Contents

Draft document for the document published in:
Self-organisation Software: from natural to artificial.
First edition. Natural computing series chapter 11, pp. 238-263.
Springer. 2011. ISBN: 978-3642173479.

Chapter 1
Holonic

Name of First Author and Name of Second Author

This a chapter skeleton for the Self-Organising Systems *textbook.*

Abstract It is recommended that chapters should follow the structure outlined below. Each chapter should be preceded by an abstract (10–15 lines long) summarising its content. The abstract will appear *online* at www.SpringerLink.com and it will be publicly available with unrestricted access.

Abstracts will not appear in the printed version of the book. Please use the 'starred' version of the new Springer `abstract` command for typesetting the text of the online abstracts (cf. source file of this chapter template `abstract`) and include them with the source files of your manuscript.

Sebastian Rodriguez
UTBM, 90010 Belfort FRANCE e-mail: sebastian.rodriguez@utbm.fr

Vincent Hilaire
UTBM, 90010 Belfort FRANCE e-mail: vincent.hilaire@utbm.fr

Nicolas Gaud
UTBM, 90010 Belfort FRANCE e-mail: nicolas.gaud@utbm.fr

Stphane Galland
UTBM, 90010 Belfort FRANCE e-mail: stephane.galland@utbm.fr

Abderrafia Koukam
UTBM, 90010 Belfort FRANCE e-mail: abder.koukam@utbm.fr

Objectives After reading this chapter the reader will:

- understand what holons and holarchies are and on what type of applications they can be successfully applied;
- know the different questions which are to be answered for engineering an holonic system
- understand a specific framework which eases the analysis, design and deployment of such systems

1.1 Introduction

Multi-Agent Systems (MAS) stand out as a paradigm for the design of Complex Systems. Indeed, this paradigm proposes new strategies for the analysis, modelling and implementation of such systems. Its elementary constituents are called ’agents’, i.e. software entities which exhibit autonomous and flexible behaviours.

Complex Systems are characterised by a large number of entities in interaction, exhibiting self-organisation features and emergent behaviours. Nobel Laureate Herbert Simon states :

> Empirically a large proportion of the complex systems we observe in nature exhibit hierarchic structure. On theoretical grounds we could expect complex systems to be hierarchies in a world in which complexity had to evolve from simplicity.

[20]

This asseveration about real-world Complex systems raises the question, if nature has selected this path, should scientists trying to model Complex Systems privilege it too?

In 1967 Arthur Koestler coined the term *holon* as an attempt to conciliate holistic and reductionist visions of the world. A holon represents a part-whole construct that can be seen as a component of a higher level system or as whole composed of other holons as substructures.

This elegant idea was recently adopted by the DAI Community under the name of holonic multi-agent system, HMAS for short [11].

The underlying idea is that a set of agents can create and compose other agents, thus generating a hierarchic system.

Even if this basic idea has been supported and put forward by many MAS researchers, most widely used models still consider agents as atomic entities. Indeed, as [10] points out, almost all the proposals for agent architectures have not addressed the general problem of how to treat collections of ”agents” as higher-order entities, for example, how to treat organisations as agents.

1.2 Background Context

1.2.1 Hierarchies and holons in MAS

Even as MAS theory evolves and matures, some open issues are yet to be answered. One of those issues is how to deal with organisations as agents. In other terms, how to represent the fact that a group of agents in interaction exhibit a specific behaviour and that, at a certain level of abstraction, they behave as if they were one single entity.

Several works have studied this question and they have proposed a number of models inspired from their experience in different domains. In many cases we find the idea of *agents composed of other agents*. Each researcher gives a specific name to this type of agent, [8] discusses *individual* and *collective* agents; *meta-agents* are proposed by [14]; [16] uses *intermediate* agents to detect earthquakes; [3] proposed *recursive* or *intermediate* agents between the reactive (lower level) and cognitive (upper level) agents; *Agentified Groups* are taken into account more recently in the work of [17]; etc. All of these are only examples of how researchers have called these ”aggregated” entities that are composed of lower level agents.

We called them *Holonic Agents*. The term *Holon* was coined by the Hungarian philosopher Arthur Koestler in 1967 while trying to explain social phenomena in human societies.

Holonic Systems grew from the need to find comprehensive construct that could help explain social phenomena. Since then, it came to be used in a wide range of domains, including Philosophy [25], Manufacturing Systems [23], and Multi-Agents Systems [11].

Holonic Multi-Agent Systems are being the subject of more and more research. However, in the research community a recurrent question comes to mind: How are holons related to agents?

Most certainly, different answers to this question can be given according to how holons are seen. But in this section we will try to define the difference (if any) between Agents and our view of Holons.

Holons are, by definition, composed of other holons, referred in this work as the super- and sub-holons, while agents are not **necessarily** composed of other agents. This does not mean that agents can not be composed of agents, but that ”agents are atomic entities” is the general assumption. Indeed, as pointed out by [10], almost all the proposals for agent architectures have not addressed the general problem of how to treat collections of “agents” as higher-order entities, for example, how to treat organisations as agents. Holonic MAS represent an attempt to tackle this problem.

1.2.2 The Social Metaphor in MAS

Already in the eighties, links between human organisations and computational systems were suggested [9]. Since then, organisational approaches have become the subject of an increasing interest in the research community.

In MAS several approaches have been proposed inspired from a Social Metaphor, where terms like ”role”, ”group”, ”community” represent the main concepts of the model. We can realise the usefulness of the concepts when we consider the number of methodologies [e.g. GAIA[26] or MESSAGE [2]] and (meta-)models [e.g. AGR [6], RIO [13] or MOCA [1]] using these concepts.

As Ferber [6, 7] points out Organisational approaches can contribute to Agent Software Engineering in the following points:

Heterogeneity of Languages : If each group is considered as an interaction space, inside each group we can find specific communication means such as KQML or ACL without modifying system-wide architectures.

Modularity : Organisations can be seen as modules that provide a description to obtain a particular behaviour of the members. We can use them to define clear visibility rules that help in the design of MAS.

Multiple Architectures : An organisational approach makes no assumptions about the internal architecture of the agent, thus leaving the specification open for a number of models and implementations

Security of Applications : If all agents communicate without any external control it may lead to security problems. If we allow, when required, each group to control the access to the roles defined in the group, we can then reach a level of security without the need of a ”global” centralised control.

By considering organisations as blueprints that can be used to define a solution to a problem, we believe that an organisational approach encourages a reusable model.

Based on these elements, we have selected RIO as organisational model for HMAS framework. In the next section we present an overview of the RIO Model of [13] and define what we understand by terms like role and organisation.

1.3 Theoretical Notions

1.3.1 A Generic Framework for Holonic Systems Modelling

A holon is a self-similar structure composed of holons as sub-structures.

