Towards The Adoption of a Perception-Driven Perspective in the Design of Complex Robotic Systems

Antonio Chella
Dip. di Ingegneria Informatica
University of Palermo
Viale delle Scienze
Palermo, Italy
chella@dinfo.unipa.it

Massimo Cossentino
ICAR-CNR, Consiglio
Nazionale delle Ricerche
Viale delle Scienze
Palermo, Italy
cossentino@pa.icar.cnr.it

Valeria Seidita
Dip. di Ingegneria Informatica
University of Palermo
Viale delle Scienze
Palermo, Italy
seidita@dinfo.unipa.it

ABSTRACT

Awareness and autonomous interaction with the environment in a robotic system is the base of the new discipline of *machine consciousness.*

In this paper we present the results of a first attempt in order to engineer these robotic systems by applying a Situational Method Engineering approach that extends PASSI and to create a model for conscious systems.

Keywords

Machine Consciousness, Design Process, Robotic System

1. INTRODUCTION

Latest years have seen a growing interest towards robotic systems able to interact with human beings and with a dynamic environment in an autonomous fashion without the need of external control. Perception is one of the most important feature that robotic systems must have in order to present these capabilities. Awareness and enactive capabilities of the environment are highly desirable but missing in the today state-of-the art robots.

In other words, a robotic system should be able to have some sort of *consciousness* of itself and of the external world. In [1], for the first time the term *machine consciuosness* was mentioned meaning the capability of having experience or being aware of what happen in the environment and until today several researches in the field have been developed [9].

What we are currently investigating is the engineering perspective of the new discipline of machine consciousness: how to cope with modelling, designing and implementing artificial conscious robotic system able to process the stimuli coming from the outer and inner world, to generate its own intentions and motivations and and to reach a specific goal following intentional plans.

The authors developed in the recent past some design processes for engineering robotic systems (PASSI and Agile PASSI) and a cognitive robotic architecture based on these design processes [7]. The cognitive architecture was also experimented on an autonomous robot where visual perception is integrated with knowledge representation [8][24].

Besides this architecture is the first attempt made for realizing what it is argued in [5], i.e., a perception process is modeled and implemented as a continuous interaction loop by continuously comparing actual and expected “data” coming from the environment. In this way, the robot achieves the ability to gain perceptual experience and to react by simulating a *conscious* behavior with respect to the external stimuli.

Engineering these robotic systems requires ad-hoc design processes; in this paper an experiment based on a well known Situational Method Engineering approach is presented; by applying it to the mentioned case the result is a set of portions of design process (fragments) to be used for extending PASSI, the design process we use in our laboratory.

The rest of the paper is organized as follows: section 2 deals with the background of our work, section 3 presents the obtained results, in section 4 the fragments are described and finally in section 5 some conclusions and future works are sketched.

2. THEORETICAL BACKGROUND

2.1 The Design of Robotic Systems

Robotic systems, and moreover robotic system for artificial consciousness, are very complex systems; in the latest years we used the agent oriented approach for developing them and we mainly used PASSI design process that allows the development of robotic systems made of a society of peer agents.

Several agent oriented design processes there exist today therefore it does not exist a unique design process useful to everyone without some kind of customisation.

In the past years we developed ad-hoc design methodologies by following the paradigm of Situational Method Engineering (SME) [22][3]; the first experiment we made was about the construction of an agile [2] methodology for the quick engineering of a robotic system avoiding all the - time consuming - phases resulting in the production of detailed documentation thus giving the designer the possibility to quickly go from the requirement to the code.

The result was Agile PASSI [6][13], created by reusing and adapting most parts from the preceding PASSI[10] and following the Situational Method Engineering process that in the latest two years we finalized [11][27].

The SME process (PRoDe) [27] we use is mainly based on the reuse of existing portions of design processes and on the early identification of the main design concepts each system requires, respectively the *process fragment* and the *metamodel.* For instance a class of systems - robotic systems in our case - envisaging the actuation of a set of plans for reaching a specific goal requires at design time a modeling phase for robot plans, for its goals and for matching each plan with the related goal. At the end we can see each class

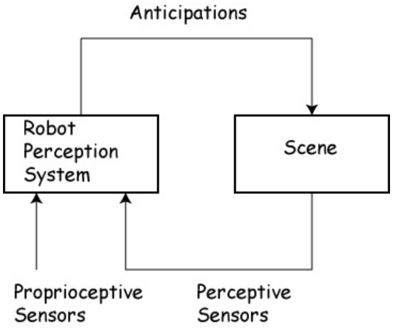

Figure 1: The perception loop.

of systems, for which we want to construct the most suitable design methodology, as a set of interrelated concepts to be designed following a collection of design phases.

