A FORMAL HOLONIC FRAMEWORK WITH PROVED SELF-ORGANIZING CAPABILITIES

SEBASTIAN RODRIGUEZ and VINCENT HILAIRE* and PABLO GRUER and ABDER KOUKAM
Systems and Transports Laboratory, University of Technology of Belfort Montbéliard, 90010 Belfort Cedex, FRANCE
vincent.hilaire@utbm.fr

Received (Day Month Year)
Revised (Day Month Year)

Numerous works aim to design agents and multi-agent systems architectures in order to enable cooperation and coordination between agents. Most of them use organizational structures or societies metaphor to define the MAS architecture. It seems improbable that a rigid unscalable organization could handle a real world problem, so it is interesting to provide agents with abilities to self-organize according to problem’s objectives and environment dynamics. We have chosen the holonic paradigm to provide these abilities to agents. Holons are recursive self-similar entities which are organized in an emergent society - an holarchy. The aim of this paper is to present a formally specified framework for holonic MAS which allows agents to self-organize. The framework is illustrated by an example drawn from a real world problem. Some pertinent properties concerning the self-organizing capabilities of this framework are then proved.

Keywords: Formal specification; Engineering emergent societies; Organizational structures and roles.

1. Introduction

Numerous works aim to design agents and multi-agent systems (MAS in the sequel) architectures in order to enable cooperation and coordination between agents. Most of them use organizational structures or societies metaphor to define the MAS architecture. The MAS organizational structure facilitates agent cooperation and coordination in order to solve a given problem. It seems improbable that a rigid unscalable organization could handle a real world problem, so it is interesting to provide agents with abilities to self-organize according to problem’s objectives and environment dynamics. The organizational structure emerges from agents interactions. Even if self-organized MAS are recognised as useful, they are still difficult to engineer. When massive number of autonomous components interact it is very difficult to predict that the emergent organizational structure fits the system goals.

*corresponding author
or that the desired functionalities will be fulfilled. The aim of this paper is to present a formally specified framework for Holonic MAS (HMAS in the sequel) which allows agents to self-organize. The framework is illustrated by an example drawn from a real world problem. We prove some pertinent properties concerning the self-organizing capabilities of this framework.

The term holon was originally introduced in 1967 by the Hungarian Philosopher Arthur Koestler\(^1\) to refer to natural or artificial structures that are neither wholes nor parts in an absolute sense. According to Koestler, a holon must respect three conditions: (1) being stable, (2) having the capability of autonomy and, (3) being capable of cooperation. The stability means that a holon is capable of reacting when strong perturbations occur. The autonomy implies that a holon is capable of self-management in order to achieve its own goals. The capability of cooperation denotes that holons are capable of working in common projects according to shared goals with other holons or other layers of holons.

Holonic organizations have proven to be an effective solution to several problems associated with hierarchical self-organized structures (e.g. \(^2\), \(^3\), \(^4\), \(^5\), \(^6\)). In many MAS applications, an agent, that appears as a single entity to the outside world, may in fact be composed of several agents. This hierarchical structure corresponds to the one we find in Holonic Systems. Frameworks have been proposed to model specific problem domains, mainly in Flexible Manufacturing Systems (FMS) and Holonic Manufacturing Systems (HMS), such as PROSA\(^7\) and MetaMorph\(^8\). The Holonic paradigm has also been applied in other fields such as cooperative work\(^9\). However, these frameworks and models are strongly related to a specific domain of application, and may be difficult to apply to other domains.

Our framework isn’t domain-dependent so it can be easily reused. This framework is based upon organizational concepts which have been successfully used in the MAS domain\(^10\),\(^11\),\(^12\). We base our approach on the Role-Interaction-Organization (RIO) Methodology. Even if other methodologies, such as GAIA\(^12\), are based on similar concepts, RIO is the only one, to our knowledge, with proof support. Indeed, RIO uses a specific process and a formal notation (OZS) that is described in\(^11\). OZS is a multi-formalisms notation that integrates in Object-Z classes\(^13\) a statechart\(^14\). OZS classes have constructs for specifying functional (Object-Z) and reactive (statecharts) aspects. We have defined a formal semantics for OZS\(^15\). This semantics is based upon the translation of Object-Z and statecharts in transition systems and allows the use of theorem prover and model checker such as SAL.