This hierarchic structure composed of holons is called a *holarchy*. A holon can be seen, depending on the level of observation, either as an autonomous ”atomic” entity, or as an organisation of holons. This duality is sometimes called the *Janus*

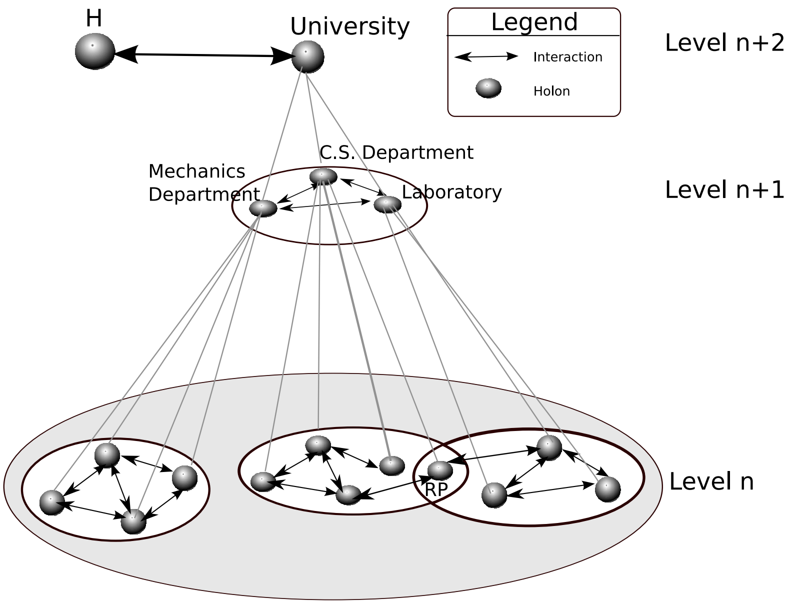

Fig. 1.1 University Example Holonic Structure

Effect[1], in reference to the two *faces of a holon*. A holon is a whole-part construct that is composed of other holons, but it is, at the same time, a component of a higher level holon.

Examples of holarchies can be found in everyday life. Probably the most widely used example is the human body. The body can not be considered as a whole in an absolute sense. It is, in fact, composed of organs, that in turn are composed of cells, molecules, etc.

Hundreds of other examples can be mentioned, enterprises, cities; even a galaxy can be seen as a holarchy.

If we consider our university as holon, we can say that it is composed of departments and laboratories. In turn, we can consider a department to be composed of the academic staff and students, while a laboratory is composed of researchers. This holarchy is depicted[2] in figure 1.1. At the highest level we find the university (level $n+2$) that, in this example, is composed of three sub-holons: Computer Science department, Mechanics departments and a laboratory (level $n+1$). At lowest level, we find the professors, students and researchers (level n).

Holonic structures offer a powerful abstraction to model Large Complex Systems. But, before we can start using holons as a modelling abstraction, we have to

---

[1] Roman god with two faces. Janus was the god of gates and doorways, custodian of the universe and god of beginnings

[2] The representation of the compositions was inspired from [8]

find means to represent and describe these entities. This description language must be generic enough to let us represent the different aspects of holons and holarchies.

1.3.2 Overview of the Approach

Our framework is based on an organisational approach to minimise the impact on the underlying architecture. However, in order to maintain this framework generic, we need to distinguish between two aspects that overlap in a holon. The first is directly related to the holonic character of the entity, i.e. a holon is composed of other holons. And the second is related to the problem the members are trying to solve. For example, let's consider the laboratory of our university. The holonic aspect makes reference to the fact that the researchers compose and manage the laboratory. We call this a *holonic aspect* since every holon, no matter the application, is always composed of other holons. On the other hand, the laboratory is created with a specific purpose and to fulfil a number of goals/tasks in the system (e.g. complete a research project). How the members organise and interact so that the super-holon can achieve its goals is specific to the application or domain of application. Even more, the members of two different super-holons may follow different interaction patterns to achieve the same result.

So, the first aspect is common to all holonic systems, while the second is directly related to the domain of application. In section 1.3.4 we describe how those holonic related aspects can be modelled. How to model application related aspects will be presented in section 1.3.8.

Our framework is concerned with the modelling and representation of three important aspects of a Holonic MAS:

Holon Structure and Management : A super-holon is an entity in its own right, but it is composed by its members. This part of the framework considers how the member organise and manage the super-holon.

Goal-Dependent Interactions : Super-holons are created with an objective and to fulfil certain tasks. To achieve these goals/tasks, the members must interact and coordinate their actions. Our framework also offers means to model these aspects of the super-holons' functioning.

Dynamics : Dynamics are inherent characteristics of MAS. The framework considers in particular two of the most attractive characteristics of Holonic MAS: Merging (Creating and Joining a super-holon) and Self-Organisation.

We will tackle each of these elements of the framework separately. First, in section 1.3.4, we will discuss how can we model a super-holon considering only the structure and management and the role the members play in these mechanisms. Section 1.3.8 presents how the goal-dependent behaviour of the members can be modelled and how elaborated coordination mechanisms can be introduced. Finally, in sections 1.3.9 and 1.3.10 we focus in the dynamic aspects of a holonic MAS.

1.3.3 Definitions and Terminology

Before we discuss the building blocks of our framework, we present in this section a number of definitions and terminology that will make it easier to discuss their properties.

At a certain level of observation, a super-holon can be seen as a set of sub-holons in interaction. These interactions shape the structure and behaviour of the super-holon. In order to model these interactions and behaviours of the sub-holons in an orderly and modular manner, we isolate them by means of organisations that assemble them according to a common context. A holon can, therefore, be modelled as:

Definition 1.1. A holon of level n, $\mathscr{H}_n$, can be modelled by a tuple:
$\mathscr{H}_n =< H_{n-1}, OP, \psi >$
Where:
H_{n-1} is the set of sub-holon members of the super-holon
OP is the set of Organisations that govern the life and functioning of the super-holon $\mathscr{H}_n$.
$\psi : H_{n-1} \mapsto 2^{roles(OP)}$: a function relating a sub-holon with the set of roles played in $\mathscr{H}_n$. The *roles* function returns the set of roles defined in the organisations of OP
So that:

$$\forall h_i \in H_{n-1}, \psi(h_i) \neq \emptyset$$

This definition is not without consequences on the way we see a holon, and, therefore, a holonic system. The main consequences of this definition can be stated as follows:

1. a set of entities can generate a new super-holon **only** if they interact.
2. a holon is not defined only by its members, but also by their pattern of interaction. The main consequence is that two different super-holons can appear from the same set of sub-holon if the interactions that influence the behaviours of the super-holons differ. In other terms, the sub-holons play different roles inside each super-holon.
3. the structure of the super-holon is defined by the social commitments and interactions of its components. This is consistent with Ferber's discussion about $individual$ and $collective$ agents [8].
4. the level of commitment of a member is defined by the obligations of the role it plays. In other words, the member's commitment towards its super-holon are defined by the obligations that such member has concerning the role it plays.

In the way we see a super-holon, its components are forced to interact in order to create the super-holon (item 1). The term *interact* is not limited, in this context, to direct messages, but also includes other types of interaction such as indirect (e.g. via the environment) or commitments that modify the super-holons state.

Item 2 states that two different super-holons may appear from the same set of holons. Imagine for instance a group of people that work together. At work, the meaningful interactions are those related to the tasks inside the company. This means that if they go out for a drink, it does not affect the behaviour or functioning of their company. Now imagine that the same group of people are the only members of the company, and they have created a sports association. The association is interested only in the actions that may change its behaviour or functioning. In other words, from the same set of members, two super-holons (company and association) have been created. Their behaviour is defined by different interactions and they can evolve separately.

When a holon takes a role inside a super-holon, it accepts to honor the behaviour associated with that role [3]. Among the elements of a role we find the requirements for taking and leaving that role. These commitments are the binding forces that keep the super-holon's body (sub-holons) together. Sub-holons are then forced to respect their engagements towards the super-holon and the structure and stability of the super-holon can be characterised by the commitments of its members (items 3 and 4).

1.3.4 Holon Structure and Management

This part of the framework concentrates on the modelling of the structure and management of the super-holon. Here we will discuss what we have called *holonic related aspects*.

The first question to clarify is how holons are organised internally to generate and manage a super-holon. Three different structures were proposed by [11] for holonic multi-agent systems (HMAS): Federation of autonomous agents, Moderated Group and Fusion.

In our approach we have adopted the moderated group structure. This decision is based on the wide range of configurations that are possible by modifying the commitments of the members toward their super-holon.

