Engineering Applications of Artificial Intelligence 55 (2016) 14–25

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Engineering multi-agent systems using feedback loops and holarchies

Gillian Basso [a], Massimo Cossentino [b], Vincent Hilaire [c,*], Fabrice Lauri [c], Sebastian Rodriguez [d], Valeria Seidita [e]

[a] Institute of Systems Engineering, University of Applied Sciences and Arts Western, Switzerland
[b] ICAR – National Research Council, Italy
[c] Univ. Bourgogne Franche-Comté, UTBM, IRTES EA7274, F-90010 Belfort, France
[d] GITIA, UTN-FRT, Rivadavia 1050, San Miguel de Tucuman, Argentina
[e] Dip. Ingegneria Chimica, Gestionale, Informatica, Meccanica – University of Palermo, Italy

ARTICLE INFO

Article history:
Received 6 October 2015
Received in revised form
18 May 2016
Accepted 23 May 2016

Keywords:
Feedback loops
Methodology
Multi-agent systems
Holarchies

ABSTRACT

This paper presents a methodological approach for the engineering of Multi-Agent Systems using feedback loops as a first class concept in order to identify organizations. Feedback loops are a way for modeling complex systems that expose emergent behavior by means of a cause-effect loop between two levels called micro and macro levels of the system. The proposed approach principles consist in defining an abstract feedback loop pattern and providing activities and guidelines in order to identify and refine possible candidates for feedback loops during the analysis phase of the Aspecs methodology. This approach is illustrated by using an example drawn from the smart grid field.

1. Introduction

Since the beginning of the Distributed Artificial Intelligence field, many Multi-Agent System (MAS) approaches have taken inspiration in metaphors coming from other domains, such as, for example, control theory and biology. Among these metaphors, one can refer to ant colony systems, artificial immune system, swarming systems and feedback loops.

The aim of feedback loops is, in this case, to allow to define self-organizing systems. The principle consists of a feedback effect on its own cause. The sequences of causes and effects define a loop (called feedback loop). Feedback loops have already been used in the context of MAS, for example, for applications such as manufacturing, collective robotics, simulation and information systems design (Brückener, 2000; Wolf and Holvoet, 2007; Caprarescu et al., 2009; Beurier et al., 2003; Schmickl et al., 2011).

However, although there are many success stories, systems based on feedback loops remain difficult to analyze and design for any type of application. Indeed, despite the numerous and important works in this specific field, there are no principled approaches that define feedback loops as first-class entities and take them into account in a strictly defined methodological framework (Brun et al., 2009). By first class entities, we mean an object that can be used and manipulated (intrinsic identity) in the analysis, design and implementation models.

In this paper, we propose a set of abstractions and a methodological framework to address this gap. For this, we refer to and change the Aspecs methodology (Cossentino et al., 2010, 2013a) which is dedicated to the analysis, design and implementation of complex systems by MAS. The Aspecs methodology adopts an organizational approach, sets of functionalities are assigned to organizations that accomplish them also by means of the hierarchical decomposition in sub-organizations (holonic paradigm). The Aspecs methodology is among the most comprehensive MAS methodologies (Isern et al., 2011). Indeed, assuming an organizational framework, Aspecs provides the concepts necessary to decompose a complex system. The realization of these systems is then simplified by an implementation and deployment platform Janus (Gaud et al., 2009), which facilitates the implementation of the concepts of the methodology.[1]

The abstractions and methodological guidelines we propose in this paper are used to analyze and identify feedback loops and

* Corresponding author.
E-mail address: vincent.hilaire@utbm.fr (V. Hilaire).

[1] The interested reader can see the ASPECS website: aspecs.org for more informations, examples and references.

http://dx.doi.org/10.1016/j.engappai.2016.05.009

organization in self-organizing complex systems; then, we use them to develop a simulator of smart grids (Basso et al., 2011) used as illustration in this paper.

The general principle of the proposed approach is twofold. The first principle is based on the description of the combinatory automaton from Mella (2008). This description acts as a conceptualization of feedback loops and is expressed as a kind of feedback loop pattern. The second principle uses an ontological analysis of the problem domain of interest. The aim of the ontological analysis is to find the elements candidate for instantiating the feedback loop pattern. Ontological analysis has already proven its interest for the analysis and design of MAS based on self-organizing systems like, for instance swarm systems (Hilaire et al., 2016).

The elements resulting from the ontological analysis can be subsequently refined and realized within organizational structures underlying the MAS to-be and internal mechanisms of agents.

This ontological analysis is integrated in the Aspecs methodology as one of its initial activities. The ontology resulting from this activity is exploited to define the roles and organizations that will be embodied by the agents.

This paper is organized as follows: some related works are presented in Section 2, Section 3 presents the Aspecs methodology and combinatory automaton concepts. Section 4 details the approach principles. Section 5 illustrates the approach with an example. This example is a smart grids software simulator (Basso et al., 2011) that results from the presented approach, eventually Section 6 concludes.

2. Related works

Coordination, regulation and control of agents are subjects of interest for the MAS community as shown by recent works (Heras et al., 2014; Mariani and Omicini, 2015). The main contribution of this paper concerns the engineering of systems using feedback loops as a regulation and control mechanism. As such, existing works on feedback loops are obviously of interest but we will not limit this section to only those works. Indeed, underlying the concept of feedback loops there are other related fields such as self-organization, emergence and multi-level modeling and control.

The concept of feedback loops is already existing in many disciplines such as biology and has been used as an engineering principle in control theory for several decades (Ogata, 1997). In computer science it has been used as a regulation mechanism that allows the definition and maintenance of a self-organized system. The general idea is to produce the emergence of a global phenomenon from micro-interactions and to control these micro-interactions from the global phenomenon level in order to maintain a satisfying state. In this context, Parunak (1997) identified the need of a (positive) feedback in order from a MAS to self-organize instead of simply producing disorder. Brun et al. (2009) present a survey of existing feedback loops in several domains related to computer science. Among the identified open issues, two are of importance for the contribution presented in this paper: the explicit modeling of feedback loops and the methodological support for system architecture and design. A possible way of answering these two open issues is to reify feedback loops as proposed in this paper.

A generic but simple model of feedback loops can be found in Dobson et al. (2006). This model is too simple to act as a reification of feedback loops and is more intended to reason about feedback loops. Wolf and Holvoet (2007) contribute to the engineering of intentional feedback loops for system control but the focus is on information flows within feedback loops.

Concerning multi-level modeling and behavior/organization emergence, there also exist some previous works. For example, Beurier et al. (2012) propose a study of multi-level emergence and contribute to the definition of a modeling approach in this context. However, the authors do not propose any metamodel for concepts reification and no methodology.

Some methodologies try to deal with multi-level modeling. Aspecs integrates holonic related concepts that allow a kind of multi-level modeling of the system. However, before this paper, Aspecs provides no support for feedback loops. In the SODA methodology (Cossentino et al., 2013b), the authors propose the layering principle in order to deal with multi-level modeling. The layering principle consists in the abstraction of the system-to-be in order to manage complexity. Layers can thus be distinguished and studied separately. This concept is useful but it does not suppose the existence of several simultaneous and distinct levels with different timescales or granularity. Moreover, the layer concept is not present as a first class entity in the SODA metamodel.

Concerning the holonic field, there are numerous approaches. One can cite Brussel et al. (1998), Giret and Botti (2009), Barata (2006), and Leitão and Restivo (2006). The Anemona methodology (Giret and Botti, 2009) relies on a problem decomposition approach based on a "divide and conquer" principles. The PROSA model (Brussel et al., 1998), Cobasa (Barata, 2006) and ADACOR approaches (Leitão and Restivo, 2006) are specific to Holonic Manufacturing Systems and as such make hypothesis specific to manufacturing systems. These approaches do not integrate the use of feedback loops as a first class engineering concept.

3. Background

3.1. The Aspecs methodology

Aspecs (Cossentino et al., 2010, 2013a) is an agent oriented design methodology for the analysis and design of hierarchic multi-agent solutions starting from the requirements analysis to code production and deployment of the system on a specific platform. The main principles underlying Aspecs are described by metamodels defining organizational concepts such as Organizations, Roles, Interactions and Capacities. In this context, an Agent plays roles within organizations. A notion of compound agents is modeled by the concept of *Holon*.

In the Aspecs methodology and in this paper we use the term Holon as it was defined by Koestler (1967): *the term Holon comes from the Greek* '*holos*' *meaning* '*whole*', *and the suffix* '*on*' *meaning* '*part*' *or entity* (*for instance as a proton or neutron is a part of an atom*); *hence a holon is a whole to those parts beneath it in the hierarchy but at the same time a part to those wholes above it.* A holon is an element which can be seen as both a component part of an upper level, and as a compound of any other (lower level) holons. Therefore, the notion of holon is inherently recursive and can naturally describe hierarchical systems. This concept has been adopted by the community of distributed artificial intelligence as holonic multi-agent systems (Gerber et al., 1999) (designated as HMAS from now on).