In order to illustrate this approach we will take as example the adaptive mesh problem. A distinguishing feature of cellular radio mobile networks is the rapid increase of the consumer demand and the ensuing complexity in their design and management. Responding to this demand requires to partition the geographical area into a number of service units or cells. The adaptive meshing problem for mobile network dimensioning considers traffic statistics as a predefined resource that must be attributed to many adaptive low power Base Transceiver Station (BTS in the sequel). This is a fundamental step in the design of a radiomobile network.
The paper is organized as follows: section 2 presents the holonic framework and illustrates it with the meshing problem, section 3 presents RIO and specifies the holonic framework, the section 4 describes proven properties and eventually section 5 gives a conclusion and future works.

2. Holonic Framework Overview and Example

2.1. Meshing Problem

The meshing system takes as input a discretized environment containing the communications inside each region. This information is obtained by discretizing a specific geographical area, as shown in figure 1. A regular grid is used to obtain a matrix where each element represents the communication traffic inside that region. The values are obtained either by measuring on the field, possibly corrected including future requirements, or by empirical estimations. This matrix represents the input of adaptive meshing, and is called Resource Matrix.

The result of the system is an Adaptive Mesh. The mesh is composed of a number of Cells. Each Cell covers a certain communication traffic. We illustrate this process in figure 2. The size of a cell depends on the sum of communication traffic values of the resources covered.

The mesh, composed of cells, must cover the resources in the matrix while respecting a number of constraints. The first constraint is a geometrical one, where we impose that each cell should respect a given geometrical shape. The BTS is placed inside a Cell to cover the communication. The cell may have different geometries. Figure 3 shows a rectangular (a) and a hexagonal (b) geometry. In this application we will consider a rectangular geometry.

The second constraint is the maximal communication traffic covered by a cell. This should not exceed the capacity of a BTS. So, the resulting mesh should be composed of cells that respect both constraints.
In order to treat this problem using a HMAS, each resource of the Resource Matrix will be assigned with a holon (Resource Holon) whose main and unique goal is to become a part of a cell (Cell Holon) that will cover the communication traffic. The set of Cell Holons compose then the Mesh Holon, whose structure represents the solution to the problem.

The complete holonic solution of the adaptive meshing problem can be found in\textsuperscript{16}. In this paper we only use it to illustrate some concepts of the HMAS framework.

\subsection*{2.2. HMAS Roles and Organizational Structure}

The HMAS Structure can be seen as a set of hierarchical levels. This hierarchical structure is called a holarchy. For clarity reasons, at a given level, we will call super-holon the higher-level entity composed by its members or sub-holons.

The behaviour of the members of a super-holon and their interactions are described in terms of roles. Roles in RIO are defined as generic behaviours or status in an organization. These roles represent the status of the holon inside a specific super-
The first question we should answer concerns the internal structure of a super-holon. Indeed, three different proposition were presented by 17. Our approach is based in an HMAS as a moderated group, where the head represents the members of its super-holon with the outside world 17. This decision is based on the wide range of possible configuration that can be obtaineda. Different ways to select this representative can be stated, eg. voting, authority, predefinition, etc. However, selecting the most suited one remains problem dependent.

In a complex system, multiple holarchies can be identified. A holarchy should be seen as a loose hierarchy in the sense of 18, where there are no subordination relation.

A moderated-group can be modeled by an organization as presented in figure 4. As the representative, the holon plays the Head role. According to the objective and rules of the super-holon, the Head’s responsibilities and rights may range from merely administrative tasks to being able to take decisions concerning all members. The head is not necessarily an unique holon. After a holon starts performing the Head role, it will be the representative of the members of its super-holon ; and therefore may be able to engage the super-holon in new tasks.