In a moderated group, we can differentiate two status for the members. First, the *moderator* or *representative*, who acts as the interface with non-member holons, and, second, *represented* members, who are masked to the outside world by their representatives.

Even if we use the name "*Moderated Group*" for compatibility with earlier works in this domain, it can be misleading. As we see it, the structure does not necessarily introduce any authority or subordination. The name makes reference to the different status found in the group. We can then adapt this organisation by giving the representatives specific authorities according to the problem or constraints.

[3] A role represents for us the abstraction of a behaviour

1.3.5 Holon's Members Modelling

In order to represent a moderated group with an organisational approach, we need to identify a set of roles that can represent these concepts. We have chosen to use four roles to describe a moderated group as an organisation: Head, Part, Multi-Part and StandAlone. The three first roles describe a status of a member inside a super-holon. The StandAlone role represents, on the other hand, how non-members are seen by an existing holon.

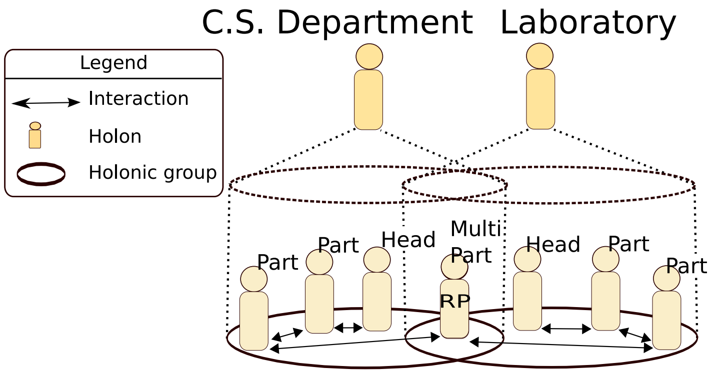

Fig. 1.2 Department and Laboratory Holons

In this section we consider an existing super-holon and study how the holonic aspect can be represented as an organisation.

Inside a super-holon, members can play three different roles: Head, Part and Multi-Part. The *Head* role is the *representative* or *moderator* of the group. Represented members can either play *Part* or *MultiPart* role. *Part* role is played by those holons belonging to only one super-holon. And *Multi-Part* role by those holons shared by more than one super-holon. We call these roles *holonic roles*. The adjective ”holonic” is used to distinguish these roles (present in every holon) from the roles used to model application-dependent behaviours.

If we isolate the Computer Science and Laboratory Holon and their components from the university example and we add these holonic roles, we obtain figure 1.2. *Part* role players for the laboratory represent researchers that belong only to the laboratory (e.g. full time researchers). On the other hand, some researchers may, in addition to their activities in the laboratory, give lectures in the computer science department. These holons, like holon *RP* in figure 1.2, belong to both holons simultaneously and thus they play the *MultiPart* role. In this example, the department and laboratory directors would be the *Heads* of the C.S. Department and the laboratory respectively.

Let us now consider only the members of one super-holon. We can see the super-holon as a set of sub-holons in interaction. This organisation, called *Holonic Organisation*, makes abstraction of application-dependent interactions to concentrate solely on the status and behaviour of the members from the super-holon point of view. An instance of this organisation is called a *Holonic Group*. As for the roles, the term ”holonic” is intended to distinguish this group (present in every holon) from other groups used to describe goal-dependent interactions. Every member of a super-holon **must** play at least one role in the holonic group. Like this, a super-holon contains at least one group that identifies the status of its members.

In the following paragraphs we detail each of these roles. The representatives of the super-holon play the *Head* role. A Head member becomes then part of the visible face of the super-holon. This means that the head becomes a kind of interface between the members of the holon and the outside world. The head role can be played by more than one member at the same time.

Thus, the *Head* role represents a privileged status in the super-holon. Heads will generally be conferred with a certain level of authority. However, these members also have an administrative load. This load can be variable depending on the selected configuration.

It is important to remark that when a set of holons merge into a super-holon a new entity appears in the system. In this case, they are not merely a group of holon in interaction as in ”traditional” MAS theory. The super-holon is then an entity of its own right. Thus, it has a set of skills, is capable of taking roles, etc. Even if heads represent the members and may have a certain power over the skills or services of the super-holon, heads are **not** the super-holon. Consider, for instance, the example of an army as a holon (composed of divisions, battalions, platoons, ...); its head would be a General with full authority over the divisions and so on. What the army is capable of achieving (e.g. secure an area) is not an ability of the general (head) but of the army itself. Evenmore, if the general is killed, it does not imply that the army will disappear.

At the same time, as *Heads* constitute the interface of the super-holon, they are in charge of redistributing the information arriving from the outside. And, thus to ”trigger” the (internal) process that will produce the desired result. We will discuss this issue further when we introduce how task-related interactions can be modeled.

The *Part* role identifies members of a single holon. These members are represented by *Head*s with the outside world.

While the holon belongs to a single super-holon, it will play this role. However, when the holon is not satisfied with its current super-holon it has two possibilities. The first is to quit its super-holon entirely and try to find a new holon to merge and collaborate with. The second is to try to merge with a second super-holon while remaining as a member of the first super-holon. In this case the holon will change his role to *Multi-Part*.

The *Multi-Part* role is an extension of the Part role. It emphasises on a particular situation when a sub-holon is shared by more than one super-holon.

Examples of this type of situation can be easily found. For instance, in our University example, holon *RP* is a researcher in the laboratory and at the same time a lecturer in the Computer Science Department (c.f. figure 1.2)

There are several reasons to differentiate between the Part and Multi-Part role. First, there are a set of problems that can arise from the fact that the shared holon is represented by more than one Head

Imagine for instance that a holon offers services that are conflicting, e.g. a mechanism to create/access/destroy a resource. If this holon is shared by several super-holon, it might be possible that it receives a contradictory request from its heads, like destroy/access the resource at the same time.

We could say that three types of conflicts can arise from a shared member:

- Interest conflicts: The super-holons do not share the same goal, or they have contradictory objectives.
- Authority Conflicts: The representatives of the super-holon request contradictory action from the shared member.
- Unbalanced Authority Conflicts: One of the super-holon's Head has more power than the other over the shared member.

Several problems can appear as a combination of these conflicts. For instance, in open systems, a self-interested head could use its authority over a shared member to avoid the progress of other holons.

Beside these problems, shared holons can be the cause of bottle-necks and performance issues.

These cases must be analysed in detail to maintain the coherence and stability of the designed system.

Even if we have only considered the possible disadvantages related to shared members, Multi-Part holons offer also a great number of interesting possibilities.

One of such possibilities is *Message Forwarding*. This consists of allowing MultiPart holons to forward messages from the members of one super-holon to members of the second super-holon. For instance, if a researcher wants to delegate a task to a student, he could ask a shared member (e.g. *RP* in figure 1.2) to look for a candidate and delegate the task. This could reduce the administrative load of head and avoid "formal requests" between the laboratory and the C.S. department.

Another possibility is to implement a *trust mechanism* to accept members introduced by shared holons.

As we have already mentioned a holarchy is a hierarchical structure of holons. We can then find Multi-Part holons in different levels of the holarchy. These shared members may produce feedback and shortcuts in the holarchy.

1.3.6 Super holon management

A super-holon is, internally, a community of holons that cooperate to achieve a commonly agreed objective or task. So, during the creation of the super-holon, certain rules have to be defined. These rules will govern the evolution of the holonic structure.

Trying to enumerate all possible issues would probably result in an incomplete or domain-dependent list. So instead of presenting an exhaustive list, we will discuss the most common and important rules that must be defined.