PRoDe is composed of three main phases, *Process Requirements Analysis*, *Fragment Selection* and *Fragment Assembly*; following the same rationale of the software development we start from gathering the set of process requirements that contributes in creating the metamodel of the process to be that is the main element, in addition to the process designer's skills, for selecting the fragments from the repository and for assembling them.

Requirements are elicited on the base of the *problem type*, the *development context* and the *organization maturity*, they will be detailed in 3.1. This approach was created for the purpose of developing agent-oriented design processes but, due to the use of the metamodel, it proved to be general enough to be applied to every kind of application context.

In this paper we apply the PRoDe approach to the construction of a portion of design process that supports the development of a complex robotic system where decisions about robot's behavior are evaluated by means of a perception loop. The perception loop (described in the next section) uses a simulation module for forecasting decision consequences and defines a robot's expectation (called anticipation) of what is going to happen.

2.2 The Perception Loop

The robot perception loop, at the basis of the proposed robot architecture, is based on tight interactions between the robot brain, body and environment. The model is described in Figure 1; see [4] for a detailed description. Haikonen [15, 16] proposed a similar feedback loop in which the model of the environment is implicitly learned in terms of weights of an associative neural network. The loop is in turn the basic block of the Haikonen cognitive architecture for robot brains.

A robot system, able to build an internal model of the environment and to generate suitable predictions, has been proposed by Holland and Goodman [18]. The system is based on a neural network that controls a Khepera minirobot and it is able to build a model of environment and to simulate perceptual activities in a simple environment. Following the same principles, Holland et al. [19] present the robot CRONOS, a very complex *anthropomimetic* robot whose operations are controlled by SIMNOS, a 3D simulator of the robot itself and its environment. Hesslow [17] , from the standpoint of the neuroscience, discussed in details the role of inner simulations in relations with sensorimotor and cognitive functions.

The first order perception loop takes into account the schema proposed by Grush [14] and inspired to the Kalman filter as a model for perception process.

The robot vision system receives in input the robot position, speed and so on from the proprioceptive sensors and it generates the scene anticipations, i.e., the expectations about the perceived scene. The perception loop is then closed by the perceptive sensors that acquire the effective scene by means of the video camera.

Macaluso et al. [23] describe a robot based on a perception loop in which the process of scene anticipations is performed by a 3D computer graphics simulator. The simulator generates the expected 2D image scene on the basis of the robot movements.

In the proposed model, the mapping between the anticipated and the perceived scene is achieved through a *focus of attention* mechanism implemented by means of suitable recurrent neural networks with internal states. A sequential attentive mechanism is hypothesized that suitably scans the perceived scene and, according to the hypotheses generated on the basis of the anticipation mechanism, it predicts and detects the interesting events occurring in the scene.

Hence, starting from the incoming information, such a mechanism generates expectations and it makes contexts in which hypotheses may be verified and, if necessary, adjusted.

The focus of attention mechanism selects the relevant aspects of the acquired scene by sequentially scanning the image from the perceptive sensors and by comparing them in the generated anticipated scene. The attention mechanism is crucial in determining which portions of the acquired scene match with the generated anticipation scene: not all true (and possibly useless) matches are considered, but only those that are judged to be relevant on the basis of the attentive process.

The match of a certain part of the acquired scene with the anticipated one in a certain situation will elicit the anticipation of other parts of the same scene in the current situation. In this case, the mechanism seeks for the corresponding scene parts in the current anticipated scene. We call this type of anticipation *synchronic* because it refers to the same situation scene.

The recognition of certain scene parts could also elicit the anticipation of evolutions of the arrangements of parts in the scene; i.e., the mechanism generates the expectations for other scene parts in subsequent anticipated situation scenes. We call this anticipation *diachronic*, in the sense that it involves subsequent configurations of image scenes. It should be noted that diachronic anticipations can be related with a situation perceived as the precondition of an action, and the corresponding situation expected as the effect of the action itself. In this way diachronic anticipations can prefigure the situation resulting as the outcome of a robot action.

Two main sources of anticipation are taken into account. On the one side, anticipations are generated on the basis of the structural information stored in the robot by design. We call *phylogenetic* these kind of anticipations. On the other side, anticipations could also be generated by a purely Hebbian association between situations learned during the robot operations. We call *ontogenetic* this kind of anticipations. Both modalities contribute to the robot conscious perception process.