Members not playing the Head role are considered as Parts of the super-holon. These members may confer a certain authority to their Head. Once a holon is accepted in a super-holon, its autonomy is reduced because of its obligations with the super-holon. The degree of this autonomy lost may variate according to the

aThe reader may refer to 16 or 17 for further discussion on the subject.
super-holon’s purpose.

The **Multipart** role is a special case of the Part role. This role is played by holons belonging to more than one super-holon. Interesting possibilities are available when a holon is shared. For instance, we can now see this holon as gateway between super-holons, allowing message forwarding. Imagine that holon $h$ is shared by super-holons $H_1$ and $H_2$. Suppose that $H_1$ confers to $h$ the authority to accept new members coming from $H_2$. Now, $h$ can not only forward request among members of different super-holons, but also act as an ambassador of $H_1$ inside $H_2$. This can be used to reduce the administrative load to the Head of $H_1$, but also to provide means for members of $H_2$ to enter a new holarchy. Other possibilities can be used like trust mechanism based on recommendations of shared members, translation of messages in different languages, etc.

As a holon joins a HMAS Organization, it has no special bindings and does not collaborate with any other holon. This situation represents a **Stand-Alone** behavior. In this state, the holon’s decisions are not attached to any restriction but its own goals and objectives. The holon will remain in this state as long as it is satisfied. The Stand-Alone represents how non-members are seen by an existing super-holon.

Following the holonic paradigm, the holon seen as a Stand-Alone can actually be a super-holon at a lower level. Stand-Alone holons will interact only with the representatives (heads) of the super-holons in order to request one of the services they provide. According to its needs, a holon may want to join an existing super-holon and request to its head its admission as a member. This process is called **Merging** and is represented in the RIO diagram of fig. 5 by the **Merging Interaction**. If accepted, the holon becomes a part of that holon (Merge). From that moment on, until it leaves the holon, either by self decision or command of the holon’s head, it can directly interact with the member of the holon.

These holonic roles represent a generic framework describing high-level behaviours and interactions between components of Holonic System. Examples of such systems can be found in every-day life. In a human organization, for instance, it is extremely difficult to describe all the interactions a person may have. Someone can be an em-
ployee (Part) of a company, while at the same time, president (Head) of a club. Even more, the president of the company may be a member of the club. Trying to describe all these relations and possible interactions with only one perspective on the subject increases the difficulty of process. On the other hand, if we describe it using several holarchies we can isolate a set of interaction and behaviours that are important within a given context. Like this we will describe the Company Holon as being composed of Department Holons that are composed of Employees Holon. We will decompose the Club holon in associates, directive board and a president. In figure 6, we can see a possible disposition of a holarchy. Each member of a super-holon is tagged (in super-index) with its holonic role. It is important to keep in mind, that Head, Part, MultiPart and Stand-Alone are roles and they describe the status and interactions of members of a holon.

2.3. Roles Dynamics and Self-Organization

In order to enable holons to dynamically change their roles, we define a satisfaction based on the progress of their current tasks. This satisfaction, called instant satisfaction, depends on the played role and is calculated using the following definitions.

The Self Satisfaction ($SS_h$) of the holon $h$ produced by its own work.

The Collaborative Satisfaction ($CS_h^H$) is the satisfaction produced for the holon $h$ by its collaboration with other members of the super-holon $H$. This satisfaction can be either positive, when the other members’ work help $h$ in its task, or negative,
when the other members’ work imposes barriers to the achievement of the holon’s task.

The Accumulative Satisfaction (\( AS_h \)) is the satisfaction produced for the holon \( h \) by its collaboration with members of multiple super-holons. This satisfaction is only used when the Multi-Part role is allowed, i.e. holon \( h \) may belong to more than one super-holon. When a holon belongs to a super-holon and it is unsatisfied, two options are available. The holon may quit its current super-holon and join a new one, or it may join a second super-holon without leaving the first. This satisfaction guides the decision in this situation.

\[
AS_h = \sum_p CS^p_h \quad \forall p \in \text{superholon}(h) \quad (1)
\]

where the \( \text{superholon} \) function returns the super-holons of \( h \).

