Extreme Sensitive Robotic
A Context-aware Ubiquitous Learning

Nicolas Verstaevel[1,2], Christine Régis[1], Valérian Guivarch[1],
Marie-Pierre Gleizes[1] and Fabrice Robert[2]
[1]*IRIT, Université Paul Sabatier, Toulouse, France*
[2]*Sogeti High Tech, Toulouse, France*

Keywords: Ambient Intelligence, Self-organizing Systems, Machine Learning, Adaptive Multi-Agent Systems, Robot and Multi-robot Systems.

Abstract: Our work focuses on *Extreme Sensitive Robotic* that is on multi-robot applications that are in strong interaction with humans and their integration in a highly connected world. Because human-robots interactions have to be as natural as possible, we propose an approach where robots *Learn from Demonstrations*, memorize contexts of learning and self-organize their parts to adapt themselves to new contexts. To deal with *Extreme Sensitive Robotic*, we propose to use both an *Adaptive Multi-Agent System* (AMAS) approach and a *Context-Learning* pattern in order to build a multi-agent system ALEX (Adaptive Learner by Experiments) for contextual learning from demonstrations.

1 INTRODUCTION

The drastic reduction in the cost of electronic equipment allows populating our environment with a multitude of devices and functions of rich interaction capabilities. Information technologies, formerly confined inside computers, are now distributed in our homes, factories and companies. Those ambient systems (also called *ubiquitous systems*) are characterised by their dynamic and their complexity: a huge number of heterogeneous devices evolves autonomously and new devices can appear or disappear at any time (Perera, 2014). One of the desired properties for such systems is the ability to self-adapt to the specific and changing needs of its users. Nevertheless, such adaptation is complex since we can make no a priori supposition on the task to perform or on the entities composing the system. Furthermore, users in these systems can interact in several ways and this interaction brings both new challenges and new solutions.

More and more works among the robotic community focus on the design of physically distributed applications where autonomous robots interact with other systems to perform complex and changing tasks in interaction with humans (Brambilla, 2013). Factories of Future (FoF) are a good illustration of this as they involve multiple entities evolving in a complex and highly dynamical environment with a need for sustainability and adaptability to end-users (Siciliano, 2014). Hyper-connectivity of FoF offers a new challenge to their designers: how to handle the complexity and dynamic brought by the affluence of data coming from electronic systems and human activity.

In this paper, we propose an approach named *Extreme Sensitive Robotic* to use the inherent interactivity of ambient systems as the motor of self-adaptation. The system learns the way users interact with it. With this approach, the design is not guided by the finality, but by the system's functionalities. Each functionality is then seen as an autonomous system having the capacity to self-adapt its behaviour to what it can perceive from its environment (including human activity). Self-observation capacities allow each functionality to correlate its own activity to the observation it makes from its environment.

First, we consider the use of Learning from Demonstration, a paradigm to dynamically learn new behaviours (Argall, 2009), mainly studied in the robotic field. Next, we will dress our vision of *Extreme Sensitive Robotic* and propose the use of the *Adaptive Multi-Agent Systems* (AMAS) to achieve this vision.

Verstaevel N., Régis C., Guivarch V., Gleizes M. and Robert F..
Extreme Sensitive Robotic - A Context-Aware Ubiquitous Learning.
DOI: 10.5220/0005282002420248
In *Proceedings of the International Conference on Agents and Artificial Intelligence* (ICAART-2015), pages 242-248
ISBN: 978-989-758-073-4

2 LEARNING FROM DEMONSTRATIONS

Since we consider exploiting the interactivity of ambient systems, we need a user-centred approach. Observing the field of application made in robotic, one approach has retained our attention. Learning from Demonstrations (LfD, also called Imitation Learning or Programming by Demonstrations) is a paradigm to allow a system to autonomously learn new behaviours from demonstrations (Argall, 2009). The hypothesis is that a system can learn its behaviour from the observation of a human's activity. To adapt behaviour to specific needs, the classical approach is to decompose the task to realise into sub-tasks and to manually program the resolution of each sub-task. At the opposite, the LfD approach proposes that a new task can be derived from the observation of human activity (Calinon, 2008). As it is a social learning that requires for the user no expertise on the system (Dautenhahn, 2003), LfD seems to be a good paradigm when the task to perform is not *a priori* known.