The most relevant phases of the Aspecs methodology are:

- the requirements analysis phase, providing a description of the problem domain from an organizational point of view. It formalizes the available knowledge about the problem domain within an ontology.
- The agent societies conception phase that should provide a solution to the problem described in the previous phase in terms of agents/holons,
- The implementation phase that describes the architecture of the

MAS and provides the source code of the application.

An extension of the UML modeling language is adopted, and a new UML profile has been specially introduced.

The first activities of the process are the most relevant for the work described in this paper. Indeed, these activities prescribe the following work to be done: initially, the objectives of the application are identified and described in terms of use cases (first Aspecs activity, Domain Requirements Description), then all the available knowledge about the problem and its context is conceptualized in a Problem Ontology Description.

This ontology must provide a first definition of the application context and of the domain-specific vocabulary. It aims to deepen the understanding of the problem, and to complete the requirements analysis with the introduction of the concepts that make up the problem domain and their relations. The Ontology resulting from this activity plays a crucial role in the Aspecs development process. Indeed, its structure will be critical for the identification of organizations. Ontology is described in terms of concepts, actions and predicates. It is represented with a specific UML profile for class diagrams. This profile is based on a specification of the FIPA[2] organization (FIPA, 2001).

The stereotypes defined by this profile are Concept, Action, and Predicate.

Each *Concept* refers to an entity in the domain of interest during the analysis phase. These entities can represent resources, actors, manipulated objects, etc.

An *Action* represents a treatment or processing in the field of interest. This treatment or this transformation can also be specified by attributes and operations. Classes representing actions can possibly be bound by an association to a *Concept* performing this action. This association is named *Actor*. A second association, also optional, defines a link between an action and a set of *Concepts* used as parameters for this action. This association is named *Argument*.

Eventually, *Predicate* is used to represent domain knowledge in the form of a property. This property is expressed by a first-order logic predicate. A predicate can be combined with a set of *concepts* which are variables of the predicate.

The Problem Ontology and the requirements allow to identify Organizations and to produce a first view on the organizational hierarchy that may be further detailed and extended in order to obtain the global organization exposing the required system behavior. In order to realize such behavior, it is important to identify a set of interacting roles. From the definition adopted in Cossentino et al. (2010) a Role is an expected behavior (a set of role tasks ordered by a plan) and a set of rights and obligations in the organizational context. The goal of each Role is to contribute to the fulfillment of (a part of) the requirements of the organization within which it is defined.

Capacities allow Roles to pursue the objectives of the organization they belong to. In other words, a capacity is an abstract description of the know-how of the role. An organization is represented graphically by a stereotyped class diagram. Organizations are represented by packages containing classes stereotyped role and interaction between roles. Capacities are represented by stereotyped classes outside the organization package and linked to the relevant roles by specific relationships.

Concerning the Agent Society Design phase, the main work-product is the definition of the MAS architecture or holarchy (if holons are chosen instead of agents). This holarchy takes the form of a new kind of diagram that is an extension to the cheeseboard diagram (Ferber et al., 2003). In this diagram, groups of agents, that are instances of organizations are represented as ovals with agents inside appearing as skittles standing on the board and sometimes going through the board if they belong to several groups. Each agent is tagged with the different roles it is playing. This diagram is interesting for HMAS as the different levels of the holarchy can be visualized. For each level, a special group is added for holarchy management aspects. This group is composed of predefined roles (Peer, Representative, Head, etc.) that handle the aspects linked to holarchy dynamics. Further details about Aspecs are not relevant to this paper and will be omitted. The interested reader can find them in the cited papers.

3.2. Combinatory automaton

The principles we have adopted to define primitive concepts for feedback loops' analysis and design are inspired by the work of Mella (2000). In this work, the author models complex systems as automata called combinatory automata.

In these automata (see Fig. 1), the author distinguishes two levels: the micro and macro levels. At the micro level, there is a set of individuals or entities, called agents, exhibiting behaviors. These agents are denoted by a_i where i is the agent identifier ranging from 1 to N. Each agent also has a set of relevant features that describes it. These features are represented by the p_i set. The analytical state of the system $\Lambda(t_h)$ for a given time t_h (micro-level) is thus characterized by the set of couple (agent, agents' features). Each agent a_i micro-effect is denoted e_i. The relation between e_i and a_i is defined by a time dependant function g_i. The behaviors of agents are qualified of micro-behavior and is defined by the whole set of micro-effects for a given time interval T. The micro-behavior is denoted $A(T)$. These micro effects, once combined through combination operation $C_{1\le i\le N}$, produce a synthetic state, denoted $X(\Lambda, t_h)$, at the macro level, macro effects. The combination operation may take several forms, for example, a sum, a product, average, min, max, etc. The macro effect, observable result of the automaton behavior is defined as $F(X(\Lambda, t_h))$ and denoted $E(\Lambda, t_h)$. The synthetic state $X(\Lambda, t_h)$ determines the condition that directs the subsequent micro behaviors and forces single agents to follow the emergent behavior of the collectivity. This mechanism is represented by the Necessitating factor denoted by $N_{1\le i\le N}$. This loop from micro to macro and reverse contributes to the production of a self-organization mechanism.

The initialization of this kind of phenomenon is either due to chance or a specific event that triggers the micro behaviors. These principles are illustrated in Fig. 1.

In order to illustrate the combinatory automaton concept an example is presented. This example tries to model the urban settlement phenomenon.

Typically, people needing a house (necessitating factor) search for places that satisfy some conditions. These conditions may be the presence of water, roads, hills, beautiful view, services, etc. If these conditions are met, then a dwelling (micro effect) is built (micro behavior). Again, there are two possible futures. First, the dwelling may be abandoned after a while. Second, the chosen location may be chosen by other persons as favorable (recombining factor). In this case, the settlement will grow (macro effect).

The rules for Urban Settlement from Mella (2000) are thus:

MICRO RULE = NECESSITATING FACTOR – if you need to build a house, look for favorable conditions; if there is a city there already, we assume favorable conditions exist; leave your house to your descendants;

MACRO RULE = RECOMBINING FACTOR – the construction of new houses strengthens the urban settlement; the strengthening and growth of the city are signs that favorable conditions exist (opportunities, services, protection, etc.), and this influences the

[2] Foundation for Intelligent Physical Agents.

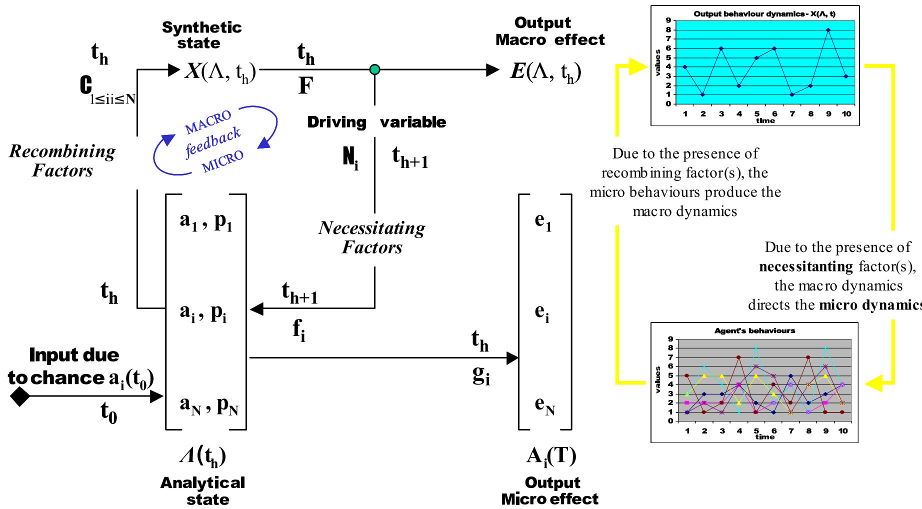

Fig. 1. Combinatory automaton extracted from Mella (2000).

micro behaviors. The older and larger the city is, the greater the incentive for new arrivals to the area to locate there;

MICRO–MACRO FEEDBACK. CHANCE AND NECESSITY – the city is the result of individual decisions to build a house in a favorable place; however, the presence of a city gives information that favorable living conditions have been found, and this influences the individual micro behaviors (the city itself indicates the favorable conditions). A city arises “by chance” but, once begun, the phenomenon is “by necessity” maintained over time as long as the necessitating factor operates;

STRENGTHENING, WEAKENING AND CONTROL FACTORS – overcrowding increases in the time needed to cross a town, the desire for solitude, and the need to preserve the surrounding areas: these all represent weakening factors, and where these prevail over the need for a community life the “urban settlement” combinatory system will not be formed. Instead, a different system would be activated that we can call “maintain-the-territorial-division”. Examples of strengthening factors are the danger of invasion, tourist attractions, tax incentives for construction, and the supply of attractive urban services. The macro control can act by means of urban planning; the micro control can influence the desire to live in a town, or the opposing desire to flee the crowds.