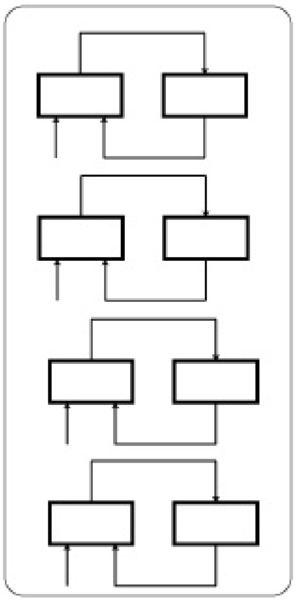

Figure 2: Many perception loops.

Ontogenetic anticipations are acquired by *online* learning and *offline* learning. During the normal robot operations, when something unexpected happens, i.e., when the generated anticipation image scene does not match the scene acquired by the perceptive sensors, the robot vision system learns to associate, by an Hebbian mechanism, the current image scene with the new anticipation image through the previously described attentional mechanism.

In the offline learning, the perception loop is employed to allow the robot to imagine future sequences of actions to generate and learn novel anticipations. The signal from perceptive sensors is related to the perception of a situation of the world out there. In this mode, the robot vision system freely generates anticipations of the perceptive sensors, i.e., it freely imagines possible evolutions of scenes and therefore possible interactions of the robot with the external world, without referring to a current external scene. In this way, new anticipations or new combinations of anticipations may be found and learned offline by the robot itself through the synchronic and diachronic attentional mechanisms.

In general, in a real operating robot, we may have many perception loops in action (Figure 2). They may be related with different sensor modalities, e.g., laser, video camera, sonar, and so on. Moreover, perception loops related with the same sensor modality may consider different aspects and operations, e.g., a vision based perception loop may consider some kind of objects while another vision based modality perception loop may consider free space. From this point of view, the perception loops play the role of *trackers* in the sense introduced by Kuipers [21] as the basic block of conscious perception.

3. TOWARDS THE NEW METHODOLOGY: FROM REQUIREMENTS ANALYSIS TO RESULTS

The aim of our work is to create a methodology for engineering and developing machine conscious architectures.

In this section we will illustrate the set of requirements resulting from the analysis of the class of systems we want to engineer with our new methodology and the metamodel resulting from this analysis.

3.1 Requirements for the construction of the new methodology

The fundamental requirement that the new process should fulfill consists in the support for the development of robotic systems basing their behavior on perception loops (or processes). A perception process can be modelled and implemented as a continuous interaction loop among brain, body and environment; by continuously comparing actual and expected data coming from the environment the robot achieves the ability to gain perceptual experience and to react by simulating a conscious behavior to the external stimuli. Besides in [5] it is argued that the robot possesses two (or more) orders of perception, the first one serves for the immediate sensing of the world around whereas other loops (higher order perception loops) may be used for discriminating the robot's inner world. During each perception loop the robot perception system generates an anticipation (i.e. an expectation) starting from its knowledge of the environment (including both the world model and the self model) and it composes the set of goals to be reached.

According to the input of the process requirements analysis, the requirements of the design process under construction can be summarized as follows:

Problem Type

1. Developing a robotic architecture composed of two main levels of abstraction: one or more robots, and inside each robot, a society of agents responsible for the basic robot's functionalities (for instance sensors management, vision, . . .).
    - A robot is composed of:
        - Rational agents: agents with reasoning/planning capabilities and a knowledge of the world.
        - Reactive agents: agents adopting the stimulus-reaction loop.
        - Devices: artifacts [25] representing robot's hardware components.
        - Conscious agent: an agent providing self consciousness features to the robot
    - Each robot can interact with other robots, the objects in the world (regarded as artifacts) and external agents; several robots can form a society of robots. This entails:
2. The robot has to be endowed with the capability of perceiving the environment around it and its own inner world.
    - The robot has the ability of recognizing and distinguishing stimuli coming from the outer word (sensorial perception) and stimuli coming from the robot body (proprioceptive sense).

- Perception in supervised by means of perception loops. The conscious agent is responsible for the execution of the perception loop that is composed of the following step:
    - The robot perceives the outer world (sensors) and the inner world (propioceptive perceptions)
    - Perceptions are used to build a 3-D simulation of the mission (anticipation)
    - Anticipation is compared with the perceived scene during mission execution
    - Parameters are tuned according to results of that matching
- Several perception loops can be active at the same time for taking care of different aspects of robot management.

3. The robot moves in an unstructured environment and it is able to autonomously interact with it. This entails that:
    - The robot has a model of the environment;
    - The robot owns a model of the "self";
    - The environment is composed of objects that can be agents and artifacts - an artifact is a passive, function-oriented entity with no means of autonomy and control encapsulation.
4. The metaphor of software testing is adopted for detecting differences between expected and observed behaviour. This detection process represents the robot conscious reasoning. In order to fruitfully adopt the testing metaphor, we suppose that:
    - While object oriented test activities are devoted to test the different software components in order to detect errors with respect to the software scenarios, in our case the scenario is perceived by the robot (the observed behaviour) and it is compared with the expected behaviour (anticipation of the perception loop) in order to detect a not-expected situation and to activate all the fixing actions.
    - The test planning result is the simulation of the actions the robot performs in order to perceive a goal; this defines the expected behaviour - the *oracle* attribute of the *test case* defined in each test planning activity.