The idea of LfD took its origins in the 80s, thus there has been a resurgent work on the subject (Argall, 2009) (Calinon, 2008). Applications of LfD are various, like learning to fly (Sammut, 2002), learning to react to an artefact (Knox, 2013) or learning manipulation task in an industrial environment (Tosello, 2014).

Despite its variety, LfD suffers from some limitations in term of genericity. Indeed, in the variety of LfD applications, we can distinguish two families of algorithms: those using *supervised learning* and those using *reinforcement learning*. Both technics have their pros and cons.

Supervised learning algorithms use labelled data to construct a decision model. Once this model is constructed, the system is able to autonomously follow a decision process. These algorithms include the notion of demonstration, each labelled data acting as a demonstration and seem relevant in the LfD context. However, they are not robust to new demonstrations. Indeed, the two-step operating mode separating learning phase from behavioural phase restricts the use of these algorithms for real-time learning. A new demonstration implies to repeat the complete learning phase.

Reinforcement learning algorithms use a feedback function associating the performance of an action to a utility value. A *reinforcement learning* algorithm tries to establish an optimal control of its environment in order to maximise this utility value. The main advantage of these algorithms is that they explore autonomously and in real-time the possible states to find an optimal control policy. Nevertheless, this advantage can also become a default since to learn not to destroy itself, a robot needs to experiment its destruction and receive a negative feedback in response. Furthermore, those algorithms suffer from a lack of genericity, since the design of the feedback function requires knowledge on the task to perform and the environment.

Some hybrid techniques try to use both algorithms in the same approach. For example, (Knox, 2013) infers a feedback function from feedbacks coming from the user and then uses this learned function to make *reinforcement learning*. However, these approaches, which attempt to remove limitations by combining reinforcement learning and supervised learning, often simply combine their limitations.

One of the current challenge in this domain is then to propose a generic approach, independent to the task to perform and able to self-adapt in real time both to the specific needs of its users and to the dynamic of its environment. However, we see that LfD is an interesting paradigm for service robotics as it is user-centred and allows day-to-day interactions.

(Nehaniv, 2001) postulates that LfD has to answer the following key questions:

- **What** to imitate?
- **How** to imitate?
- **When** to imitate?
- **Whom** to imitate?

On the next section, we propose to answer to those questions with an approach that is not guided by the goal to achieve, but by the functionalities composing the system.

3 EXTREME SENSITIVE ROBOTIC

Since their beginning, robots become ever more elaborated in terms of both hardware and software. It may be considered that these systems will be truly complex, complexity increased by their necessary adaptation to the high dynamic of their environment (including humans) and a dynamical coordination with other robots or artificial ambient systems. This is the presupposition made by *Extreme Sensitive Robotic* where the system's design should be made bottom-up rather than top-down. We have at our disposal libraries of various components realising functions rather than objectives. Thus, a robot consists of the aggregation of necessary functions to satisfy user's needs, but a group of robots has to be considered exactly in the same way: a set of macro-functions (each robot) working in coordination. An *Extreme Sensitive Robot* is made of simpler *Extreme Sensitive Functions*, each of these functions having the ability to self-adapt to what it can perceive from its environment. Thereby, the overall activity results from local interactions between *Extreme Sensitive Functions* and the environment. Such adaptation requires both openness and self-observation capacities, well known in the multi-agent field.