Fig. 2 describes the simplest model by Piero Mella's work for illustrating and describing the activities of Combinatory Systems. A complex system (or a Combinatory System) may be seen under two different points of view, the system as a whole (a collectivity) and as the set of agents it is composed of that may expose behaviors. Furthermore, the behavior of the complex system may be seen from two points of view, the macro and the micro one. The macro behavior derives from the combination of micro-behaviors that produce micro effects and the recombining factors present in

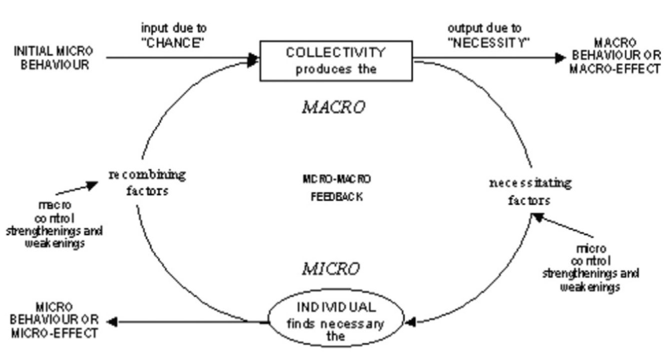

Fig. 2. The invisible hand by Mella (2000).

the environment; the system produces the macro behavior (or macro effect) by recombining the micro effects on the basis of a set of recombining factors derived from the environment (for instance: rules, constraints, algorithms, etc.). Once recombined, the micro effects produce the macro behaviors and their macro effects that, influenced by the necessitating factors such as obligation, imitation and so on, force agents to adapt their micro behavior. The complete self-organizing behavior of the complex system has to be seen as the result of the continuous feedback between micro and macro level.

The aim of our work is providing a means for identifying FLs during the Analysis phase of Aspecs and then using them for identifying organizations in a new way. For doing this, it is necessary to have a representation of feedback loop using terms and concepts of the Aspecs ontology, hence *concept*, *predicate* and *action*. First of all, we may notice that the following elements have to be represented: agents, collectivity, micro and macro effects, micro and macro behavior and recombining and necessitating factors.

In the following sections we illustrate where and how we extended Aspecs in order to include the FL in the design with Aspecs of systems exposing emergent behavior.

4. Approach principles

4.1. Overview

The principles of the approach proposed in this paper consist in (i) identifying concepts that are candidates for feedback-loops, and (ii) providing the necessary methodological activities and guidelines that will refine them down to organizational structures that will be later deployed as an HMAS. The starting point for that is to consider the conceptual model of the problem resulting from the Problem Ontology Description.

When analyzing a system, and particularly in the description of the domain ontology corresponding to the result of the POD activity, a number of items may be identified as candidates for inclusion in the feedback loop. Thus, POD constitutes the way to fill the methodological gap for identifying feedback loops as a first class entity. First of all, in order to identify feedback loops (FLs) in the POD we need an abstract representation (or meta-representation) of it, here the rationale of FLs is represented through the same elements used in the POD.

4.2. FL abstract representation

Fig. 3 shows a feedback loop by means of concepts and actions; FL, firstly, requires the presence of at least two levels of description: a macro and a micro level. These two levels may be embodied in the domain ontology by a composition of concepts. The concept corresponding to everything belongs to the macro level and the concept compound belongs to the micro level. In each level, one must distinguish acting entities or actors. An actor (or set of actors) belongs to the macro level, ActorMacro in Fig. 3, and a set of actors belongs to the micro level, ActorMicro. Each of these types of actors can perform actions. Micro level actors (ActorMicro) exercise action triggered by the necessitating factors and produce a result set (one per player) which are represented by the concept of influence. This influence (or set of influences) is the parameter of the action of the macro level actor (ActorMacro). This action is triggered by recombining the result of ActorMicro action and some external factors. The result of recombining is represented by the reaction concept that is taken as an argument by the necessitating action.

Since ActorMacro and ActorMicro expose what Mella calls macro behaviors and micro behaviors, in our abstract representation, they are the actors of two *actions*, we called them macroBehavior and microBehavior. These two actions, respectively, have two resulting concepts: the macroEffect and the microEffect.

In Mella's work microEffects are recombined with recombining factors in order to produce the collectivity's macro behavior. We model this situation by using a *predicate* (Recombining) that has both microEffects and the RecombiningFactors as arguments and is at the same time argument for macroBehavior action. The same has been made for the Necessitating predicate whose aim is to model the other arch of the loop. It has two arguments: Necessitating Factors and macroEffect, and is an argument for microBehavior action. Necessitating and recombining predicates are used to model the “invisible hand” that guides the behavior of the single entities and of the whole; from the work of Mella it arises that combinatory system behavior is guided by some kind of reasoning entity that, on the base of the result of its reasoning, triggers the actions of the micro actor or the macro actor.

A first step towards the identification of all the elements to represent in what we call a feedback loop abstract representation (see Fig. 3) was to represent the actors of combinatory system. Actors, and specifically the concepts ActorMacro and ActorMicro, respectively, model collectivity and single agents, hence entities in the domain (*concepts*). Moreover, referring to the principles given by Mella, we may represent both ActorMacro and ActorMicro as belonging to the same organization.

4.3. Conceptual model validation

In order to assess our theory, we applied the FL abstract representation to Mella's combinatory systems example used in the previous section: the Urban Settlement phenomenon. We realized that all the concepts he presents may be represented with our FL abstract representation.

The Urban Settlement example – Fig. 4 shows this system of accumulation example. Modeling the urban settlement domain implies to consider concepts such as a city that is a specialization of urban settlement, citizens aggregating society, favorable conditions that may be specialized in services opportunities and so on. The act of building a house is triggered by the “need for dwelling” predicate that has favorable condition and city as arguments, the meaning of these elements is: when the “need for dwelling” of the citizen may be accomplished by the fact that a city already exists and at the same time it offers favorable conditions (the predicate models the act of reasoning of the whole combinatory system) then the *Build* action is triggered. The same for the other predicate, if houses and strengthening factors are present, then they can be recombined and result in the city urbanization.

Our principal aim is to exploit Feedback loops and especially the FL abstract representation in a complete agent oriented

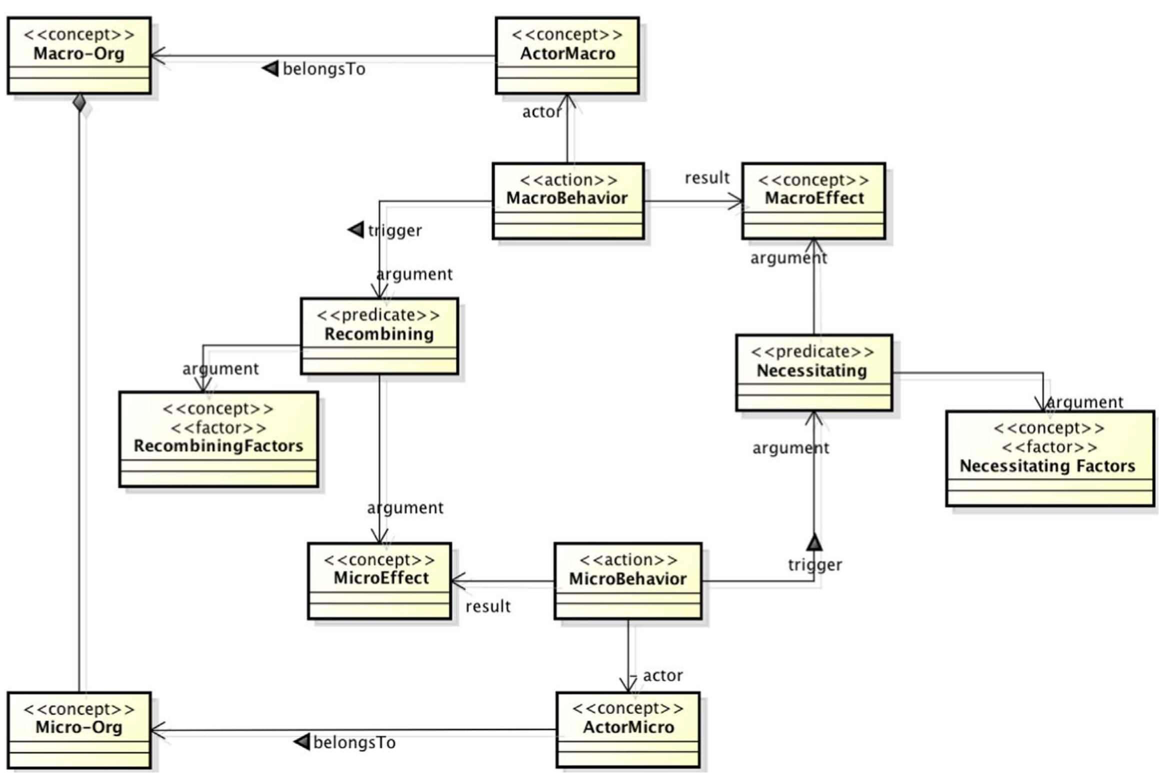

Fig. 3. Feedback loop abstract representation.