Development Context. The system will be developed in a research lab by people skilled with robot programming and multi-agent system concepts. Developers would prefer to spend their time in defining perception loop aspects, a conceptual model of the world and the profile of the robotic mission rather than looking at the definition of complex MAS social structures or other implementation details.
The reuse of previous portions of design as well as of code is very welcome and according to a consolidated tradition is left to an extensive pattern-reuse practice [12].

Organization Maturity. Several experiments has been made in the past in the construction of methodologies for designing robotic systems and several systems have been developed in this lab on the basis of the well known PASSI process as well as its extensions (Agile PASSI and more recently ASPECS).
Looking at the development group in its entireness, it has to be considered that together with people who experienced the cited past projects and researches, there are young new lab members which only recently approached the development of robotic systems by using MAS technologies. Moreover, the adoption of a perception-driven perspective is new to all lab members and specific guidelines will be necessary to improve products quality.

3.2 From Process Requirements to MAS metamodel

According to the PRoDe approach, the above reported requirements are used to define the MAS metamodel. This is achieved by identifying proper strategies in order to fulfil the new process requirements. Finally these strategies are bound to MAS metamodel elements that can concur to their realisation. A detailed description of the process we used to define strategies from requirements and then MAS metamodel elements from strategies is out of the scope of this paper. It has been partially described in [27] and some details like the definition of specific guidelines are still a work in progress. Table 1 reports a partial list of the new process requirements, descending strategies and adopted MAS metamodel elements (MMME). It is interesting to see that the second strategy (*The robot is the unique conscious agent endowed with self-reflective ability*) brings to the identification of a couple of MMMEs (Conscious Agent, Robot) as well as to a constraint on the number of instances of one element (each robot can have only one Conscious Agent that implements its self-reflective ability).

3.3 The Metamodel

According to the PRoDe approach, the result of the process requirements analysis is a metamodel where all the above mentioned requirements have been translated into a set of metamodel elements, each other related. In this section we show (see Figure 3) only a portion of the obtained result, the one concerning the conscious part of our robotic systems that is useful for engineering the perception loop described in subsection 2.2. The core of the metamodel is composed of the *Robot* and the *Environment* concepts; this latter represents the world the robot is located in and it interacts with, we also consider the inner robot's world as part of the environment, in fact environment is composed of three concepts: *Artifact*, *Agent* and *Stimulus.*
Artifact is the part of environment that does not offer any autonomous capability, it can be used as a resource or can simply be an inanimate part of the world, it can also be a *Device*, one of the physical robot's component. The *Agent* concept - that can be specialised in *Reactive Agent* or *Rational Agent* - represents a member of the society of agents taking care all the robot's functionalities or each autonomous entity the robot interacts with in order to pursue a *Goal.*
In this metamodel we consider the robot as the unique agent endowed with autonomous reasoning capabilities about the self and the environment. The robot has one or more goals

Table 1: The Methodology Requirements, the Strategy and the Resulting Metamodel Elements

Methodology Requirement	**Strategy**	**Metamodel Element**
Developing complex conscious robotic systems able to realize the robotic perception loop.	Considering the robot as the main element of the developing system and composed of a society of agents some devoted to purely reactive tasks, some others to cognitive tasks.	Robot, Reactive Agent, Rational Agent, Device.
	The robot is the unique conscious agent endowed with self-reflective ability.	One Conscious Agent in each Robot
	The robot is part of the environment it lives in and owns a model of it.	Environment, Agent, Artifact
System perceives the environment (outer and inner one).	Supporting for the creation of a model of the "self" and of the environment.	Knowledge.
	Recognizing different kind of stimuli.	Stimulus, Proprioceptive, Sensorial.
Interacting with an unstructured environment.	The robot can plan the set of operation to be done for pursuing a specific goal.	Goal, Plan, Action.
The robot detects the differences between required and observed behavior.	Use the metaphor of testing	Test, Simulated Act, Log.
	Supporting the creation of a simulated environment (anticipation) in order to compare it with the perceived situations.	Simulated Act.

to pursue, they are achievable by means of *Plans* in turns composed by a set of actions.

The *Knowledge* the robot has about the environment and about itself is affected by each action performed during the system lifetime. An *Action* is a kind of act - a physical robot act or a communicative act between the robot and an external agent - resulting in an environment change of state; each changing state is traced in the knowledge by means of the concept instance values. As in all common ontologies, world status can be enquired by means of predicates related to concepts and asserting information about their status.