The Instant Satisfaction (\( IS_h \)) is then defined as Current satisfaction of holon \( h \)

\[
\forall h \in \text{HMAS} \quad IS_h = \begin{cases} 
CS_h + SS_h & \text{if } R_h = \text{Part} \lor R_h = \text{Head} \\
AS_h + SS_h & \text{if } R_h = \text{MultiPart} \\
SS_h & \text{if } R_h = \text{Stand-Alone} 
\end{cases} \quad (2)
\]

Where \( R_h \) is the role played by the holon \( h \).

The Necessary Satisfaction (\( NS \)) is the required satisfaction to achieve the goal or task according to a given problem constraints. This satisfaction is not necessarily a fixed value.

Using these definitions we can obtain the transitions between the different roles, shown in figure 7. Conditions \( C_{MP} \) and \( C_{MH} \) are defined by the merging process and \( C_{PH} \) by the members’ interactions. Indeed, if the merging is a success the new member will be either part if \( C_{MP} \) is true or Head if \( C_{MH} \) is true. The \( C_{PH} \) becomes true if the Part holon concerned is chosen to be Head of the holon. All these conditions are domain dependant. The notation \( AS^1 \) and \( AS^2 \) is used to represent the \( AS \) when belonging to one or two super-holons.

In the adaptive mesh problem, the goal of the stand alone holon must ensure the
coverage of its resource, then it will try to join a mesh immediately. The only situation where it remains in a stand-alone role is when its resource can get an antenna of its own, which is a highly improbable situation. We can then say that $SS$ represents the holon’s resource. If we look at equation 2 we see that $IS = SS$. Then if the holon’s resource is not big enough to receive an antenna of its own, it will try to merge.

A holon that performs the head role will be responsible to respect the constraints of a cell. It will be representing a possible cell in the system. So head’s will accept or refuse other holon’s requests to merge according the constraints. Although all heads represent possible meshes in the system, a super-holon head can decide to leave its role if, after trying to improve the super-holon’s satisfaction, the satisfaction is insufficient to remain as a super-holon. In order to improve the Holon’s satisfaction, it will accept new agents to increase the super-holon covered resource, or will command member agents to leave the super-holon if they don’t respect the geometrical constraints or if the covered resource has exceeded the maximum.

In the adaptive mesh problem, the agent gets the Part role if negotiations with a super-holon succeeded. It will remain in the super-holon if its satisfaction level is raising. In order to calculate this value, we will consider $CS^H$ as the sum of the resources of the members of the super-holon $H$. If the agent remains in the super-holon, its goal will be accomplished and will stay in the HMAS until the antenna is finally assigned. However, it is also possible that the holon receives a command to leave the super-holon, in that case, it must return to Stand-Alone and restart the merging process. The Multi-Part role is a special case of the Part role. This state is reached when an agent belongs to more than one super-holon. There are several situations to consider (e.g: requests conflicts, forwarding requests, etc) however these issues are out of the scope of this paper. In the presented example, a resource must be covered only by one antenna, then no holon can assume this role. However, other approaches could be used to enable this behavior, such as overlapping meshes for example.

2.4. Affinity

A holon that is interacting within a HMAS will change its role according to its needs and it will try to merge with existing holons. The whole merging system is inspired by the Immune System. According to this approach, it is possible to determine if two holons are suited to work together considering their goals and services.

Each holon has an identifier which gives all relevant information. As in the immune system, an holon can find out the affinity it has with another by comparing its identifier with the identifier of the other holon, then uses this affinity to decide whether or not to merge. Several definitions of affinity can be found depending on the problem’s nature. In a general sense we can say that two holons have a high affinity with each other if their goals are similar and their services complementary. Once again, several approaches can be taken to define when two goals are similar.
In the meshing example, the holon’s identifier should give the position of the holon’s resource (X, Y coordinates) and the traffic it contains. Using these values, an holon can determine whether or not to merge.