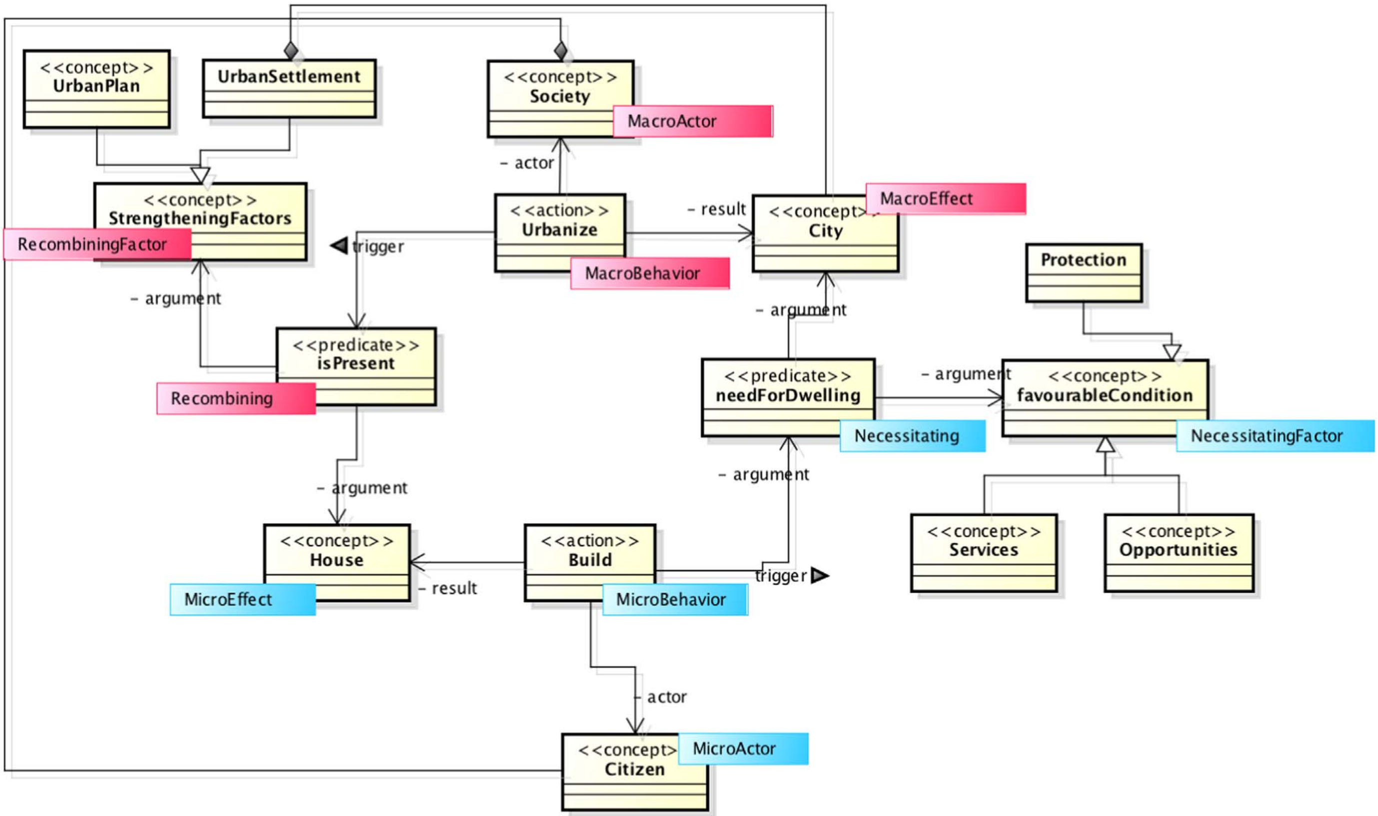

Fig. 4. The Urban settlement example.

methodological approach in order to identify organizations. By using Aspecs, the identification of organization goes through three different steps (from the Domain Req. Description by means of Problem Ontology to the Roles and Capacity identification) that result in the decomposition of behaviors into roles and capacities (see Section 3.1). The Problem Ontology description is the focal point for identifying organizations so we need a way of extracting and detailing roles, capacities and data starting from each feedback loop identified in the POD.

As it can be seen in Fig. 3, ActorMacro and ActorMicro perform actions, thus realizing expected behaviors, hence a Role as said in Section 3.1. For each entity performing actions in the POD we may discriminate two different kinds of roles, the one relating to process incoming information (performing a kind of sensing actions) and another one performing the actuating tasks hence modifying one or more properties of the entities populating the environment (see Fig. 5).

These two different Roles are, respectively, modeled as Rin and Rout.

Fig. 5 has to be intended as the pattern for the identification of organizations coming from feedback loops existing in the system. In the pattern, action has been stereotyped as capacity, indeed capacity represents that the know-how roles have to possess in order to exhibit a specific behavior. Moreover, the flow of information between roles has been reported taking into consideration both the direct and indirect communications, data

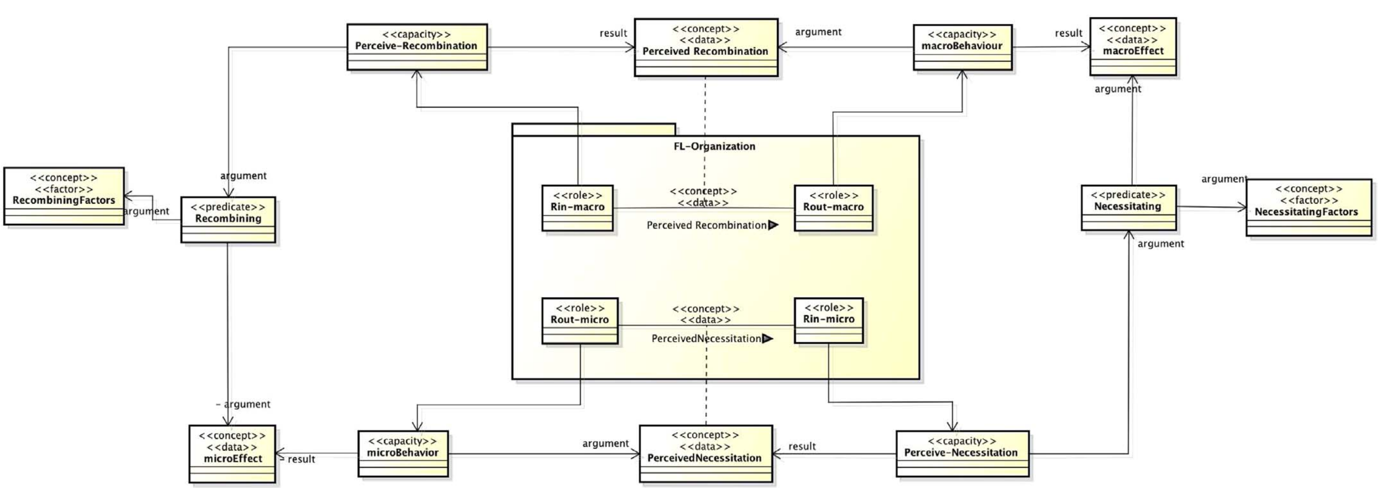

Fig. 5. Aspecs organizational pattern.

transferred in indirect communications are represented by association classes; the results of Perceive-Recombination and Perceive-Necessitation capacities are Perceived Recombination and Perceived Necessitation that are the data exchanged among Rout and Rin in a direct fashion or conveyed by the environment thus realizing indirect communications.

The collection of Roles participating both in the Micro and the Macro level constitutes the feedback loop organization (FL-Organization), indeed from the definition of organization in ASPECS: *An organization is defined by a collection of roles that take part in systematic institutionalized patterns of interactions with other roles in a common context. This context consists in shared knowledge and social rules/norms, social feelings, and is defined according to an ontology.*

In the following, we provide the description of the methodological activities and the guidelines conceived for going from the ontological representation of the problem domain to the identification of organizations. Moreover, in Section 5 we provide the results of an experiment that validates the use of the two different FL patterns for refining the POD in order to identify the organization. We used the POD coming from an existing project developed by using ASPECS; this POD is not influenced by the FL concepts and we can reasonably affirm that it is in a form that is very common, with the level of details that analysts generally reach when creating problem domain ontology from a textual description.