Kaminka's previous work focused on the necessity to roboticists and Multi-Agent System community to work together as they share same interests into autonomous decisions in complex environment (Kaminka, 2012). As a matter of fact, there has been an increasing work considering robots as agents into multi-robot application for complex tasks such as collective rescue (Lacouture, 2012) (Couceiro, 2013), collective exploration or tasks allocation (Navarro, 2012). A new challenge that rises among robotic applications is the integration of robots into smart environments where many heterogeneous devices are in hyper-interactivity with other systems and humans (Broxvall, 2006). In these systems, robots have the challenging task to use this hyper-connectivity to adapt their behaviour to achieve complex and changing goals. The traditional reductionist approach is not relevant for such systems where no assumption can be made on goals to achieve or the dynamic of the environment. *Extreme Sensitive Robotic* proposes that interactivity of such systems is more related to an autonomous observation of the dynamic of the surrounding environment (including the consequences of its own mobility) than the explicit communication between system entities. The absence of this explicit communication reduces the need for *a priori* knowledge on the system and allows each functionality to be designed separately. Self-observation capacities make the system extremely sensitive to its environment allowing it to integrate changes in its environment into its decision process. In the remainder of this section, we focus on important features that should be taken into account within *Extreme Sensitive Robotic*.

The architecture of a robot is globally composed of functions of perception, action and decision. A robot that performs well has to permanently make cooperate these three functions in association with a loop-back correlating the consequences of its own actions with the observation of changes on its surrounding environment. This is what Brooks (Brooks, 1990) expressed as the *physical grounding hypothesis.* In opposition to the classical reductionist approach, the *physical grounding hypothesis* postulates that physical interactions with the environment are the primary source of constraints for the design of intelligent systems. Thus, there is no need of symbolic representation of the environment leading to a complex decision making. On the contrary, the system's behaviour is a reaction to a stimulus coming from its environment. Brooks's *subsumption architecture* (Brooks, 1987) is the origin of behaviour-based robotics. In Brooks's architecture, a robot controller is built layer by layer, each layer responsible for one behaviour. The subsumption architecture enables the robot to select the most adequate layer in reaction to what it perceives from its surrounding environment. The postulate that can be made of Brooks's work is that direct interactions with the environment have a strong influence on the robot's decision process. Making an *Extreme Sensitive Robotic* consists in making it sensible to variations in the perception of its surrounding environment and not to an internal state representation. *Extreme Sensitive Robotic* is all about to sense, not to model. Pioneering work of Walter Grey (Walter, 1950) (Walter, 1951) in early 50s has shown that even without any form of computational intelligence, a machine can produce a behaviour that one can consider as a smart behaviour, even showing some learning skills. In Grey's robots, an active interaction between sensors and actuators allows a strong interaction with the surrounding environment and the emergence of a behaviour. Braitenberg (Braitenberg, 1986) proposed a set of vehicles where sensors are in direct interaction with actuators. The sensors could have an exciting or inhibitory influence on the actuators. With an exciting influence, more the sensor is excited more the actuator is excited. On the contrary, with an inhibitory influence, more the sensor is excited, less the actuator is excited. Depending of the type of influence and how sensors and actuators are connected, the same robot (same actuators and same sensors) could perform radically different behaviours. The only difference lies in how sensors and actuators are connected. It results that the robotic entity is not only influenced by its surrounding environment but also by the nature of the influence between sensors and actuators. Making an *Extreme Sensitive Robotic* is then considering what occurs both outside and inside the robot's body. Pfeifer (Pfeifer, 2002) has named the relation between an entity and its environment as the embodiment relation. Pfeifer postulates that the behaviour of an entity is highly influenced by the environment in which it is immersed but also by its own body. To illustrate this phenomenon, Pfeifer proposed the following experiment: looking at the trajectory of an ant walking on rocks, one could say that the behaviour of the ant is smart. Indeed, as the ant is avoiding obstacles, the trajectory appears to be complex. However, if this ant were a thousand times larger, the ant would not be blocked by stones anymore and would then walk in a straight line. The same observer would then say that the ant behaviour is not smart any more. Whereas there has been no change on the ant's mind or on the environment, the observed behaviour differs. A change in the ant body has changed the effect its body produces on the environment. This philosophical experiment shows that any changes in the body could radically change the relation between an entity and its environment.