The robot has conscious capabilities in the sense that it can reason about the required and observed behavior by means of a *Test* composed by a set of *Simulated Acts* and a *Log*; each time a robot as to reach a goal, it establishes a plan on the basis of the knowledge it has about the environment, the goal itself and the set of stimuli it receives. Once the plan is defined, an expectation about the plan results can be calculated by simulating the plan actions and then estimating their results. When the plan simulation has been completed, the robot may compare the results of the simulated acts with the ongoing situation as it is reconstructed by using propioceptive and sensorial stimuli. From this comparison a test log can be compiled and the set of correlations between the observed and required behaviour identified.

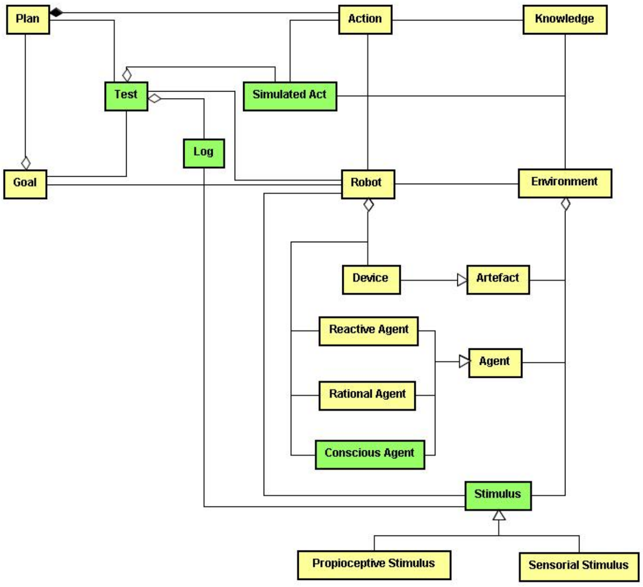

Figure 3: The Conscious System Metamodel

The log is then used for activating all the actions required to correct the plan, tune action parameters and successfully achieve the goal. The elements used for modeling robot conscious perception come from the fourth requirement (subsection 3.1); in the following subsection it will shown how, using the metaphor of software testing, we were able to create a set of design process activities that we integrated with PASSI in order to make the first step towards a design process for robot conscious system.

4. DESIGNING THE PERCEIVED SITUATIONS

The previous sections have shown the results of the application of PRoDe to the proposed case; the aforementioned process naturally conveys the idea of using test activities directly performed by the robotic system; in other words the approach can be summarized as it follows: the robot defines a kind of *preliminary* plan for achieving its goal; this preliminary plan is a portion of program to be executed. It is not sure yet that the plan will be successful since experience teaches that there are relevant differences in robotics between theoretical and practical results.

The preliminary plan is therefore executed in a simulated way. If it reaches its objectives during simulation this is supposed to success in reality too but for the sake of affordability it still has to be considered *not affordable*. Nonetheless the robot starts executing it in the real world. This choice seems reasonable because at the best of its knowledge the robot cannot produce a better plan and only feedbacks coming from its real execution can prove if it was correct. While the plan is executed, intermediate situations are compared with similar steps obtained in the simulated plan. If they match the robot increases its confidence in the plan (and the related set of choices thus including execution parameters like speed, position estimation and so on). If intermediate situations differ from results obtained by the simulated plan, then a corrective action has to be undertaken. In the case of a real software testing operation this should include modifications in the program in order to get it tackle the expected behavior and achieve the goal. In our case, corrective actions may include tuning the parameters used to execute the plan (for instance, speed differences can influence navigation precision) or aborting the preliminary plan and replacing it with a new (again preliminary) one. The final plan actually is the totally executed plan that proved successful. After its execution and continuous tuning obtained by means of the perception-loop implemented as a testing strategy, if the plan is well performed, it becomes part of the robot knowledge. It now knows that in the specific situation the final plan is a winning strategy and will naturally choose it again. Of course again perception loop will be used to monitor plan execution and tune it. In order to conceive the portion of design process implementing to the above described process, we analyzed the test activities suggested by the Unified Process (UP [20]) that is becoming the de-facto standard design process in the area of object oriented methodologies. Our aim was to identify and extract useful process fragments and to modify them according to differences/similarities with the activities to be made by the system.

We considered the UP activities related to *test plan and design* and *test execute*: with the first activity we can identify the system functionalities to be tested, the resources and the scope of test whereas the second activity aims at executing the test in order to investigate the results and to identify the defects.