4.4. Feedback loop identification

In order to identify the FLs in this POD we should follow these guidelines:

Guideline 1. *Identify among all the concepts present in the POD which of them can be labeled as Actor; this can be done by considering that Actor is an entity able to perform some action. Thus, Actors may be identified by looking at concepts which make some kind of actions.*

Guideline 2. *After the Actors have been identified, they have to be divided into (candidate) ActorMacro and ActorMicro; this may be done following Fig. 3, each time a couple of concepts are Actors and are related by an aggregation relationship they may be labeled, respectively, as ActorMacro or ActorMicro. If you do not find any couple of actors respecting this relationship then you have to consider each single actor, look at the actions it performs and reason if it can be considered an ActorMacro or an ActorMicro of a possible FL. Hence, you have to analyze the POD and, if necessary, refine it. This is the first case of POD refining. Note that until this point the identified ActorMacro and ActorMicro are only candidate one, they can be confirmed after the following guidelines have been executed.*

Guideline 3. *For each (candidate) ActorMacro/ActorMicro search for the action it performs that may represent a candidate MacroBehavior (or MicroBehavior). This action should produce a concept as a result that (directly or through a predicate) is used as an argument by another action. This action becomes a candidate for being the MacroBehavior; a first structure of Fl has been identified.*

Guideline 4. *Refine the POD in order to insert missing elements of the identified FL.*

4.5. Identification of organizations

The Organization Identification activity (OID) as defined in the ASPECS process is based upon the domain ontology. It aims to identify organizations that will be the basis for meeting the objectives of the system. Fig. 5 illustrates the principles that lead from the results of the previous activity to candidate organizations. The idea is to associate an organization to the macro/micro levels corresponding to the candidates feedback loops. These organizations are not final and may be further decomposed or refined in subsequent activities. This skeleton organizational structure also integrates the links representing the necessitation/recombination relations linking each adjacent level. In this structure, the macro and the micro level are together represented by organizations.

There are no theoretical limits to the inclusion of cascaded feedback loops. The following guideline may help in identifying organizations:

Guideline 5. *A feedback loop involves defining an organizational structure composed by at least two levels in which each organization at level* $i+1$ *incorporates a (sub-) organization at level i.*

4.6. Identification of roles and interactions

Once organizations are defined, the analyst can populate them with interacting roles. The principle is that each organization includes two couples of roles, Rin-macro and Rout-macro corresponding to the macro level and Rin-micro and Rout-micro, corresponding to the micro level of the organization.

Guideline 6. *A feedback loop involves two roles at the macro level and another two at the micro level. Substitute ActorMicro and ActorMacro with the corresponding Rin and Rout and identify the concepts, or data exchanged between them in order to determine PerceivedRecombination and PerceivedNecessitation. These concepts, or data, depend on the specific domain context and on the information roles have to exchange in order to pursue the objective of the organization, hence in order to transfer the influence between macro and micro level (and vice versa) thus realizing the feedback loop.*

It is obvious that the proposed organizations are not complete. It may be necessary to add roles and interactions between these roles in order to meet the overall objectives of the system. Such roles are problem-specific and may have no influence or link with the feedback loop.

4.7. Capacities identification

A feedback loop involves, for the role corresponding to the macro level (and the micro level), the presence of two kinds of capacities capable of perceiving results from the environment and producing the effects *MicroEffect* and *MacroEffect*. The idea is that Rin/Rout macro must have the ability to perceive (through *Perceive-Recombination Capacity*) the recombining effect and to calculate the corresponding reaction in the feedback loop, or *macroBehavior Capacity* and Rin/Rout micro have the capacity of perceiving necessitating and of transmitting influences to the macro level, i.e. *Perceive-Necessitation and microBehavior Capacity*.

Guideline 7. *In each level Rins have the capacity of perceiving the results of the recombination and the result of the necessitation, hence how the environment influences the loop; so, looking at the problem and at the recombining/necessitating predicates, respectively, identifies Perceive-Recombination Capacity and the Perceive-Necessitation Capacity. As regards the Routs in each level change the macroBehavior/microBehavior in the related capacity.*

These capabilities can then be implemented differently in the design of agents.

5. Case study

A microgrid is a part of a low voltage electric power system that may include distributed power generators (and potentially renewable), storage systems and some of the network users. The problem of energy management in a microgrid includes many sub-issues such as demand management, production management, systems management, storage and network stability (Vasiljevska et al., 2013; Basso et al., 2013).

There are many solutions for energy management in a microgrid. Each solution usually deals with a specific problem such as production planning, quality of energy, self-healing, and adaptation of consumption to production. In all the current work, the resolution of these problems is still carried out independently. In other words, there is no unified way to solve a subset of these problems.

The approach incorporating feedback loops allows the introduction of several levels of problem within the same network. In order to take into account all the objectives of smart grids, we suppose that there exist distributed energy sources (renewable), storages and controllable devices within homes. These controllable devices are characterized by the fact that both users and some decision making mechanism can have an influence on them. These devices either sources or loads can thus be switched on/off, delayed or regulated (by energy production/consumption setting) according to a given specific situation.

The economic aspect of the problem naturally separates into two parts. The producers want to sell at the highest possible price. As the price of energy varies along time, production should vary accordingly to improve benefits for the producers. Consumers, for their part, want to reduce their bills. For this, a system in each home will take into account the price of energy to switch on or off certain devices. These decisions must also take into account the needs of users.

Finally, to ensure the viability of the network, a system of stability control is deployed. This system aims to ensure a continuous balance between supply and demand.

With these constraints, we can take into account most sustainability criteria. It remains to define how to group these constraints within the same problem. For this, we need to define an ontology of this problem.

Fig. 6 shows an example of Problem Ontology Description for the Supply and Demand Matching problem just described. In this problem, we have identified the concepts of Grid and Device. A Grid is a power network. Each of these power networks can be decomposed into smaller networks that can have different characteristics as, for example, a lesser voltage. These smaller networks are represented by the *Microgrid* concept. In turn, each *Microgrid* is composed of elements called *Device*. These elements represent electrical devices that either, at a given moment in time, consume or produce power.

The two types of *Device*, *Load* and *Source*, inherit from *Device*. A third type of *Device*, *Storage* can either be a *Load*, in case of charge, or a *Source*, in case of production.

In order to work properly, a *Grid* needs an *Aggregator* that defines the prices for the energy flowing through it. The definition of such prices is represented by the *SetPrices* action. The *Aggregator* is the actor responsible for this action. The parameters that influence this action are the *EnergyFlow*.

Each *MicroGrid* is an open system in the sense that *Device* can switch on/off without any reliable prediction mechanism. These devices influence the *MicroGrid* with their respective *EnergyFlow* which can be either due to production or consumption. In case of production (resp. consumption) it is a positive (resp. negative) flow.

Once the Problem Ontology (Fig. 6) has been identified for a domain, the analyst may carry on the feedback loop identification activity by following guidelines 1–4 and by looking at Fig. 3. The first step is to identify which concepts may be a candidate actors (Micro and Macro); for doing this let us look for two concepts related by an aggregation relationship. *Grid* and *Microgrid* correspond to this situation. *Grid* performs an action *DefineEnergyPrice* that has a result, *EnergyPrice*, and a predicate as argument, *ExtEnergyBalance*; these three elements cover the left top part of Fig. 3 and may be candidate for, respectively, ActorMacro, MacroBehavior, MacroEffect and Recombining. In order to close the left arch of the loop, we consider the arguments of *ExtEnergyBalance*, *SoldEnergy* and *EnergyFlow*, and consider them, respectively, MicroEffect and RecombiningFactors. Going on in exploring the POD