The same idea has to be applied to robotic because some part of a robot could disappear (for example, a sensor failure) or functionalities added during robot's activity. Even two robots with the same architecture can have electronic differences such as a motor rotating faster than the other does. These modifications of robot's body could have a strong impact on consequences of robot actions on the environment. Making an *Extreme Sensitive Robotic* consists in making it sensitive to the effects that their actions have on the environment.

Thus, building an extremely sensitive robot is then to make it sensitive and adaptive to:

- How its environment evolves,
- How its functionalities interact,
- Appearance or disappearance of functionalities and their effects on the environment.

Unlike traditional robotic approach, which consists in building robust controllers for robotic platforms, the *Extreme Sensitive Robotic* approach deals with functionalities. Each functionality is designed to be self-adaptive, as self-adaptation is driven by a local observation of the environment. Thus, a robot is a set of functionalities interacting through the environment.

As each functionality has to correlate its own activity to both users and the observation of the environment, the use of LfD seems here relevant. Thus, answering (Nehaniv, 2001) questions would be:

- **What:** how users use system functionalities.
- **How:** by correlating the performance of an action to the observation of the environment.
- **When:** each time a user uses a functionality.
- **Who:** whoever has to act on a functionality.

To enable these functionalities to self-adapt, we propose to use the *Adaptive Multi-Agent System* theory, which is presented now.

4 AMAS THEORY

4.1 Landscape

The Adaptive Multi-Agent System theory (Capera, 2003) addresses the problematic of complex systems with a bottom-up approach where the concept of cooperation is the core of self-organisation. The theorem of functional adequacy (Camps, 1998) states that for all functionally adequate systems, there is at least one system with a cooperative internal state that realizes the same function in the same environment. A general definition of cooperation could be the golden mean between altruism and selfishness (Picard, 2005). Three mechanisms allow repairing an uncooperative state (Capera, 2003):

- **Tuning:** the agent adjusts its internal state to modify its behaviour,
- **Reorganisation:** the agent modifies the way it interacts with its neighbourhood,
- **Evolution:** the agent can create other agents or self-suppress when there is no other agent to produce a functionality or when a functionality is useless.

The system will then self-organise to stay in a cooperative state. From cooperative interactions between the system's entities emerges a global function that is more than the sum of the parts. This theory is applied in *Extreme Sensitive Robotic* with the *Context-Learning* pattern.

4.2 Context-learning Pattern

The term *context* refers to all information external to the activity of an entity that affects its activity. This set of information describes the environment as the entity sees it (Guivarch, 2012). *Context-Learning* is based on the idea that the activity of an entity is correlated with the observation made by the entity of its own context. Thus, when an action is performed on (or by) an entity, this entity can make a correlation between the performance of this action in the current context and the effects of this action and then, learns the relevance of this action in this particular context. So the entity becomes able to correlate what it *feels* to what it *does* and to reuse its knowledge when it is confronted to an already known context. The entity is then called *context-aware*, which means that it is able to perceive, interpret and use the information from its current context in order to dynamically adapt its functionality.

The *Context-Learning* pattern has been applied to the control of bioprocess (Videau, 2011), the control of engine and energy (Boes, 2013) and the observation of users activity (Guivarch, 2012). In the next section, we will present general principles of the *Context-Learning* approach, based on an Adaptive Multi-Agent System (AMAS).

