multi-agent systems with JADE*. John Wiley & Sons, 2007, ISBN: 978-0-470-05747-6 (HB).

[23] M. N. Huhns, L. M. Stephens. "Multiagent Systems and Societies of Agents." (1999), [Online]. Available: `http://citeseerx.ist.psu.edu/`.

[24] Alexandru Coman, Mario A. Nascimento and Jorg Sandera, "Exploiting redundancy in sensor networks for energy efficient processing of spatiotemporal region queries," in *ACM Proceedings of the Conference on Information and Knowledge Management*, Bremen, Germany, 2005.

[25] M. P. Leslie Lamport and R. Shostak, "Reaching agreement in the presence of faults," *Journal of the Association for Computing Machinery*, 1980.

[26] A. S. R. Guerraoui, "The generic consensus service," *IEEE Transactions on Software Engineering*, vol. 27, no. 1, 2001.