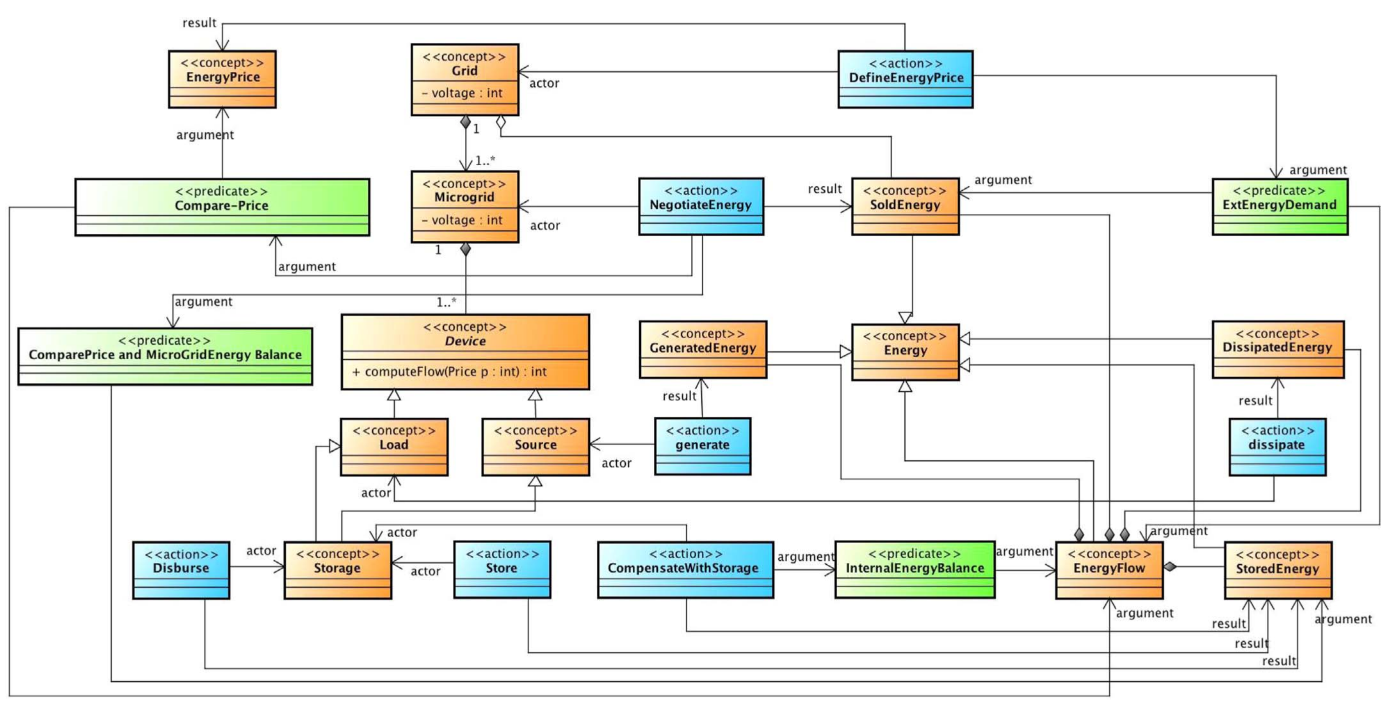

Fig. 6. Problem ontology description for the GRID example.

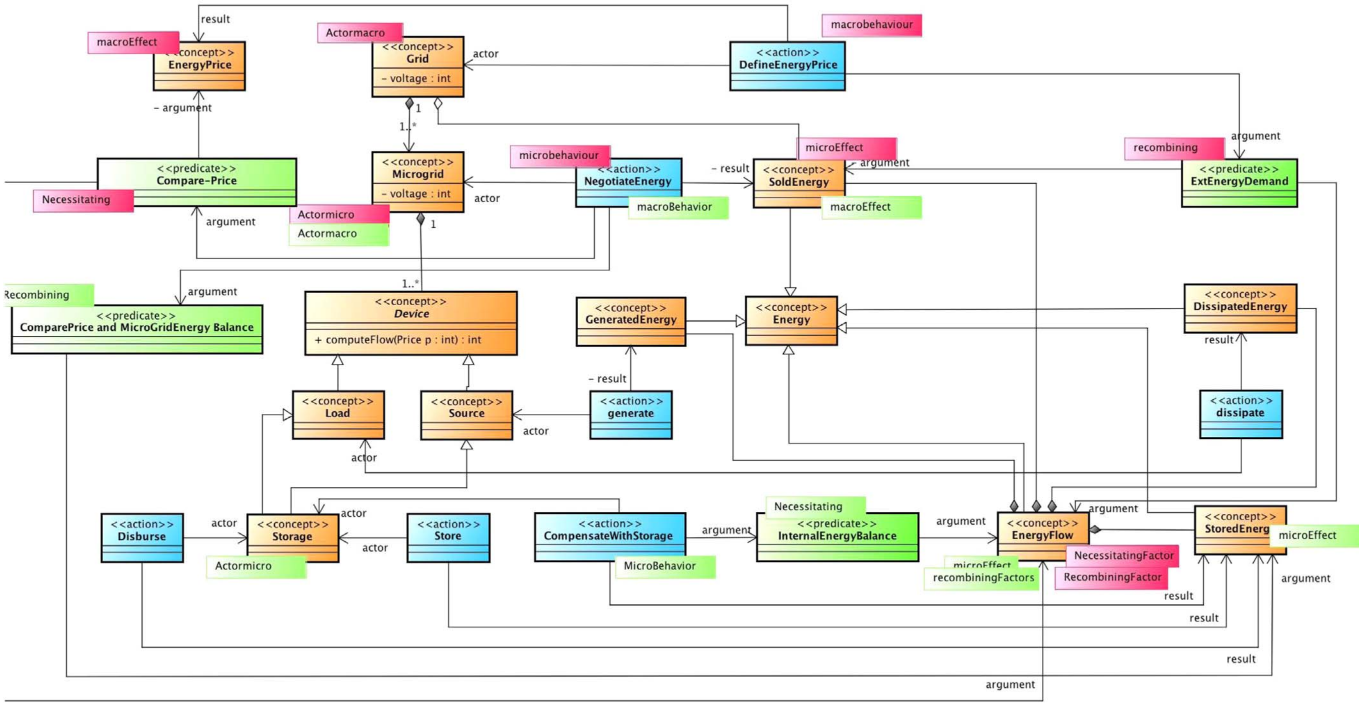

Fig. 7. Problem ontology description for the GRID example – feedback loops identification. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

in such a fashion, we identify two overlapping different feedback loops, they are labeled in red and green color in Fig. 7 and are resumed in Fig. 8 and 9.

The two loops confirm that the Supply and Demand Matching is a combinatory system of accumulation, indeed (looking at Fig. 1) the combinatory effect in the upper loop may result from the MicroGrids' action, Fig. 8 . Suppose that a MicroGrid decides (the beginning by chance) to introduce energy in the grid as the result of negotiating its energy. The new energy sold by each MicroGrid modifies the energy balance in the grid. This fact provokes the emerging behavior of the Grid that establishes a new price convenient for selling, hence a macro effect that forces all the other microgrids to sell energy rather than to store it. Note that in this case Energy Flow is at the same time Necessitating Factor and Recombining Factor; this fact does not contradict Mella's Theory. The accumulation effect is realized by the increasing of energy flow present in the grid.[3]

As regards the second loop (Fig. 9), the combinatory effect of accumulation is due to the need each Storage has of balancing its internal energy; suppose it decides to store energy (the micro effect), this fact produces an emerging behavior inducing the microgrid to have a negative value of sold energy which forces other Storages to continue in storing energy.

These two feedback loops imply the identification of two different organizations, illustrated, respectively, in Figs. 10 and 11. Let us look to the organization coming from the upper loop, following guidelines 5–7 MicroGrid and Grid, respectively, ActorMicro and ActorMacro, entail two couples of roles:

1. Rin-Grid and Rout-Grid responsible for perceiving the recombination effect and actuating the macro behavior through two capacities, *Perceive-EnergyBalance* and *DefineEnergyPrice*.

[3] In these example we do not intentionally deal with the chance that starts the loop and the event that forces the accumulation to stop because we want to point our attention on how we identify feedback loop as first class entity in order to identify organizations.

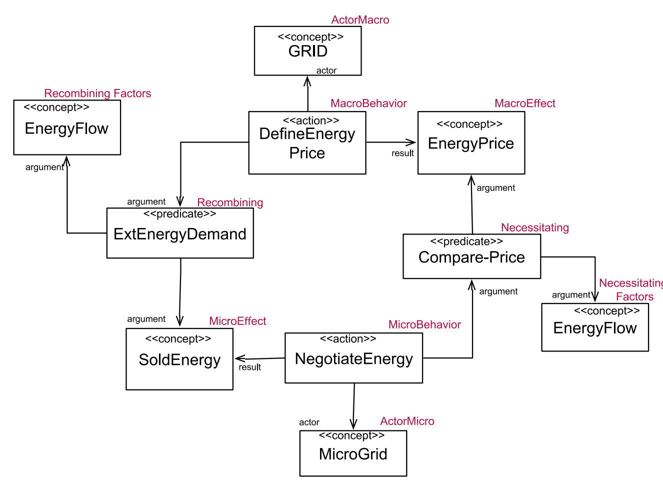

Fig. 8. Identification of the upper feedback loop in the POD.

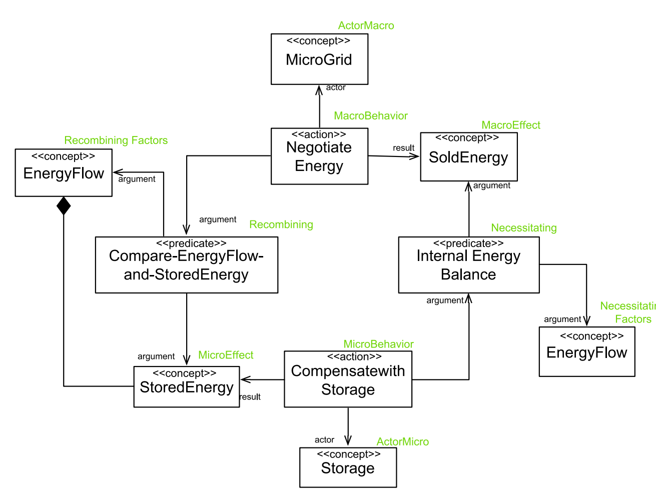

Fig. 9. Identification of the lower feedback loop in the POD.