4.3 Context-learning Principle

The *Context-Learning* process is the result of two kinds of agents, each one being responsible for a particular activity in the system:

- A *Context Agent* associates a low-level context description (See section 4.4) to an action proposal. It receives signals from the environment and uses them to characterize the context. When the current context belongs to the context description of a Context Agent, this agent considers itself as *valid*, which means the current context is relevant to make an action proposal to its associated Controller Agent. The action proposition is composed of the action description itself, and information about the relevance level to perform this action. After each proposal, it receives a positive feedback from its associated controller if the action is selected or a negative one if the action is not selected. The role of the Context Agent is then to self-adapt to feedbacks from its associated Controller Agent by modifying its situation description or by adjusting the information of its action proposition.
- A *Controller Agent* is associated with each controllable variable of the environment and controls the modification of this variable to produce an adequate behaviour. In order to do this, it receives action proposals from Context Agents. Then, it selects the best action proposal, performs this action and observes the impact of this action (this observation is domain dependant) to send feedbacks to Context Agents. It is also responsible of Context Agents creation when there is no relevant Context Agent. (See section 4.4).

Learning is then the result of a self-organisation process inside Context Agents as each Context Agent dynamically adjusts its validity domain in reaction to feedbacks from the Controller Agent.

4.4 Context-learning Formalism

A Context Agent receives signals from its environment and uses these signals to describe the current context.

Definition 1: Let $s \in S$ where S is a set of signals and $s \subset [s_{min}, s_{max}]$ a signal such as $s_{min}, s_{max} \in \mathbb{R}$.

A low-level context description of a Context Agent is made with validity ranges.

Definition 2: A validity range v_s associated to a value $s \in S$ is a range $[v_{min_s}, v_{max_s}]$ where $[v_{min_s}, v_{max_s}] \subset [s_{min}, s_{max}]$.

A validity range allows the Context Agent to compare the current signal value to its associated validity range and to decide if it is relevant to send an action proposal.

Definition 3: A validity range v_s is $valid$ if and only if $s \in [v_{min_s}, v_{max_s}]$.

For each received signal, a Context Agent creates an associated validity range. The set of validity ranges is then called a *validity domain*.

Definition 4: Let $c \in C$ a Context Agent. Let V_c a validity domain associated to a Context Agent c. V_c is a set of validity range such as $\forall s \in S, \exists v_s \epsilon V_c$.

To determine its validity state, a Context Agent verifies each validity range of its validity domain.

Definition 5: $c \in C$ is $valid$ if and only if $\forall v_s \epsilon V_c, v_s$ is $valid, invalid$ otherwise.

The role of the Context Agent is then to dynamically adjust its validity domain to feedbacks from its Controller Agent in order to be *valid* when its associated action is relevant. The main advantage of this context description is that it uses no semantic: only variations of signal value are observed and no *a priori* is made on signal value meaning.

4.5 Context-learning Properties

The *Context-Learning* pattern presents interesting properties:

- **On-line learning:** the process of self-organisation (especially the creation and tuning of the Context Agents) is performed at runtime on the fly without necessity to stop it or reboot it;
- **Openness:** new variables can easily be added in the system thanks to the low-level description of situations in the Context Agents. Moreover, new functionalities can also be easily added because of the independence between the learning processes of each controllable variable;
- **Generic:** the *Context-Learning* pattern uses no semantic on the signals perceived neither on the controlled system, making it highly generic.

4.6 Context-learning Modes

The Controller Agent in charge of the Context Agents creation can apply either a supervised strategy, either a reinforcement strategy to explore the possible states.

- **Supervised strategy:** the Controller Agent bases its Context Agents creation on the observation of the actions performed by an entity (for example, a user). When this entity performs an action whereas there was no Context Agent that proposed this action, the Controller Agent observes it and creates a Context Agent in order to represent this action associated with the current situation. With this strategy, the system only performs actions previously observed;
- **Reinforcement strategy:** the Controller Agent generates Context Agents by itself for each situation where the current set of valid Context Agents is empty or composed of unsatisfying Context Agents. In this case, the Controller Agent applies different strategies in order to evaluate what seems to be the correct action to perform (for example, the same action as that of the previously selected Context Agent if it was a satisfying action, or else the opposite action).