The first functionality that is required to manage a holon is the **Inclusion/Exclusion of members**. Once a super-holon has been created, new members may request their admission or the super-holon may require new members to achieve its goal/task. Two aspects of this functionality should be analysed. First, who makes the decisions and how (heads, vote, etc). Second, the requesting process itself. We call the inclusion process **Merging** and it will be discussed in section 1.3.9.

Another important functionality concerns the **goals/tasks of the super-holon**. For example, how does the super-holon add new goals or tasks? How does it change or modify existing ones? And who is authorised to take such a decision?

Of course, we cannot forget the **destruction of the holon**. The first, and most obvious, answer is found when considering task/goal oriented systems. However, other possibilities may be interesting like fitness, satisfactions of the members, or even, forbidding the destruction altogether.

Probably the most important functionality, and through which others can be created, is that of **adding/modifying rules**. This means that we give the holon the possibility to evolve and modify its existing configuration as it needs to. We will discuss this issue further when we present the Merging interaction.

[19] proposed the FORM Framework. The author details a number of ”parameters” that have to be defined for a holon. These parameters are intended for task-oriented systems and require further research to be adapted to our model.

1.3.7 Decision Making: Defining a Government

The different rules discussed previously require, in most cases, a decision making process. For instance, when an external holon request its admission as a member, the super-holons members can use a voting mechanism to take a decision.

This section does not intend to present a extensive discussion in decision making procedures[4], but to illustrate the range of possibilities available for a super-holon.

On one side of the spectrum we find the federation. In this configuration, all members are equal when a decision must be taken. On the other side, we find the dictatorship, where heads are omnipotent. Decisions do not have to be validated by any other member. Even more, members lose most of their autonomy by having to request permission of the head to provide a service or request a collective action.

Even if these configurations may be useful in specific domains, generally an intermediate configuration will be desirable. So, we first need to identify which func-

[4] The reader interested on the subject may refer to [24]

tionalities are needed in order to manage a super-holon; and then, define the level of authority that the Head will have over these functionalities.

This can be done by attributing to each functionality a voting mechanism. In order to parametrize a voting mechanism, three elements must be defined: **Requester**, **Participants** and **Adoption Mechanism**.

The vote requester defines which members are allowed to request for a vote. Participants makes reference to who is authorised to take part in the vote, and finally the adoption mechanism defines how a proposal is accepted or rejected.

For the requester and participants three possible option are available: all members, heads only, subgroup of holons. We will not go into detail about the adoption mechanism in this work, but we can imagine a number of options, e.g. consensus, two-thirds, etc.

Considering only the number of voters and the percentage of heads and parts involved in the decision-making process, we can distinguish four particular configurations.

Monarchy : the command is centralised in the single hands of a Head. It does not refer to the non-election of the head. We consider that the nomination process is a different issue from the decision-making process, so we also have to detail it. There is only one head who control the entire decision-making process.

Oligarchy : A little group of heads share the command without referring to the Part members.

Polyarchy [5] : A little group of heads share the command but they have to refer to the Part for certain decision.

Apanarchy [6] : The command is completely shared between all members of the super-holon. Everyone takes part in the decision-making process.

The characteristics of these government structures are summarised in table 1.1. Intuitively, we know that an Apanarchy represents a highest level of decentralisation.

If we consider the percentage of vote requesters and voters in the super-holon, we can graphically represent the decentralisation of the decision making.

The decision making of a super-holon can be highly decentralised, where heads are simply interfaces with the outside world, but with no authority over the super-holon or its members.

Name	Configuration
Monarchy	one Head, one Voter, no voting Part
Oligarchy	n Head, n Voters, no voting Part
Polyarchy	n Head, $n+k$ Voters, k voting Parts
Apanarchy	Everybody votes, All the Parts vote

Table 1.1 The characteristic forms of government

[5] We borrow the term coined by Robert A. Dahl to describe a specific type of democratic government.

[6] The name is a composition of the Greek *Apan* meaning *all* or *every* and *archein*, "to rule"

1.3.8 Goal Dependent Interaction Modelling

The previous section introduced an organisation that allows us to describe the different status of the members and how they manage the super-holon.

However, we cannot neglect the fact that this description would be incomplete if it did not include the interactions of the members concerning goal-driven actions.

In order to achieve its objective, the super-holon will often need to accomplish a number of tasks. Thus, the members need to organise internally to distribute subtasks, exchange information, etc.

These tasks are usually application dependent, and vary from holon to holon. These domain dependent organisations are called *Internal Organisations*.

We have chosen an organisational approach to describe the internal organisations. The holon's model contains a set of organisations (*OP* in the holon definition). One of these organisations is the *Holonic Organisation* defining the status of the members. The others are the organisations that define the required interactions to achieve the goal of the super-holon (Internal Organisations). In this way, the holonic non atomic agent (instantiating the model) contains:

- a unique Holonic Group, instance of the Holonic Organisation, which defines how are the members organised . All members of the (super-)holon must belong to this group.
- A set of groups, instances of the Internal Organisations, created to coordinate the interactions of the members. These groups are created based on the objectives/tasks of the members. A group may contain only a subset of the members of the super-holon.

To clarify this idea, lets consider a Department of the university. The department is modeled using two internal organisations. The first represents the Council. This organisation defines how decisions are taken and who is involved in the process.

The second represents a specific Lecture, describing the interactions between the students and their professor. A number of instances of this organisation may be present in a department at the same time.

Using this approach, the behaviour and interactions of the members can be described independently of their roles as a component of the super-holon.

The main advantages of this approach are:

- Clear separation between the *Holonic-related* (Holonic Organisation) and the domain-specific behaviours (Internal Organisations).
- Modularity in the description of the different Organisations. We can associate an organisation to each task / goal without modifying existing ones.
- It encourages a reusable modelling through the use of organisations as description units.
- This approach lets us break the intrinsic recursivity of holons in the modelling phase. The designer can describe the interactions of the members without having to take into account whether that member is an atomic holon or not.

- Complex mechanisms for task distribution, decision making, cooperation, etc. can be easily introduced into holons.

The description of a holon involves then a number of organisations. The **only mandatory** organisation is the *Holonic Organisation* that describes the member's status. Others organisations can be added to describe additional behaviours required for the functioning of the super-holon.

1.3.9 Holon Dynamics

In this section we discuss the creation of new super-holon in the system and the integration of new members into existing holons. This process is called *Merging*.

New super-holons can be created either by a set of existing holons that *merge* into a super-holon, or by decomposing a holon into subcomponents. We will not detail any further the decomposition of a holon since, in this case, the super-holon is capable of defining the intentions of its components. Thus, it controls how members will interact and even choose a specific architecture for the sub-holons. In this work, we are interested with the creation of super-holons from existing holons.

The Merging interaction is a particular interaction between two holons that want to create a new entity that assembles them.

We can distinguish two types of merging: creation of a super-holon and joining a super-holon.

In order to support the integration of new members, we need to provide external holons with a ”standard” interface so they can request their admission. From the super-holons point of view, external holons are seen as $StandAlone$ role players.

When a super-holon is created, only $Heads$ belong to the interface of the super-holon. Thus, other members (*Part* and $MultiPart$) should not be visible by external holons. This is modeled by the organisation presented in figure **??**. In this organisation, $StandAlone$ holons may interact only with the heads of the super-holon.

This organisation enables $StandAlone$ holons to interact with the representatives. In this organisation we find the *Merging* interaction that provides means for a holon to request admission as a new member.

The $StandAlone$ role represents a particular status inside a holonic system. In contrast to the roles presented previously, this role represents the way an existing super-holon sees a non-member holon.