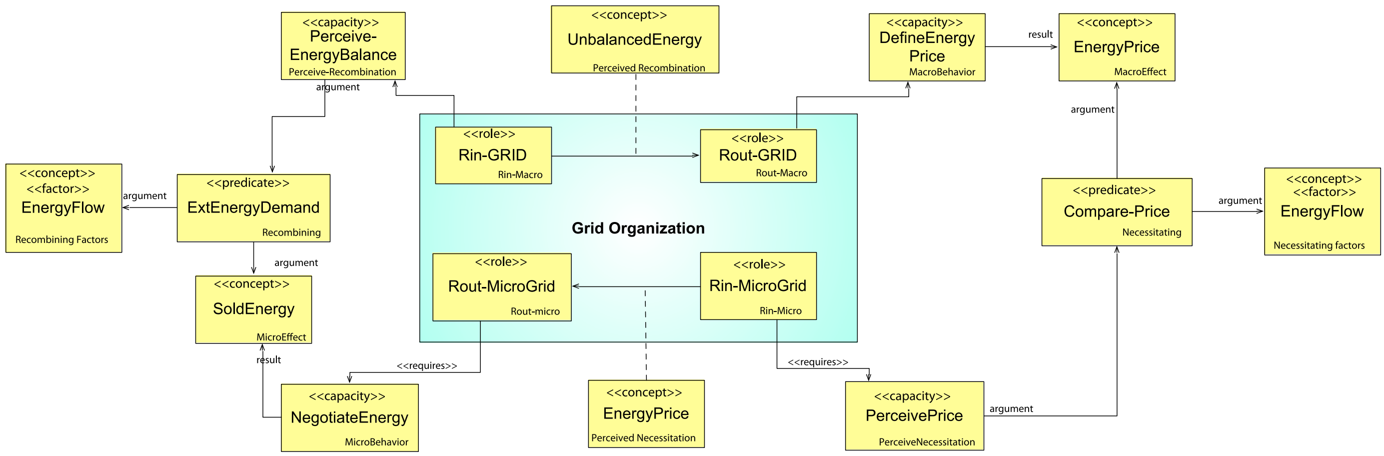

Fig. 10. The first organization corresponding to the upper loop.

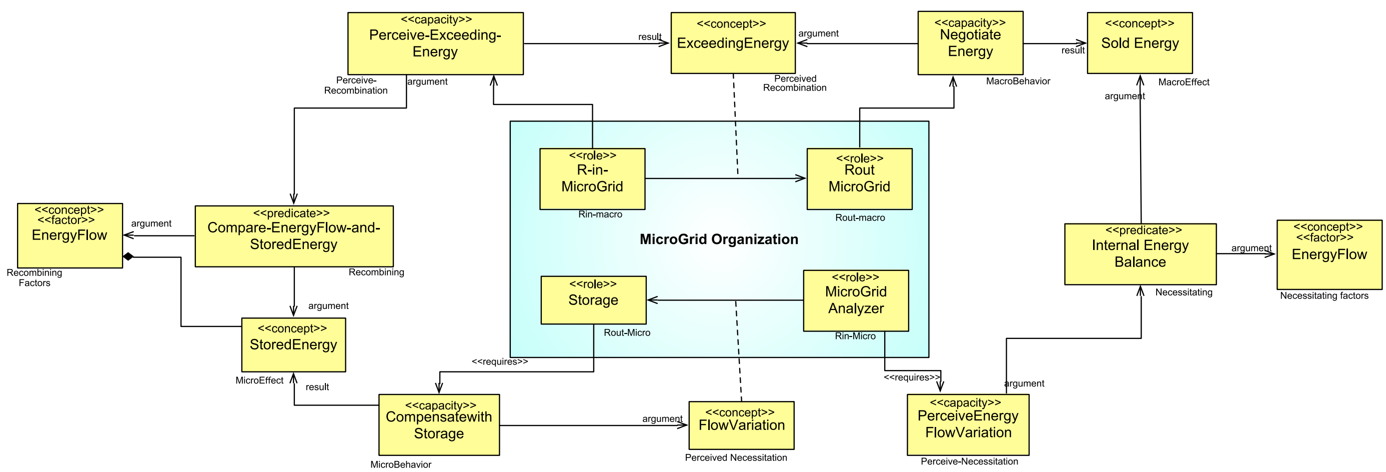

Fig. 11. The second organization corresponding to the lower loop.

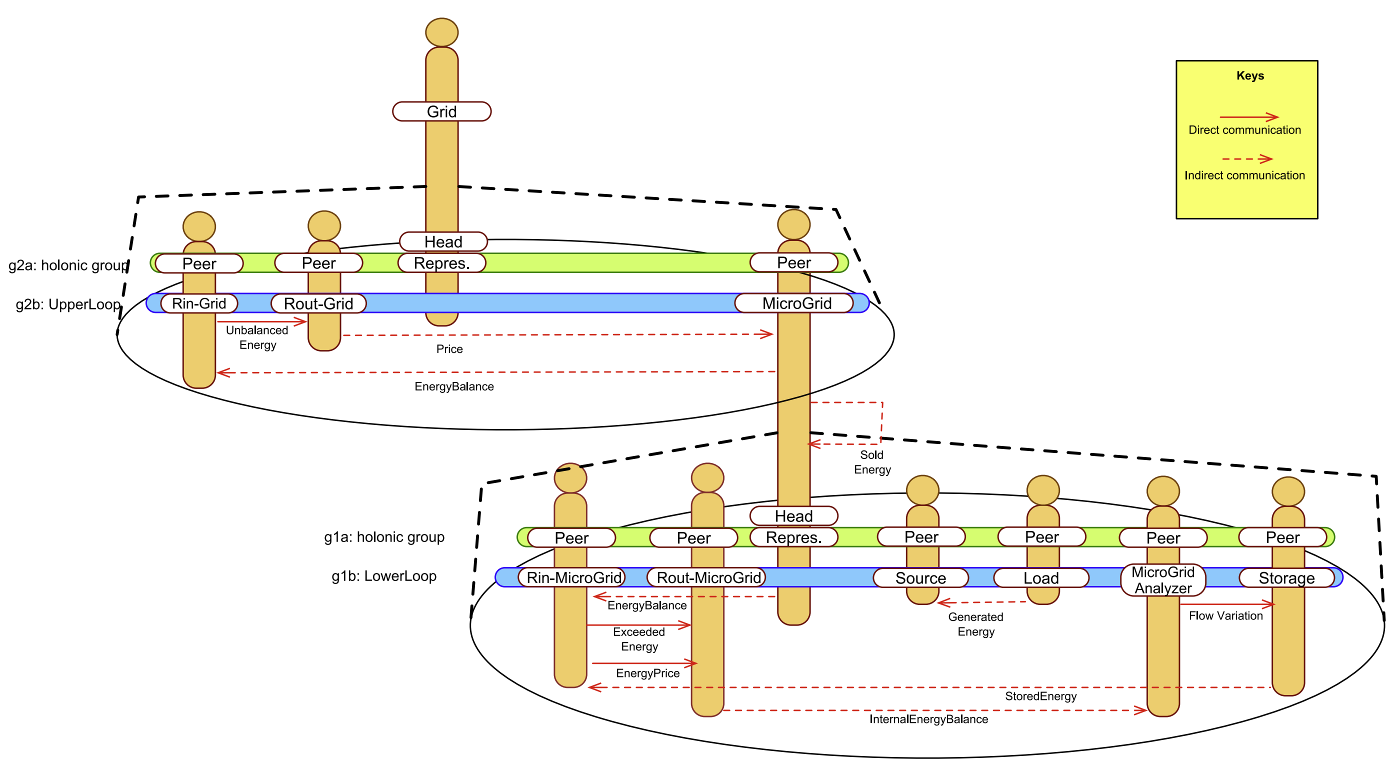

Fig. 12. The organizational structure exemplified.

Moreover, Rin directly communicates to Rout the perceived recombination using the UnbalancedEnergy concept.

2. *Rin-MicroGrid* and *Rout-MicroGrid* responsible for perceiving necessitating effect and exposing the micro behavior through two capacities, *PerceivePrice* and *NegtiateEnergy*. They also directly communicate by involving the *Price* concept

An example of the MAS structure of the smart grid is reported in Fig. 12. This structure is represented using a modified version of the cheeseboard diagram. Each agent in the system plays at least one role inside a group. Roles are represented using rounded-corner squares. Following Holonic MAS theory, agents (holons) can be composed of other agents (members). The members are coordinated using one or more production groups that instantiate an organization. The top-level agent (holon) plays the "Grid" role and represents the overall behavior exhibited by the system. Internally, its members interact according to the upper-loop organization described in Fig. 10 and tagged "g2b" in the diagram. Inside "g2b" multiple "Microgrid" holons may interact (only one is depicted for simplicity). Each "Microgrid" coordinates its members using the lower-loop organization (Fig. 11). This group is tagged "g1b".