During the planning test activity a workproduct, named *test case*, is produced; it is composed of a name, a location, a set of inputs, an oracle - the expected outcome - and a log - the correlation among expected and observed behavior.

The execution activity produces a report where the results of the test and the differences among observed and expected behavior are listed.

According to the design approach we are going to define, the designer starting from system requirements and goals, the environment description, the system architecture, all the robotic components and how they are interconnected, has to create a test plan resulting in the name, the location of the robotic module to be tested, a set of inputs deduced from the above mentioned starting considerations and finally a definition of the expected system behavior by means of simulation. For our purposes the UP test plan and design portion of process may be reused by only modifying the resulting test case; this because we need it to be composed of behavioral specification and not to contain the classic testing log. Instead, the log has to be designed in the test report, resulting from test execution activity, where a comparison between expected/simulated behavior and the real one is done.

In the following the new test fragments we extracted from UP and adapted to our needs are illustrated; we follow, for the first one, a specific template for fragment description (Test Planning and Design) whereas, for space concerns, we only give a general description of the Test Execution fragment.

4.1 Planning Test

The Test Planning and Design fragment will be described according to a template composed of the following sections:

- Introduction: describing the process the fragment was extracted from
- Fragment Description: describing the fragment in terms of its goal, workflow, activities and involved stakeholder roles

Introduction. This fragment has been extracted from UP Testing discipline and modified in order to meet the needs for designing a part of autonomous system devoted to create an expectation about what is the state of the environment after the execution of a specific plan on the base of the perception loop described in [5].

Fragment Goal. The aim of the fragment is to design the test case in order to make the system able to produce the expected behavior to the execution of a plan; each plan is used in the system for reaching a specific goal.

Fragment Descritpion. The following set of documents is the input of this fragment: *i)* the requirement document, *ii)* the environment description that the analyst takes into account for establishing which system requirement and which system goal have to be tested, *iii)* a model of the system architecture used for identifying which is the system component to be tested under the execution of a perception loop.

Portion of Process workflow. The process that is to be performed in order to obtain the result is represented in the following SPEM diagram (Figure 4)

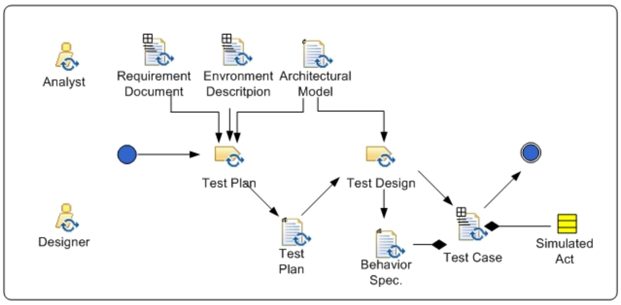

Figure 4: Test Planning and Design Activities

Activities reported in Figure 4 are detailed in terms of work to be done and involved stakeholders (see Table 2).

The above listed stakeholder roles within this fragment are responsible for the accomplishing of the following duties.

Responsibilities of *Analyst* are:

- *Analyzing Requirement Document, Architectural model and Environment description.*

- *Producing the Test Plan.*

Responsibilities of *Designer* are:

- *Analyzing Architectural in order to establish which part of the system has to be tested.*
- *Assisting the analyst in the identification of the expected system behaviour.*
- *Detailing the test case.*

Table 2: Activities Description

Activity	Activity Description	Roles Involved
Test Plan.	The analysts establishes the model component to be tested on the base of the system requirements and goal in order to sketch a plan of the test.	Analyst.
Test Design.	The designer and the analyst specify the expected behaviour of the system also producing the test case.	Designer, Analyst.

Fragment Metamodel. This fragment refers to the MAS meta-model adopted in the extended PASSI design process and contributes to define the *Simulated Act* element that is an essential part of the *Test* MMME.

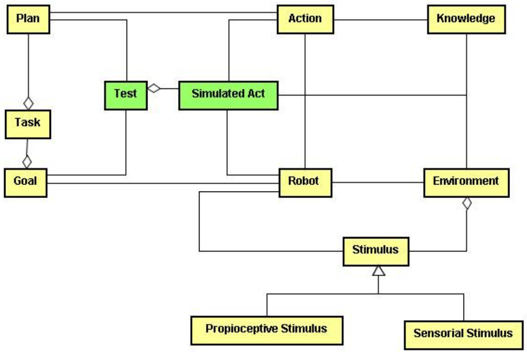

Figure 5: The Portion of Metamodel

Preconditions and concepts to be defined. Input, output and elements to be designed in the fragment are detailed in the following tables, documents and the MAS metamodel elements see Tables 3 and 4.