Until now we have discussed roles that are present inside a super-holon. But a holon may also interact with other holons without necessarily creating/merging into a higher-level entity. In this context, a holon is seen as an autonomous atomic entity. This brings us to an important concept: **the different faces of a holon are independent**. This means that even if a holon is seen as a StandAlone at one level, it can be composed by substructures at another level. We use the term *Face* with correspondence to Koestler's Janus Face characteristic of holons. In this sense, one face -looking up- presents the holon as an autonomous entity and the other -looking down- as a group of sub-holons in interaction.

An interesting characteristic of this approach is that Stand-Alone presents a standardised view of non-members.

The merging process may also be used between holons to create a new entity (super-holon) in the system. In this case, all rules that will govern the life of the super-holon have to be defined. From an Engineering point of view, different approaches can be used:

Predefined : The holons were conceived so that the rules for the super-holon are predefined and known by members in advance. This approach may be useful when developing closed applications. The adaptability of these types of system will remain constrained to the anticipated cases only, and will probably prove impossible to use in large open environments.

Negotiation : The Merging process foresees a mechanism to negotiate the configuration of the super-holon. This approach allows a wider range of applications and improved adaptive capabilities. But the negotiation process may induce important overheads. A mixture of this and the previous approach could help to reduce the overhead.

Evolutive : The super-holon is created with a minimum of engagements of the members. The members can then increase their commitment toward the super-holon when they consider it useful. The minimal rules set contains only one rule: **Add new rules**. Using this rule with a voting mechanism, any new rule or modification of it can be obtained.

The FORM framework proposed by [19] describes such a method for task-oriented systems.

A Predefined mechanism can be useful for closed, rather small, systems. However, it seems improbable that such a mechanism can be used in an open untrusted environment.

The Negotiation is what we might call a ”generic” approach. However, other problems are to be considered; for instance, the communication language used in the negotiations. In addition, trying to define all rules of a super-holon may prove to be a consequent task, introducing an enormous overhead to the creation of the super-holon.

1.3.10 Self-Organisation for Holonic Multi-Agent Systems

One of the most attractive characteristics of holonic systems is the self-organisation capability. This requires holons to be able to merge with other holons according to the compatibility they have to work together.

The second component of our framework provides a generic engine that will guide the holon in their merging process. This engine is based in the roles presented in section 1.3.4.

The different roles that we have presented in the section 1.3.4 represent the status of a member inside the super-holon. Using them we can define a set of possible transitions between the roles. These transitions represent the possible evolutions of an entity inside its super-holon.

By adding conditions to these possible transitions we can provide a guide to the evolution of the holon inside its super-holon. Each condition is enclosed by square brackets. Each state in the automata represents the role the holon will play in the super-holon. An additional state has been added to represent when the holon has started the *Merging* interaction. Even if in some systems the merging interaction can be an atomic and simple interaction, in some cases the interaction can be a complex and elaborated acceptance procedure. In that case the holon may remain in the Merging state for a certain time. This situation can be found if the designer decides to use a protocol as the Contract Net Protocol [22] to recruit new members.

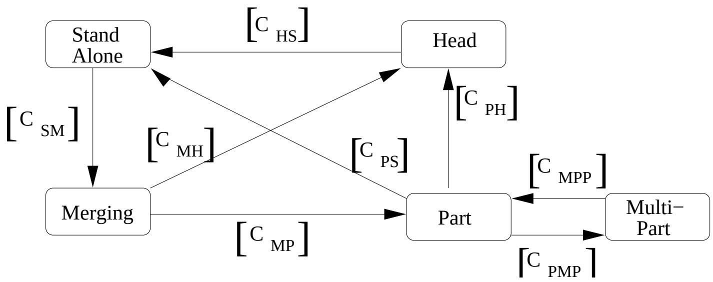

Fig. 1.3 Generic Engine for self-organisation

Figure 1.3 present this automaton and the possible transitions between the different states that we have identified.

If an application that has certain self-organising properties is desired, the automata should be refined to match the application's objective and merging criteria.

In this section we propose a specialisation of the generic engine based on the affinity and satisfaction between holons. The notion of *Affinity* was inspired by the Immune system [5]. The term *Satisfaction* has been often used to represent the gratification of an agent concerning its current state or the progress of its goals/tasks [21].

The affinity between holons must be defined according to the domain of the application. As a general rule we can define the affinity as follows:

Definition 1.2. The affinity measures, according to the application's objectives, the compatibility of two holons to work together toward a shared objective.

The *compatibility* of two holons means that they can provide help to each other to progress towards their goal. Based on the application's objective, we define a set of rules that allow us to evaluate this *compatibility*. Generally speaking, we can say that two holons are compatible if they have shared goals and complementary services.

The satisfaction, on the other hand, can be defined as:

Definition 1.3. The satisfaction measures the progress of the holon toward the accomplishment of its current goal.

In order to define the different condition present in the generic engine, we divide the satisfaction according to the actions of other agents.

Self Satisfaction (SS_i) Satisfaction for the holon *i* produced by its own work.

Collaborative Satisfaction (CS_i^H) Satisfaction produced for the holon *i* by his collaboration with other members of the Holon H. This satisfaction can be either positive, when the other members' work help *i* in its task, or negative, when the other members' work imposes barriers to the achievement of the holon's task.

Accumulative Satisfaction (AS_i) Satisfaction produced for the holon *i* by its collaboration with members of multiple super-holons. This satisfaction is only used when the Multi-Part role is allowed, i.e. holon *i* 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.

$$AS_i = \sum_p CS_i^p \ \forall p \in superholon(i) \tag{1.1}$$

where the *superholon* function returns the super-holons of *i*.

Instant Satisfaction (IS_i) Current satisfaction of holon *i*

$$\forall i \in HMAS \ IS_i = \begin{cases} CS_i + SS_i \ if\, R_i = Part \vee R_i = Head \\ AS_i + SS_i \quad if\, R_i = MultiPart \\ SS_i \quad if\, R_i = Stand - Alone \end{cases} \tag{1.2}$$

Where R_i is the role played by the holon *i*.

The conditions that will make the holon either head or part have not been replaced. Both conditions, C_{MP} and C_{MH}, should be defined by the selected merging process. The notation AS^1 and AS^2 is used to represent the AS when belonging to one or two super-holons.

In addition we define the function **Necessary Satisfaction (NS)**. This function estimates the satisfaction required for the holon to finish its task within the constraints established for the tasks. This function should be adapted to the problem under consideration.

This engine can be useful specially when the organisation of the holons into a holarchy represent the solution of the problem.

In these types of problems we do not need to specify any additional interaction among the members of a super-holon.

This engine is intended to guide the holons in their selection and merging, and thus it is not limited to applications where the holarchy provides the solution of the problem.

1.4 Applications

In order to illustrate this approach, we will take as an example the Adaptive Mesh Problem applied to the dimensioning and positioning of antennae for a radiomobile network. 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. The adaptive meshing problem for dimensioning considers communication traffic statistics as a predefined resource that must be attributed to many adaptive low power Base Transceiver Stations (BTS). This is a fundamental step in the design of a radiomobile network [4].

The developed 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 with figures 1.4, 1.4 and 1.6 which respectively show a map of the area to cover, the discretisation grid used and the resources for each cell of the grid . 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 represent the input of adaptive meshing and it is called *Resource Matrix*.