In both case, self-observation is the engine of learning. These two strategies differ in the way they explore the possibilities space. In supervised learning, exploration is guided by an external entity whereas in reinforcement learning, exploration is guided by a trial/error process.

For more information on *Context-Learning* pattern, the reader can refer to previous works of (Guivarch, 2012) and (Boes, 2013).

5 ALEX: AN ADAPTIVE MULTI-AGENT SYSTEM FOR CONTEXTUAL-LEARNING FROM DEMONSTRATION

ALEX (Adaptive Learner by EXperiments) is an AMAS based on *the Context-Learning* pattern in respect with the *Extreme Sensitive Robotic* vision. It allows a real-time learning from the observation of user's activity in distributed applications. ALEX is able to control a device or a functionality by correlating actions performed by a tutor to the observations it makes of its own environment.

The main hypothesis made by ALEX is that when a user has to act on it, it is because the ongoing behaviour is not satisfying the user anymore. Thus, the system has to self-organise to reach a functionally adequate behaviour. The actions performed by the user will be relevant under the same context. Then, ALEX tries to learn all contextual actions performed by a tutor and to reproduce them.

An ALEX is an *Extreme Sensitive Function*. Its functionality could be the control of a high-level function (such as "Go back", "Turn left") or a low-level control on an effector (such as the rotating speed of a motor or its angular position). It receives signals from its surrounding environment that could come from sensors, other *Extreme Sensitive Functions*, or even humans. A signal is composed of a unique identifier and a value. The identifier has no specific significance and by consequence, it has no semantic. These signals are used to determine contexts.

When a user acts on it, the ALEX system analyses all signals to discover the current context. Then, it determines what action should have been performed if the user did not act and adapts the behaviour in response. Every time an ALEX performs an action, it communicates its new state as a new signal. Thus, each *Extreme Sensitive Function* can sense the activity of other within a communication range. An ALEX is then an autonomous controller trying to correlate the user activity to the observation of its own environment, including other ALEX.

ALEX is currently used and tested on multi-robot and multi-user ambient applications. Some examples of experiments can be viewed on one of the author's website (www.irit.fr/~Nicolas.Verstaevel/ALEX).

6 CONCLUSIONS

This paper presents *Extreme Sensitive Robotic*, an approach where the design is not guided by the goal to achieve but by the functionalities composing the system. The overall activity results from local interactions between each functionality.

It also presents ongoing work on Learning by Demonstration. More precisely, it illustrates how we are combining the *Extreme Sensitive Robotic* approach and the AMAS approach. We propose the use of the *Context-Learning* pattern to enable self-observation capacities in each system's functionality. It results in ALEX, a multi-agent system for contextual learning by demonstration that brings *Context-Awareness* in ambient systems. Functional prototypes have been developed and we now consider the application of our approach on concrete problems coming from industry.

REFERENCES

Argall B. D., Chernova S., Veloso M., & Browning B. (2009). *A survey of robot learning from demonstration*. Robotics and autonomous systems, 57(5), 469-483.

Braitenberg, V. (1986). *Vehicles: Experiments in synthetic psychology*. MIT press.

Brambilla M., Ferrante E., Birattari M., & Dorigo M. (2013). *Swarm robotics: a review from the swarm engineering perspective*. Swarm Intelligence, 7(1), 1-41.

Boes, J., Gatto, F., Glize, P., & Migeon, F. (2013). *Controlling Complex Systems Dynamics without Prior Model*. In ICAART (1) (pp. 389-392).

Brooks, R. A. (1987). *A hardware retargetable distributed layered architecture for mobile robot control*. In Robotics and Automation. Proceedings. 1987 IEEE International Conference on (Vol. 4, pp. 106-110).

Brooks, R. A. (1990). *Elephants don't play chess*. Robotics and autonomous systems, 6(1), 3-15.