As explained before, the affinity should give a measure of the compatibility of the holon’s goal and services. In this particular case, both holons will have the same goal, to ensure the coverage of their resources. Therefore, the main problem is to ensure that the geometrical constraints are respected. The affinity could be decomposed in two main parts. The first part, the distance affinity will provide a geometry dependent value used to ensure that the geometrical constraints are respected. As we need square meshes, we will use two parameters to test the distance affinity. First, we will check if the holon trying to merge is inside the acceptance distance (see Figure 8). If not, the holon will be rejected. However, if the holon is inside this distance, the head will calculate the real distance affinity. The affinity equals the number of resources, that are already part of the mesh, and have an unitary distance with the resource the new holon represents.

Lets consider three different situations shown in Figure 9. Holons in (a) and (b) will be accepted into the holon if the maximum traffic that an antenna can handle has not been reached. If the mesh is close to this value, holon (a) will be privileged. However, holon c will be rejected in all circumstances.

The second part, the resource affinity is used to ensure that the limits of an antenna are not exceeded.

Another important parameter in the holons behavior is the holon’s satisfaction. It enables the holon to move between different roles according to its needs. In the mesh example, we will use the satisfaction to ensure that the holon’s resource is covered by a mesh.
3. RIO Concepts

In this section we present the RIO framework and its extension for Holonic MAS. We use the OZS formalism which is based upon the integration of Object-Z and statecharts. Object-Z is an object oriented extension of Z and thus use set theory and first order predicate logic. Statecharts add hierarchy of state, parallelism and broadcasted communication to finite state automata. Each RIO concept (eg. role) is specified by an OZS class.

3.1. RIO classes

A role is an abstraction of an acting entity. We have chosen to specify it by the Role class. This class represents the characteristic set of attributes whose elements are of [Attribute] type. These elements belong to the attributes set. A role is also defined by stimulus it can react to and actions it can execute. They are specified by stimulus set and actions set respectively. The [Attribute], [Event] and [Action] types are defined as given types which are not defined further.

The reactive aspect of a role is specified by the sub-schema behaviour which includes a statechart. It is to say that the behaviour schema specifies the different states of the role and transitions among these states. The obtainConditions and leaveConditions attributes specify conditions required to obtain and leave the role. These conditions require specific capabilities or features to be present in order to play or leave the role. Stimulus which trigger a reaction in the role behaviour must appear in one transition at least. The action belonging to the statechart transitions must belong to the actions set. The constraints specified in the Role ensure the coherence between Object-Z and statechart parts.
Role

attributes : \( \mathbb{P} \) Attribute
stimulus : \( \mathbb{P} \) Event
actions : \( \mathbb{P} \) Action
obtainConditions, leaveConditions : Condition

\( \forall s \in \text{stimulus}, \exists e \in \text{behaviour} \bullet (\exists t \in e.\text{transitions} \bullet t.\text{label}.\text{event} = s) \)
\( \forall e \in \text{behaviour} \bullet (\forall t \in e.\text{transitions} \bullet t.\text{label}.\text{action} \subseteq \text{actions}) \)

behaviour

An interaction is specified by a couple of role which are the origin and the destination of the interaction. The role orig and dest interacts by the way of operations \( op_1 \) and \( op_2 \). These operations are combined by the \( \parallel \) operator which equates output of \( op_1 \) and input of \( op_2 \). The \( \Diamond \) symbol is a temporal logic operator which states that eventually the predicate is true. In order to extend interaction to take into account more than two roles or more complex interactions involving plan exchange one has to inherit from the Interaction class.