Fig. 1.4 Area to cover

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 1.4. 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

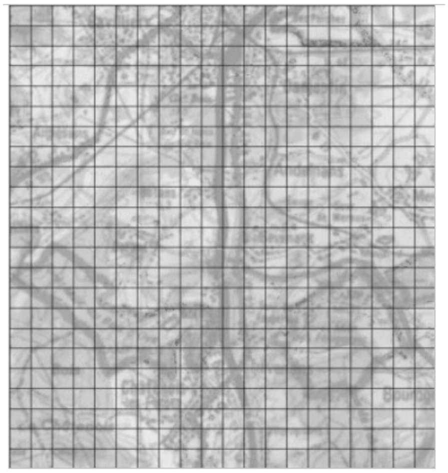

Fig. 1.5 Regular grid used to discretize the area

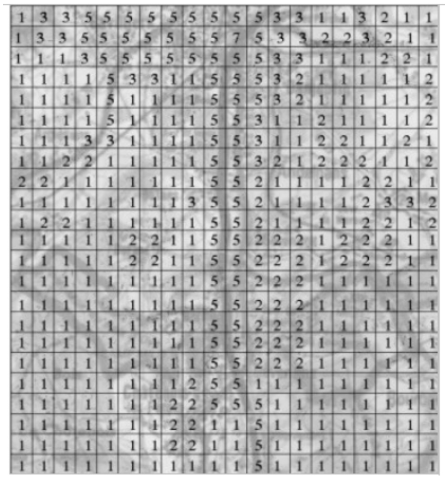

Fig. 1.6 Communication Traffic Discretization

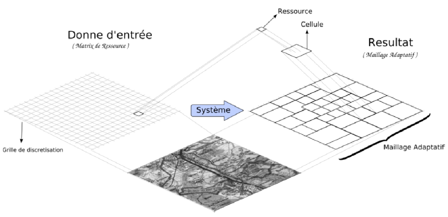

Fig. 1.7 Mesh composition

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. 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 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 reader should bear in mind that this model does not aim to provide a final solution to the adaptive mesh problem nor to the antennae positioning. We simply intend it to be an introduction and example to the application of the framework ”out of the box”. Work like [15] show the complexity of this problem and the number of characteristics to take into consideration in order to obtain a realistic approach on the subject. Nevertheless, the holonic model presented in this chapter offers all the required elements to be enriched and improved to obtain an application capable of handling the problem as in the mentioned works. It is important to notice that the solution obtained does honor the problem as stated previously.

1.4.1 Holonic Model

The objectives of the application are grouped under a single view and thus comprise only one holarchy. This view is modeled using what we have called a *Behavioural Approach* since what we know is the behaviour intended from the system.

This behavior can be summarised as: *The resources should group into cells that respect the geometrical and maximal communication constraints. These cells will finally compose the Mesh which is the solution of the problem.*

We need then to define *when* a resource needs to merge with a Cell, and, if several choices are available, *how* to choose the cell to merge with.

In order to enable this behavior in the Resource holons, we will use the automata presented in section 1.3.10. This automata is based on two main concept: *Satisfaction* and *Affinity*. We will detail how these two concepts are refined for the Adaptive Mesh Problem in sections 1.4.2 and 1.4.3 respectively.

But before we describe the automaton, let’s overview the holarchy obtained for this problem (see figure 1.4.1).

At the lowest level of the holarchy we find the *Resource* Holons. They have two attributes:

- a geographical position, expressed as (x,y) coordinates
- a communication traffic value obtained from the *Resource Matrix*

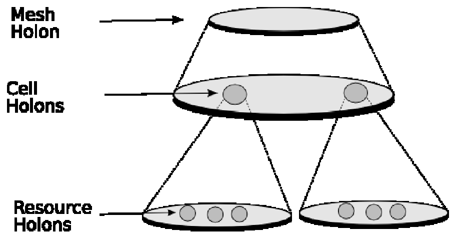

Fig. 1.8 Holarchy for the Adaptive Mesh Problem

As mentioned, the goal of the resource holons is to ensure that the communications traffic is covered by a BTS. This means that the holon must group with other resource holons into cells.

The Cell Holon will then obtain an antenna (BTS) if it respects the constraints for the problem (maximal communication and geometrical shape).

Cell holons also have two attributes. The first is the communication traffic covered by the cell. This value is the sum of the communication traffic of the members (Resource holons). The second is the shape of the cell. This shape is calculated by considering the coordinates of the members. Cell holons control that the constraints are respected and their *heads* are granted the authority to include or exclude any member to obtain a better configuration.

Since a resource must be covered by one and only one BTS, we do not allow *Multi-Part* holons. If we did, it would mean that a resource may be covered by more than one antenna. This possibility is a clear evolution of the model and will be taken into consideration in future research.

Whereas we use only one Holarchy to model the problem and *Multi-Part* holons are not allowed, we will accord us certain terminology flexibility. When we use *StandAlone* holon we understand the holon that plays the StandAlone role, and thus is not a member of any super-holon.

The top level is the mesh as a whole and, when it has achieved a stable state, the solution to the problem.

There is only one Mesh Holon in the system and all Cell holons are considered to be members. This holon is in charge of determining when the system has arrived to a stable state and thus to a solution.

1.4.2 Satisfaction

In the adaptive mesh problem, the goal of the Resource holon is to ensure the coverage of its communication traffic, then it will try to join a cell immediately.

First, we define the organisation that will allow the *Resource Holons* to merge with *Cell Holons*. This organisation is a specialisation of the *Merging Organization* (cf. section 1.3.9). We have named it *Cell Merging Organization*. Since our system is a closed system, we do not need to describe a Merging process based on negotiations. Indeed, in this application we use a Predefined Merging Process. In other words, the holons know in advance how to request its admission into a super-holon.

The satisfaction will define when the *Merging* Interaction is triggered and the Affinity will help the *CellResponsable* select a candidate.

In the Adaptive Mesh Problem we will use communication traffic to define the satisfaction of the Resource Holon. We present now how the satisfactions introduced in section 1.3.10 are adapted to our problem:

Self Satisfaction (SS_i) Satisfaction for the holon i produced by its own work. In our problem, this represents the communication traffic of each Resource holon.

Collaborative Satisfaction (CS_i^H) Satisfaction produced for the holon i by its collaboration with other members of the Holon H. In the meshing problem, this satisfaction is calculated using the communication traffic of the other members of the Cell Holon.

Accumulative Satisfaction (AS_i) Satisfaction produced for the holon i by its collaboration with members of multiple holons. This satisfaction is only used when MultiPart holons are allowed, thus not useful in the meshing problem.

Instant Satisfaction (IS_i) Current satisfaction of holon i. This satisfaction is calculated according to the roles the holon plays:

$$\forall i \in HMAS\ IS_i = \begin{cases} CS_i + SS_i & if\, R_i = Part \vee R_i = Head \\ SS_i & if\, R_i = Stand - Alone \end{cases} \tag{1.3}$$

Where R_i is the role played by the holon i.

In the automaton, we finally define the *Necessary Satisfaction* (NS) as the instant satisfaction required to accomplish the objective. In our problem, we define the necessary satisfaction as the maximal communication traffic a BTS can cover. Like this, we do not only require a resource holon to become a part of a cell, but also that the cell should be as close as possible to the BTS's maximum capability.

If we look at equation 1.3 we see that $IS = SS$, when the holon plays the StandAlone Role. Then if the resource holon's traffic is not enough to receive an antenna of its own, it will try to merge ($IS < NS$).

When a group of Resource holons merge into a Cell Holon, one of them is designated as the *Head* of the Cell. A number of approaches can be used to appoint the super-holon's head. In order to avoid unnecessary communications among members, and since we are operating in a closed system, we have preconditioned certain Resource holons to take the Head role. In fact, the resource holon that is the closest to the center of the cell takes the head role. In the adaptive mesh problem, the holon that plays the head role guaranties that the Cell holon respects the constraints. In order to do this, it is granted the authority to accept or refuse other holon's requests to merge according to the constraints. Furthermore, it can exclude any member of the Cell holon if it can improve the satisfaction of the super-holon.