Broxvall, M., Gritti, M., Saffiotti, A., Seo, B. S., & Cho, Y. J. (2006). *PEIS ecology: Integrating robots into smart environments*. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on (pp. 212-218).

Calinon, S. (2008). *Robot programming by demonstration*. In Springer handbook of robotics (pp. 1371-1394). Springer Berlin Heidelberg.

Camps V, (1998). *Application of a self-organizing method based on cooperation to information retrieval*. In Young researcher paper to European Conference on Artificial Intelligence (ECAI-98), Brighton, England.

Capera, D., Georgé, J., Gleizes, M. P., & Glize, P. (2003). *The AMAS theory for complex problem solving based on self-organizing cooperative agents*. In Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on (pp. 383-388).

Couceiro, M. S., Portugal, D., & Rocha, R. P. (2013). *A collective robotic architecture in search and rescue scenarios*. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 64-69).

Dautenhahn, K., Nehaniv, C. L., & Alissandrakis, (2003). *Learning by Experience from Others—Social Learning and Imitation in Animals and Robots*. In Adaptivity and Learning (pp. 217-241). Springer Berlin Heidelberg.

Guivarch, V., Camps, V., & Péninou, A. (2012). *Context awareness in ambient systems by an adaptive multi-agent approach*. In Ambient Intelligence (pp. 129-144). Springer Berlin Heidelberg.

Kaminka, G. A. (2012). *Autonomous Agents Research in. Robotics: A Report from the Trenches*. In AAAI Spring Symposium: Designing Intelligent Robots.

Knox W. B., Peter Stone, and Cynthia Breazeal, (2013). *Training a Robot via Human Feedback: A Case Study*. In International Conference on Social Robotics.

Lacouture, J., Gascueña, J. M., Gleizes, M. P., Glize, P., Garijo, F. J., & Fernández-Caballero, A. (2012). *Rosace: Agent-based systems for dynamic task allocation in crisis management*. In Advances on Practical Applications of Agents and Multi-Agent Systems (pp. 255-259). Springer Berlin Heidelberg.

Navarro, I., & Matía, F. (2012). *An introduction to swarm robotics*. ISRN Robotics, 2013.

Nehaniv, C. L., & Dautenhahn, K. (2001). *Like me?-measures of correspondence and imitation*. Cybernetics & Systems, 32(1-2), 11-51.

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). *Context aware computing for the internet of things: A survey*. Communications Surveys & Tutorials, IEEE, 16(1), 414-454.

Picard G and Glize P, (2005). *Model and Experiments of. Local Decision Based on Cooperative Self-Organization*. In second International Indian Conference on Artificial Intelligence (IICAI'05).

Pfeifer, R. (2002). *On the role of embodiment in the emergence of cognition: Grey Walter's turtles and beyond*. In Proc. of the Workshop "The Legacy of Grey Walter".

Sammut, C., Hurst, S., Kedzier, D. et al (2002). *Learning to Fly*. Imitation in animals and artifacts, p. 171.

Siciliano Bruno & al., (2014). *EuRoC - The Challenge Initiative for European Robotics* ISR/Robotik 2014. Proceedings of 41st International Symposium on Robotics, vol., no., pp.1,7, 2-3.

Tosello, E., Michieletto, S., Bisson, A., Pagello, E., & Menegatti, E. (2014). *A learning from demonstration framework for manipulation tasks*. In ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of (pp. 1-7).

Videau, S., Bernon, C., Glize, P., & Uribelarrea, J. L. (2011). *Controlling bioprocesses using cooperative self-organizing agents*. In Advances on Practical Applications of Agents and Multi-agent Systems (pp. 141-150). Springer Berlin Heidelberg.

Walter, W. G. (1950). *An imitation of Life*.

Walter, W. G. (1951). *A machine that learns*. Scientific American, 185(2), 60-63.