Interaction

\( \text{orig}, \text{dest} : \text{Role} \)
\( op_1, op_2 : \text{Action} \)
\( op_1 \in \text{orig.action} \)
\( op_2 \in \text{dest.action} \)
\( \Diamond (\text{orig}.op_1 \parallel \text{dest}.op_2) \)

3.2. Framework specification

In our approach each holon may play four roles : Stand Alone, Head, Part and Multi-Part. The class inheritance hierarchy of these roles is presented in figure 10. A Stand Alone role player may interact with Heads in order to enter a specific holon. This interaction is specified by the Merging class which inherits from Interaction. Part role players interact with their Head during the holon life. These interactions consists in command or request. The HolonicRole class inherits from Role class and defines the generic elements for all Holonic role players. These elements are the satisfied boolean, different satisfaction criterions defined in the section 2.3 and some available services accessible with the function availableServices. All following roles inherit from Holon and add specific attributes, operations and behaviours.
A StandAlone role is the entry point of an holarchy. Each agent which is satisfied by its actions plays this role. As soon as it satisfaction is lesser than the necessary satisfaction required to finish the current task, the NS real value (Necessary Satisfaction), the StandAlone actor search an holon to merge with. The [Answer] given type specifies the answers given by neighboor heads in response to fusion request. Operation askToJoin broadcast a request to holons to ask for merging. Operation getBestAnswer choose among the answers received the one which fit the best according to holon criteria.
The class *Merging* specifies the interaction of Merging between two holons. The origin of the merging must be a StandAlone and the destination Head of another holon. The sequence of interactions starts with an askToJoin and then an examineFusionRequest.

The *Part* class specifies a role which is part of a holon. This actor knows the other members of the holon which are the elements of the *others* set. It also knows the head of its holon, *myHead*. The *Part* role has one more satisfaction criterions than a Holon : *CS*. It may also be engaged on some of its available services. These engagements are specified by the *engagements* function.
The Head class specifies the Head role. It contains a set of holons and store available services of this holon and already allocated service.
Head
HolonicRole

\[ \text{holons} : \mathcal{P} \text{ Holon} \]
\[ \text{CS} : \mathbb{R} \]
\[ \text{availableServices}, \text{allocatedServices} : \text{Service} \to \mathbb{B} \]

\[ IS = CS + SS \]
\[ \text{leaveCondition} = \{(IS < NS) \lor (CS < 0)\} \]
\[ \text{availableServices}(s) \Rightarrow (\exists h \in \text{holons} \bullet h.\text{availableServices}(s)) \]
\[ \text{allocatedServices}(s) \Rightarrow \text{availableServices}(s) \]

\text{examineMergeRequest}
\[ \Delta \text{holons} \]
\[ h? : \mathcal{P} \text{ Holon} \]
\[ \text{holons}' = \text{holons} \]
\[ \lor \text{holons}' = \text{holons} \cup h \]

\text{sendCommand}
\[ c! : \mathcal{P} \text{ Service} \]
\[ \forall s \in c \bullet \text{allocatedService}(s) = \text{true} \land \text{send}(c) \]

\text{behavior}
\[ \text{mergeRequest/examineMergeRequest} \]
\[ \text{waitingForRequests} \]

4. Proofs
OZS semantics is based upon transition systems as defined in. It means that for each OZS specification there is an associated transition system. This transition system represents the set of possible computations the specification can produce. With such transition systems and software tools like SAL one can verify specification properties.

Among the tools proposed by SAL we have chosen the SAL model checker which enables the verification of the satisfiability of a property. The SAL model-checker proves or refutes validity of Linear Temporal Logic (LTL) formulas relatively to a transition system. To establish the satisfiability of history invariant \( H \) one must actually establish that \( \neg H \) is not valid. This technique is the simplest to use but is limited by the specification state space.

In order to prove the properties presented in this section we have generated the Transition System considering only the Holonic Roles and the automata presented
4.1. **Agent self-organization**

The first property proven with the SAL model checker may be interpreted as "if the holon’s satisfaction is not enough then it will try to merge". This property is specified as follows:

\[
\forall a : HMASAgent \bullet a.\text{is} < a.\text{ns} \Rightarrow \Diamond (\text{StandAlone} \not\in a.\text{playing})
\]

It states that for all agent of an holonic MAS, if its instant satisfaction becomes less than the necessary satisfaction, eventually this agent will not play the StandAlone role. It will try to be engaged in a super-holon and thus play either the Part or the Head role.