A super-holon's Head can decide to destroy the super-holon if, after trying to improve the super-holon's satisfaction, the satisfaction is insufficient to obtain a BTS. Such a situation may arrive when the communication traffic covered by the

Cell is far below the maximum of an antenna and all surrounding resources are already covered by another cell.

In order to improve the super-holon's satisfaction, the head will accept new holons to increase the super-holon's covered resource. It may also command member holons to leave the super-holon if they don't respect the geometrical constraints or if the covered resource has exceeded the maximum.

Any Resource Holon may request its admission into an existing Cell holon. If accepted, it becomes a part of the Cell holon[7]. It will remain in the holon if its satisfaction level is raising. In order to calculate this value, we consider the Collaborative Satisfaction (CS^H) as the sum of the resources of the members of the Cell holon H. If the holon remains in the Cell holon, its goal will be accomplished and will stay in the Cell until the antenna is finally assigned.

However, two possible situations may force the resource holon to leave its super-holon:

Command to Leave : If the head of the super-holon considers that the resource holon harms the satisfaction of the super-holon, it can command the resource holon to leave. In that case, it is forced to leave and to restart the merging process.

Satisfaction decreases : Any member of a Cell holon, including the head, can decide to leave its current Cell at any time. This is usually the case when the cell's covered resource stalled below the required minimum.

In the merging process one of the most important elements is to select the super-holon to merge with. In this task, the affinity measures the "compatibility" of two holons. In the next section we describe how this value is obtained.

1.4.3 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.

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. A holon can find out the affinity it has with another by comparing its identifier with the identifier of the other holons. Then, it uses this affinity to decide whether or not to merge.

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 a holon can determine whether or not to merge.

The affinity should give a measure of the compatibility of the holon's goals 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:

[7] When a holon's request is accepted or refused will be discussed in the next section

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 1.9).

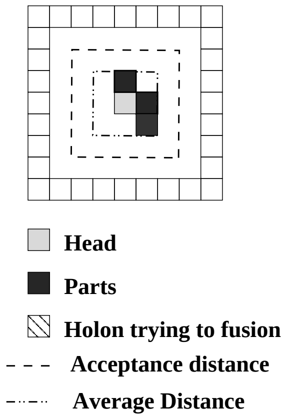

Fig. 1.9 Acceptance Distance

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 that have an unitary distance with the resource the new holon represents.

Lets consider three different situations shown in Figure 1.10. 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.

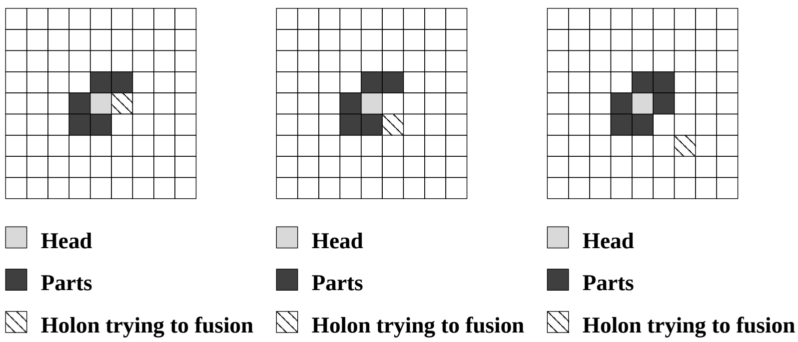

Fig. 1.10 Distance Affinity Examples.

Resource affinity : Used to ensure that the limits of an antenna are not exceeded. As a BTS is positioned inside a Cell, the communication traffic covered by a cell should not exceed the capacity of an antenna.

The affinity is used by both holons involved in the merging process and it does not necessarily give the same value. The head of the cell holon will constantly re-evaluate the affinity of the super-holon with its members. In this way, if a new request arrives, the head may decide to exclude a member in order to accept a different one that will increase the global satisfaction.

Using both mechanisms, affinity and satisfaction, the resource holons group into cells. Each cell has a head that will ensure that the constraints are respected. One of those constraints is the communication traffic that the cell must cover to obtain an antenna. The cell is said to be ”stable” when the covered resource is within the limits of the resource per cell and the geometrical constraint respected. In this case, all new requests are refused and thus the members of the cell are stable in time.

The Mesh holon, highest level of the holarchy, will constantly examine the Cell’s state. If all cells are stable, the holarchy’s structure is used to obtain the solution.

1.5 Conclusion

In this chapter we have presented the concepts underlying holonic systems and a framework which allows the engineering of such systems. Our framework is concerned with the modelling and representation of three important aspects of a Holonic MAS:

Holon Structure and Management : A super-holon is an entity in its own right, but it is composed of its members. This part of the framework considers how the members organise and manage the super-holon.

Goal-Dependent Interactions : Super-holons are created with an objective and to fulfil certain tasks. To achieve these goals/tasks, the members must interact and coordinate their actions. Our framework also offers means to model these aspects of the super-holons functioning.

Dynamics : Dynamics are inherent characteristics of MAS. The framework considers in particular two of the most attractive characteristics of Holonic MAS: Merging (Creating and Joining a super-holon) and Self-Organization.

This framework is illustrated on an application, namely the adaptive meshing problem. The solution to this problem is given by the structure of the holarchy itself. Holons self-organise following a given mechanism and the stable holarchy is supposed to be a good solution to the problem at hand. Other mechanisms can be used as it is the case in the chapter **??**.

Obviously the use of holon is not well fitted for every problem. It can be naturally applied to systems where there exist a sort of, even loose, hierarchy. Indeed it would be very difficult to apply such concepts to flat systems.

Problems - Exercises

1.1. The problem consists in designing a simulator of pedestrian movements. This simulator deals with large numbers of pedestrian in a known environment. Each pedestrian has a goal which is to reach a point in the environment. This goal may change even it is not satisfied in other words the pedestrian may decide to go to another point. This decision may be randomly made. The holonic paradigm will be used to realise this simulator. The most basic, non composed, holons will obviously be pedestrians. Composed holons can be based upon group of pedestrians which share a same direction or destination point.
Questions

1. Propose a simple model of the environment
2. Propose an holarchy for this problem.
3. Define an affinity function based upon the distance between holons and respective destination points.
4. Define satisfaction criterion based upon the holon goal satisfaction.

A given problem or Excercise is described here. The problem is described here. The problem is described here.

1.2. The following problem is extracted from a benchmark for self-organisation techniques discussed within the Self-Organisation Special Interest Group of AgentLink III. Consider a set of applications (processes) processing on an open network of workstations, while dynamically satisying a set of constraints such as:

- dynamic load balancing of processors;
- minimizing communications costs between highly communicating applications;
- optimal sharing of resources and data between process belonging to a same application;
- mutual exclusion between applications accessing a same set of resources.

The openess of the environment implies that during the processing, new workstations (respectively existing workstations) can join (respectively leave) the network. New applications (respectively existing applications) could also be launched (respectively resume their activity). The network could also be subject to perturbations such like workstations breakdowns for example.
Others aspects

Differents aspects could be considered in this benchmark, dynamic routing of processors, network topology evolution (specificities of the problem with respect to some kind of topologies: simplified problem or a more complex problem, etc), prioritizing some constraints satisfaction (different weights accorded to different constraints), etc.
Scenario

Data files on : network topology, applications caracterization , relations between applications , relations to resources access, etc. can be made available. Breakdowns are generated along stochastic distribution including correlated occurrences of disturbances. Operation of the system is described above.

Questions

1. Propose an holarchy for this problem.
2. For each holon type define affinity functions and satisfaction criterion.
3. Analyse the consequences of affinities and satisfaction criterion on the relationships between holons and holonic roles.