It is important to notice that agents interact using direct and indirect communication. Direct communication represents messages between roles as specified in the organizations. For instance, the "UnbalancedEnergy" message between "Rin-Grid" and "Rout-Grid" in "g2b". On the other hand, agents also communicate indirectly as a result of the feedback loop via their "Perceive-Recombination" and "Perceive-Necessitation" Capacities (see Fig. 5). In order to represent this complex set of interaction, we use a dashed arrow. For example, the dashed arrow labeled "Price" in the cheeseboard diagram in "g2b" represents the following of information between "Rout-Grid" and "Rin-MicroGrid" in Fig. 10. This flow is triggered when "Rout-Grid" calls its "DefineEnergyPrice" capacity and finally arrives at "Rin-Microgrid" via its "Perceive-Price" capacity (see right-hand concepts in Fig. 10).

As described above, agents' behaviors following a feedback loop pattern are influenced by direct messages and indirect communications resulting from the combinatory automaton proposed by Mella. The system presented was implemented and validated in Basso et al. (2011). The validation consisted in comparing the developed simulator with SimPowerSystems which is, in the energy community, a well-known and widely used simulator.

6. Conclusion

In this paper, we present an approach for employing feedback loops as a first-class entity for the identification of multi-agent organizations. Feedback loop (FL) is a way of modeling complex systems that expose emergent behavior by means of a cause-effect loop between the micro and the macro levels of the system. The underlying principle, issued from the work on Combinatory Systems of Mella (2008), is that complex systems which can be modeled as combinatory systems own two different levels, micro and macro, composed of agents; agents' behavior at the micro level influences the agent at the macro level that forces the micro level, as a consequence, in a continuous loop. The established loop between micro and macro levels results in the regulation of interactions among different parts of a system hence the self-organizational trend.

Starting from this theory, we propose an approach for modeling feedback loops and the related complete methodological procedures. We have firstly illustrated the abstract representation of the FL by means of concepts, predicates and actions, and then the guidelines for identifying FLs from the problem domain description (POD) down to the identification of organizations. In order to validate our proposal, we have used a two step approach. First, the proposed methodological approach was applied on the examples used by Mella in his work to assess that the initial Mella concepts were reproducible by using the proposed approach. Second, another system, a smart grids software simulator, was analyzed, modeled and developed (Basso et al., 2011). The functioning of this simulator was validated against a professional electrical simulator.

It is worth noting the important role of the ontology from which candidate FL are retrieved and the fact that identification of FLs, and then of organizations, has proven to be advantageous for refining domain description for all those systems intrinsically exhibiting emergent behavior and for which classical approach is not natural and intuitive cause the lack of right abstractions for managing their features.

Future works will consist in refining the presented guidelines and approach in order to provide guidance and assistance for analysts/designers. Another possible direction can be the definition of a CASE tool[4] supporting this kind of analysis by providing assistance such as semi-automatic detection of FL candidates. Considering advances within, for example, ADACOR and PROSA, we also plan to study mechanisms that allow a prediction (or anticipation) of the future state of the system in order to improve the FLs engineering approach presented in this paper.

References

Barata, J., 2006. The Cobasa architecture as an answer to shop floor agility, manufacturing the future. In: Kordic, V., Lazinica, A., Merdan, M. (Eds.), In Tech, http://dx.doi.org/10.5772/5044.

Basso, G., Lauri, F., Hilaire, V., Roche, R., Cossentino, M., 2011. A MAS-based simulator for the prototyping of smart grids. In: European Workshop on Multi-agent Systems.

Basso, G., Gaud, N., Gechter, F., Hilaire, V., Lauri, F., 2013. A framework for qualifying and evaluating smart grids approaches: focus on multi-agent technologies. Smart Grid Renew. Energy 4 (4), 333–347.

Beurier, G., Simonin, O., Ferber, J., 2003. Un modèle de système multi-agent pour l'émergence multi-niveau. Tech. Sci. Inf. 22 (4), 235–247, URL 〈〉.

Beurier, G., Simonin, O., Ferber, J., 2012. Model and simulation of multi-level emergence. In: Proceedings of IEEE 2nd IEEE International Symposium on Signal Processing and Information Technology, ISSPIT, pp. 231–236.

Brückener, S.A., 2000. Return From the Ant—Synthetic Ecosystems for Manufacturing Control (Ph.D. thesis).

Brun, Y., di Marzo Serugendo, G., Gacek, C., Giese, H., Kienle, H.M., Litoiu, M., Muller, H.A., Pezze, P., Shaw, M., 2009. Engineering self-adaptive systems through feedback loops. In: Lecture Notes in Computer Science. Springer, Berlin, Heidelberg.

Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: Prosa. Comput. Ind. 37, 255–274.

Caprarescu, B., Petcu, D., 2009. A self-organizing feedback loop for autonomic computing. In: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World: 2009, pp. 126–131.

Cossentino, M., Gaud, N., Hilaire, V., Galland, S., Koukam, A., 2010. ASPECS: an agent-oriented software process for engineering complex systems—how to design agent societies under a holonic perspective. Auton. Agents Multi-Agent Syst. 2 (2), 260–304.

Cossentino, M., Hilaire, V., Molesini, A., Seidita, V. (Eds.), 2013a. The ASPECS Process, 1st edition Springer, Berlin, Heidelberg, pp. 216–236 (Chapter 4).

Cossentino, M., Hilaire, V., Molesini, A., Seidita, V. (Eds.), 2013b. The SODA Methodology: Meta-Model and Process Documentation, 1st edition Springer, Berlin, Heidelberg, pp. 216–236 (Chapter 13).

Dobson, S., Denazis, S.G., Fernández, A., Gaïti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F., 2006. A survey of autonomic communications. ACM Trans. Auton. Adapt. Syst. 1 (2), 223–259.

Ferber, J., Gutknecht, O., Michel, F., 2003. From agents to organizations: an organizational view of multi-agent systems. In: Giorgini, P., Müller, J.P., Odell, J. (Eds.), AOSE, Lecture Notes in Computer Science vol. 2935. Springer, Berlin, Heidelberg, pp. 214–230.

FIPA, Fipa rdf Content Language Specification. Technical Report XC00011B, 2001.

Gaud, N., Galland, S., Hilaire, V., Koukam, A., 2009. An organisational platform for holonic and multiagent systems. In: Hindriks, K., Pokahr, A., Sardina, S. (Eds.), Programming Multi-Agent Systems, Lecture Notes in Computer Science vol.

[4] Computer Aided Software Engineering tool.

5442. Springer, Berlin/Heidelberg, pp. 104–119.
Gerber, C., Siekmann, J., Vierke, G., 1999. Holonic Multi-Agent Systems. Research Report RR-99-03, DFKI.
Giret, A., Botti, V.J., 2009. Engineering holonic manufacturing systems. Comput. Ind. 60 (6), 428–440. http://dx.doi.org/10.1016/j.compind.2009.02.007.
Heras, S., Botti, V., Julián, V., 2014. Modelling dialogues in agent societies. Eng. Appl. Artif. Intell. 34, 208–226.
Hilaire, V., Cossentino, M., Gechter, F., Rodriguez, S., Koukam, A., 2016. An approach for the integration of swarm intelligence in mas: an engineering perspective. Expert Sys. App., 40 (4), 1323–1332, ISSN 0957-4174, http://dx.doi.org/10.1016/j.eswa.2012.08.058.
Mariani, S., Omicini, A., 2015. Coordinating activities and change: an event-driven architecture for situated mas. Eng. Appl. Artif. Intell. 41, 298–309.
Mella, P., 2000. Combinatory System Theory. Technical Report. University of Pavia. URL ⟨http://ea2000.unipv.it/Sistemi_combinatori/genindex.htm ⟩.
Mella, P., 2008. Observing collectivities. The combinatory systems approach in social sciences. Int. J. Interdiscip. Soc. Sci. 3, 213–223.
Ogata, K., 1997. Modern Control Engineering, 3rd edition. Prentice Hall.
Parunak, H.V.D., 1997. Go to the ant: engineering principles from natural multi-agent system. Ann. Oper. Res. 75, 69–101.
Schmickl, T., Stradner, J., Hamann, H., Winkler, L., Crailsheim, K., 2011. Major feedback loops supporting artificial evolution in multi-modular robotics. In: Doncieux, S., Bredèche, N., Mouret, J.-B. (Eds.), New Horizons in Evolutionary Robotics, Studies in Computational Intelligence vol. 341. Springer, Berlin, Heidelberg, pp. 195–209.
Vasiljevska, J., Lopes, J.P., Matos, M., 2013. Integrated micro-generation, load and energy storage control functionality under the multi micro-grid concept. Electr. Power Syst. Res. 95, 292–301, URL ⟨⟩.
Wolf, T.D., Holvoet, T., 2007. Designing self-organising emergent systems based on information flows and feedback-loops. In: SASO. IEEE Computer Society, pp. 295–298.