Guideline. The analyst may start from information reported in the requirements document and identify a set of scenarios that are candidate to be tested. These scenarios (each pursuing a specific goal) should be associated with a set of robot's internal modules whose perception loops can provide useful feedbacks for testing the preliminary plan the robot generates to cope with that. For each scenario and related robot's module, the set of input ensuring the adherence to the prescribed flow of events has to be defined.

Table 3: Input/Output

Input	Output
Requirements Document, Environment Description, Architectural Model.	Test Plan, Beavioural Specification, Test Case.

Table 4: MAS metamodel elements

Input	To Be Designed	To Be Related	To Be Quoted
Plan, Goal, Requirement, Knowledge, Agent.	Simulated Act.	Knowlegde, Agent	Plan, Goal, Requirement.

Composition Guideline. None specific beyond the satisfaction of input constraints.

Aspects of fragment. The fragment has been conceived to support a specific robotic architecture and planning approach: the robot is composed of several agents and uses a perception loop in order to estimate the goodness of a preliminary plan. This plan is accepted and its execution started once the simulation provides positive results. During execution feedbacks for plan trimming are deduced from the comparison of real world results with the expectation generated by the simulation step.

Dependency Relationships with other fragments. This fragment has been conceived for adoption in a process work flow where the next fragment is the "Executing Test".

4.2 Executing Test

The aim of this fragment is to design the portion of robotic system devoted to produce the *log* metamodel element, this is the element resulting from the comparison between expected and observed system behaviour. The test case generated from the "Planning Test" fragment is the input document.

The fragment is composed of two activities: *Test Execution* and *Test Evaluation*, both performed by the Designer. During Test Execution, starting from all the simulated acts detailed in the test case, a document describing the test results is produced and then used by the Test Evaluation activity for producing a Test Report describing the *Log* element of the metamodel. In Figure 6 the portion of metamodel affected by this fragment can be seen.

Figure 7 shows the process that is to be performed in order

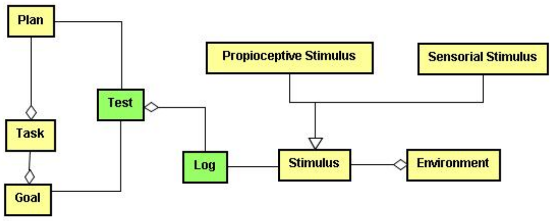

Figure 6: The Portion of Metamodel for Test Execution

to obtain the result as a SPEM diagram.
The fragment delivers two outputs: *Test Result* that is a structured text document [26] and *Test Report* that is a free text document.

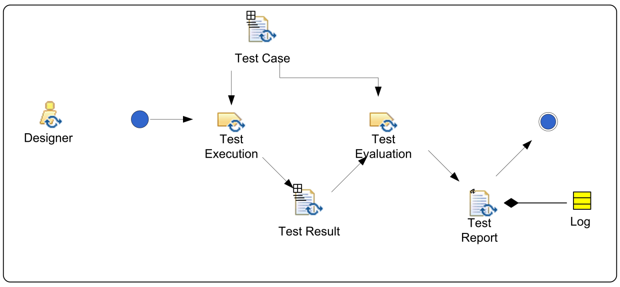

Figure 7: Test Execution and Evaluation Activities

5. CONCLUSIONS

In this paper we presented an experiment based on PRoDe (a Situational Method Engineering approach) for modelling, designing and implementing complex robotic systems.
By following this approach and basing on the results it provided we decided to extend PASSI, the design process we used in the latest years for several robotic experiments and for developing a cognitive robotic architecture.
The proposed work aims at creating a design process (or better to extend PASSI with a new portion of design process) for designing the portion of a robotic system with the following features: *(i)* generating expectations about the state of the world (therefore also about itself) as results of a plan execution, starting from the goals and the knowledge on the environment, *(ii)* comparing this “simulated state” with the effective reality after plan execution.
For these purposes two fragments have been extracted from UP and modified for meeting the new process requirements. In this paper we illustrated only the first iteration on the PRoDe process, the two fragments have been integrated with PASSI and we are going to test the whole design process in order to start a further iteration of the process if necessary.
This work also proposes a first model of *conscious systems* that has not been still developed and recognized by researchers in the field cause the lack of a precise and standard definition of consciousness.

6. ACKNOWLEDGEMENTS

Part of this work makes use of the results produced by the EU project FP7-Humanobs.

7. REFERENCES

[1] I. Aleksander. Capturing consciousness in neural systems. In editors. Aleksander I, Taylor JG, editor, *Proceedings of the 1992 international conference on artificial neural networks: (ICANN-92).*, volume 2, pages 11–15. Amsterdam: North-Holland Publishing Company, 1992.