References

1. Matthieu Amiguet. *MOCA: un modele componentiel dynamique pour les systemes multi-agents organisationnels*. PhD thesis, Université de Neuchatel, 2003.
2. Giovanni Caire, Wim Coulier, Francisco J. Garijo, Jorge Gomez, Juan Pavón, Francisco Leal, Paulo Chainho, Paul E. Kearney, Jamie Stark, Richard Evans, and Philippe Massonet. Agent oriented analysis using message/uml. In Michael Wooldridge, Gerhard Weiß, and Paolo Ciancarini, editors, *Agent-Oriented Software Engineering II, Second International Workshop, AOSE 2001, Montreal, Canada, May 29, 2001, Revised Papers and Invited Contributions*, volume 2222 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2001.
3. Kelly Christine Correa e Silva Fernandes. *Systémes Multi-Agents Hybrides: Une Approche pour la Conception de Systémes Complexes*. PhD thesis, Université Joseph Fourier- Grenoble 1, 2001.
4. J.-C. Creput, A. Koukam, T. Lissajoux, and A. Caminada. Automatic mesh generation for mobile network dimensioning using evolutionary approach. *IEEE Transactions on Evolutionary Computation*, 9(1):18–30, Feb 2005.
5. Dipankar Dasgupta. *Artificial Immune Systems and Their Applications*. Springer-Verlag, November 1998.
6. Jacques Ferber and Olivier Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In Y. Demazeau, E. Durfee, and N.R. Jennings, editors, *ICMAS'98*, july 1998.
7. Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations: an organizational view of multi-agent systems. In *Agent-Oriented Software Engineering IV 4th International Workshop, (AOSE-2003@AAMAS 2003)*, volume 2935 of *LNCS*, pages 214–230, Melbourne, Australia, July 2003.
8. Jaques Ferber. *Les Systémes Multi-Agents: Vers une Intelligence Collective*. InterEditions, 1995.
9. Mark S. Fox. An organizational view of distributed systems. *IEEE Trans. on System, Man, and Cybernetics*, SMC-11(1):70–80, January 1981.
10. Les Gasser. Boundaries, identity, and aggregation : plurality issues in multiagent systems. *SIGOIS Bull.*, 13(3):13, 1992.
11. Christian Gerber, Jörg Siekmann, and Gero Vierke. Holonic multi-agent systems. Research Report RR-99-03, DFKI, 1999.
12. Olivier Gutknecht and Jacques Ferber. The MADKIT agent platform architecture. In *Agents Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55, 2000.
13. Vincent Hilaire, Abder Koukam, Juan Pablo Gruer, and Jean-Pierre Müller. Formal specification and prototyping of multi-agent systems. In Andrea Omicini, Robert Tolksdorf, and Franco Zambonelli, editors, *Engineering Societies in the Agents' World*, number 1972 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2000.
14. John H. Holland. *Hidden order: how adaptation builds complexity*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1995.

15. T. Lissajoux, A. Koukam, D. Renaud, A. Caminada, and J.C. Créput. Evolutionary Meshing for Mobile Network Dimensioning. *Evolutionary Meshing for Mobile Network Dimensioning,*, 10 2000. La Rochelle, France.
16. Pierre Marcenac and Stéphane Calderoni. Self-organisation in agent-based simulation. In Walter Van de Velde Magnus Broman and Staffan Hägg, editors, *Poster Proceedings of the 8th European Workshop of Modelling Autonomous Agents in a MultiAgent World*, pages 116–131, Ronneby, Sweden, May 1997. Springer Verlag.
17. James Odell, Marian Nodine, and Renato Levy. A metamodel for agents, roles, and groups. In James Odell, P. Giorgini, and Jorg Muller, editors, *Agent-Oriented Software Engineering (AOSE) IV*, Lecture Notes on Computer Science. Springer, 2005.
18. H. Van Dyke Parunak and J. Odell. Representing social structures in uml. In M. Wooldridge, G. Weiss, and P. Ciancarini, editors, *Agent-Oriented Software Engineering II*, volume 2222 of *Lecture Notes on Computer Science*, pages 1–16. Springer-Verlag, 2002.
19. M. Schillo. *Multiagent Robustness: Autonomy vs. Organisation.* PhD thesis, Department of Computer Science, Universitat des Saarlandes, 2004.
20. Herbert A. Simon. *The Science of Artificial*. MIT Press, Cambridge, Massachusetts, 3rd edition, 1996.
21. O. Simonin and J. Ferber. Modélisation des satisfactions personnelle et interactive d'agents situés coopératifs. In *JFIADSMA'01: 9èmes Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, pages 215–226, 2001.
22. R. G. Smith. The contract net protocol: high-level communication and control in a distributed problem solver. *Distributed Artificial Intelligence*, pages 357–366, 1988.
23. E.H. van Leeuwen and D. Norrie. Holons and holarchies [intelligent manufacturing systems]. *Manufacturing Engineer*, 76(2):86 – 88, April 1997.
24. Gerhard Weiss, editor. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, Cambridge, MA, USA, 1999.
25. Ken Wilber. *Sex, Ecology, Spirituality*. Shambhala, 1995.
26. F. Zambonelli, N. Jennings, and M. Wooldridge. Developing multiagent systems: the gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3), 2003.

Appendix A
Chapter Heading

Use the template *appendix.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style appendix of your book in the Springer layout.

A.1 Section Heading

Instead of simply listing headings of different levels we recommend to let every heading be followed by at least a short passage of text. Furtheron please use the LaTeX automatism for all your cross-references and citations.

A.1.1 Subsection Heading

Instead of simply listing headings of different levels we recommend to let every heading be followed by at least a short passage of text. Furtheron please use the LaTeX automatism for all your cross-references and citations as has already been described in Sect. A.1.

For multiline equations we recommend to use the `eqnarray` environment.

$$\begin{aligned} \mathbf{a} \times \mathbf{b} &= \mathbf{c} \\ \mathbf{a} \times \mathbf{b} &= \mathbf{c} \qquad (A.1) \end{aligned}$$

A.1.1.1 Subsubsection Heading

Instead of simply listing headings of different levels we recommend to let every heading be followed by at least a short passage of text. Furtheron please use the

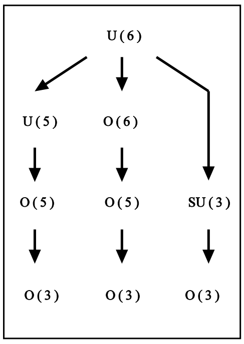

Fig. A.1 Please write your figure caption here

LaTeX automatism for all your cross-references and citations as has already been described in Sect. A.1.1.

Please note that the first line of text that follows a heading is not indented, whereas the first lines of all subsequent paragraphs are.

Table A.1 Please write your table caption here

Classes	Subclass	Length	Action Mechanism
Translation	mRNA[a]	22 (19–25)	Translation repression, mRNA cleavage
Translation	mRNA cleavage	21	mRNA cleavage
Translation	mRNA	21–22	mRNA cleavage
Translation	mRNA	24–26	Histone and DNA Modification

[a] Table foot note (with superscript)

Glossary

Use the template *glossary.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your glossary in the Springer layout.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

Acronyms

Use the template *acronym.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your list(s) of abbreviations or symbols in the Springer layout.

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced `description` environment.

ABC	Spelled-out abbreviation and definition
BABI	Spelled-out abbreviation and definition
CABR	Spelled-out abbreviation and definition

Solutions

Problems of Chapter 1

Answers to questions of first chapter go here.

1.1 The solution is revealed here.

1.2 Problem Heading
(a) The solution of first part is revealed here.
(b) The solution of second part is revealed here.