For a StandAlone agent which is unsatisfied by its current actions and which may not be able to reach its goals the solution within a HMAS is to find a super-holon to merge with. This super-holon will provide services and holons to cooperate with. This self-organization property ensures that an agent unable to achieve its goals alone will join a holon and thus modify the holarchy.

Since we consider only the holonic roles, if the played role is not the StandAlone role, then the holon must play either Head, Part or MultiPart. In any of those cases, the holon must belong to a super-holon and thus have merged.

4.2. **Holarchy self-organization**

The second property we have proven may be interpreted as "if the holon’s satisfaction evolves and becomes less than necessary satisfaction the system will try to reorganize". This property is specified as follows:

\[
\forall a : HMASAgent \bullet \text{currentRole} = h.\text{playing} \land \nabla h.\text{is} < h.\text{ns} \Rightarrow \Box (h.\text{playing} \neq \text{currentRole})
\]

It states that for all agents of an holonic MAS whatever role it plays, if its instant satisfaction becomes less than the necessary satisfaction, eventually this agent will change the role it is playing.

A Part or Head agent with an instant satisfaction lower then it necessary satisfaction will not stay in its currently played role. When the agent’s environment evolves the super-holon it belongs to may become unadapted. The agent has then to quit the holon and find another to join with or become StandAlone according to its goals and problem constraints. This self-organization property ensures that an agent which belongs to a super-holon which doesn’t allow it to achieve its goals will quit this super-holon and reorganize the holarchy.

The holarchy is then evolving according to agents satisfactions and environment dynamic.
4.3. Affinity safety

The third property we have proven may be interpreted as "when a StandAlone holon tries to merge with another holon positive answers must come from holons which are in its neighborhood". This property is specified as follows:

\[ \forall a \text{HMASAgent} \bullet a.\text{playing} = \text{StandAlone} \land a \text{ in (SearchingHolon)} \Rightarrow \forall \text{merge : Merging} \bullet (\text{merge.orig} = a \bullet (\text{distance}(a, \text{merge.dest}) \leq 2)) \]

It states that for all agent which is StandAlone and currently in the SearchingHolon state and for all instance of merging, the destination is lesser than 2 distance units.

The result of the merging interaction between a StandAlone agent and a Head must respect affinities between holons. Indeed, the holarchy is built upon affinity grouping. One component of the affinity for the adaptive mesh problem is the distance between holons. In order to merge holons have to be neighbors. This self-organization property ensures that the choice of the holon to merge with is coherent with the topology of the terrain and does not produce non convex meshes.

5. Conclusion

In this paper we have presented a framework for the design of Holonic MAS. This framework is based upon the roles the agent can play and the satisfactions which characterise the progression of the agent towards achievement of its goals. We have presented this framework through its formal specification using the OZS formalism. The semantics of this formalism enables the verification of properties. We have proven three pertinent properties for this framework. The first property we have proven may be interpreted as "if the holon’s satisfaction is not enough then it will try to merge". It’s an important property of such self-organized systems. Indeed, it means that if one holon is unsatisfied by its current goals achievements it will try to merge to find complementary capabilities or services.

The second property may be interpreted as "if the holon’s satisfaction evolves and becomes less than necessary satisfaction the system will try to reorganize". This property ensures that if the current holarchy doesn’t correspond to the current context it will evolve in order to find a better one.

We then show how to prove problem-dependent properties to ensure the merge of holons. This is done with the third property which is related to the affinity between merging holons and states that the distance between merging holons can’t be greater than 2 distance units. This property ensures that the system doesn’t produce non convex meshes.

Other frameworks and methodologies have been proposed\(^9,^8,^7\) and, although they have shown to be effective inside specific domains, a more generic framework is needed. Indeed, it is difficult to design a Holonic MAS without clear and specific
Formal Holonic Framework

definitions that can lead from the analysis in terms of holon to the design of the system. Moreover, a framework with predictable properties, such as those we have proven, constitutes a solid foundation for the development of Holonic MAS.

The aim of the paper is to contribute to the definition of a well founded framework for the analysis and design of Holonic MAS. We have given some elements of this framework but it needs more work to constitute a methodology for Holonic MAS.

References