[2] Agile Alliance. *http://www.agilealliance.org.*

[3] S. Brinkkemper, R.J. Welke, and K. Lyytinen. *Method Engineering: Principles of Method Construction and Tool Support.* Springer, 1996.

[4] A. Chella. Towards robot conscious perception. In A. Chella and R. Manzotti, editors, *Artificial Consciousness*, pages 124 – 140. Imprint Academic, Exeter, UK, 2007.

[5] A. Chella. A robot architecture based on higher perception loop. 2009.

[6] A. Chella, M. Cossentino, L. Sabatucci, and V. Seidita. From passi to agile passi: tailoring a design process to meet new needs. In *IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT-04)*, Beijing, China, Sept 2004.

[7] A. Chella, M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. *Artificial Intelligence*, 89(1-2):73–111, 1997.

[8] A. Chella and I. Macaluso. The perception loop in cicerobot, a museum guide robot. *Neurocomputing.*, 72:760–766, 2009.

[9] A. Chella and R. Manzotti. *Artificial Consciousness.* Imprinting Academic, Exter, UK, 2007.

[10] M. Cossentino. From requirements to code with the PASSI methodology. In *Agent Oriented Methodologies*, chapter IV, pages 79–106. Idea Group Publishing, Hershey, PA, USA, June 2005.

[11] M. Cossentino, S. Gaglio, A. Garro, and V. Seidita. Method fragments for agent design methodologies: from standardisation to research. *International Journal of Agent-Oriented Software Engineering (IJAOSE)*, 1(1):91–121, 2007.

[12] M. Cossentino, L. Sabatucci, and A. Chella. Patterns reuse in the PASSI methodology. In *ESAW*, pages 294–310, 2003.

[13] M. Cossentino and V. Seidita. Composition of a New Process to Meet Agile Needs Using Method Engineering. *Software Engineering for Large Multi-Agent Systems*, 3:36–51, 2004.

[14] R. Grush. The emulator theory of representation: motor control, imagery and perception. *Behavioral and Brain Sciences*, 27:377 – 442, 2004.

[15] P.O. Haikonen. *The Cognitive Approach to Conscious Machines.* Imprint Academic, Exeter, UK, 2003.

[16] P.O. Haikonen. *Robot Brains.* John Wiley & Sons, Chichester, UK, 2007.

[17] G. Hesslow. Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6(6):242–247, 2002.

[18] O. Holland and R. Goodman. Robots with internal models - a route to machine consciousness? *Journal of Consciousness Studies*, 10(4 -5):77 – 109, 2003.

[19] O. Holland, R. Knight, and R. Newcombe. A robot-based approach to machine consciousness. In

A. Chella and R. Manzotti, editors, *Artificial Consciousness*, pages 156 – 173. Imprint Academic, Exeter, UK, 2007.

[20] I. Jacobson, G. Booch, and J. Rumbaugh. *The unified software development process.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1999.

[21] B. Kuipers. Consciousness: Drinking from the firehose of experience. In *Proceedings of the Twentieh National Conference on Artificial Intelligence*, pages 1298 – 1305, Menlo Park, CA, 2005. AAAI Press.

[22] K. Kumar and R.J. Welke. Methodology engineering: a proposal for situation-specific methodology construction. *Challenges and Strategies for Research in Systems Development*, pages 257–269, 1992.

[23] I. Macaluso, E. Ardizzone, A. Chella, M. Cossentino, A. Gentile, R. Gradino, I. Infantino, M. Liotta, R. Rizzo, and G. Scardino. Experiences with CiceRobot, a museum guide cognitive robot. In S. Bandini and S. Manzoni, editors, *AI*IA 2005*, volume 3673 of *Lecture Notes in Artificial Intelligence*, pages 474 – 482, Berlin Heidelberg, 2005. Springer-Verlag.

[24] I. Macaluso, E. Ardizzone, A. Chella, M. Cossentino, R. Gradino, M. Infantino, I.and Liotta, R. Rizzo, and G. Scardino. Experiences with cicerobot, a museum guide cognitive robot. In *S.Bandini, S. Manzoni (eds.) AI*IA 2005, Lecture Notes in Artificial Intelligence.*, volume 3673, pages 474–482. Springer-Verlag, Berlin Eiderlberger.

[25] A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, December 2008. Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems.

[26] V. Seidita, M. Cossentino, and s. Gaglio. Using and Extending the SPEM Specifications to Represent Agent Oriented Methodologies. *Agent-Oriented Software Engineering IX*, pages 46–59, 2009.

[27] V. Seidita, M. Cossentino, V. Hilaire, N. Gaud, S. Galland, A. Koukam, and S. Gaglio. The metamodel: a starting point for design processes construction. *International Journal of Software Engineering and Knowledge Engineering. (in printing).*, 2009.