Dynamic Pyramidal Structure with Associations for Multi-agent Coordination
Structure pyramidale et associations pour la coordination multi-agents

Aurélien Slodzian Samir Aknine
LIP6 Université Paris 6
8 rue du Capitaine Scott
75015 Paris

aurelien.slodzian@lip6.fr
samir.aknine@lip6.fr

Résumé

Cet article présente un nouveau modèle de coordination multi-agent, qui combine une structure de coordination en "pyramide" avec des associations inter-agents, deux modèles dont nous montrerons l'utilité. Nous présentons aussi l'application de ces modèles de coordination à la problématique de la recherche d'informations multi-critères.

Mots Clef

Coordination multi-agents, recherche d'information

Abstract

In this paper, we present a new model of multi-agent coordination. This model combines a pyramidal coordination structure with agent associations, two notions which we introduce herein and of which we demonstrate the relevance. We present the application of our coordination model to an information retrieval project.

Keywords

Multi-agent coordination, information retrieval

1 Introduction

This article presents an original multi-agent coordination scenario, which has the main property of allowing for an efficient organisation of the computer resources and without relying on central control nor on plan sharing. This feature is fundamental for applications where no sufficiently optimal planning algorithms are available, which is the case in our information filtering application.
The SAFIR project[1] was initiated at the LIP6 as an attempt to overcome two important issues raised by information retrieval engines:

Lack of precision The search engine retrieves irrelevant documents, subsequent to the occurrence of homonyms of some query terms or to the usage of poorly discriminating terms.

Poor recall The engine does not find relevant documents because, for example, they do not contain exact matches of the query terms but rather variants like synonyms, declinations, etc.

All along this document we will take as example a simple query like "gas pollutants emitted by gas turbines in cogeneration power plants". On the one hand, isolated occurrences of "gas" and "turbine" are not really relevant to this query. On the other hand, documents talking about "trigeneration", which is a form of cogeneration, will not be brought back by search engines.
These issues were addressed in this project by improving two existing information retrieval techniques – query expansion and content based filtering – and by experimenting novel ways to combine them.

Query expansion consists in improving the user's query before submitting it to classical search engines. On the one hand, variants of query terms – like synonyms or plural – are added with the aim to collect more documents. On the other hand, domain specific terms are added to eliminate off-topic documents.

Content based filtering aims at re-sorting the documents found by the classical engines by means of a deep, multi-criteria, analysis of their content. For this analysis, we rely on computational linguistic techniques like terminological databases, derivational morphology, part-of-speech tagging, etc.

We would like to stress that the usage of these techniques imposes a preliminary analysis of the type of documents which will be filtered so as to build up the necessary linguistic resources. We thus had to focus on one application domain, which we did with the project partner EDF[2]. The

[1]Project funded by the French Ministry of Industry, `www.projet-safir.org`. The consortium is also composed of the Institut National des Langues et Civilisations Orientales and of the EDF and the Xerox companies.

[2]Electricité de France is the principal electricity producer in France

role of EDF was to constitute a corpus of documents and to provide validated terminological databases.
Early experiments in the context of the Safir project have shown that there is no optimal algorithm for combining the criteria for content based filtering. Hence came the idea of comparing filtering criteria combination to multi-agent cooperation. By associating one criterion with one agent, criteria combination comes down to agent coordination. Furthermore, the absence of a static algorithm imposes that the candidate coordination models are dynamic ones.
We propose here a coordination model that fits these requirements. It is a dynamic pyramidal coordination structure combined with multi-agent associations. Both notions are described in section 3.3 and the latter will be compared with the notions of coalition [19], team [18, 17] and congregation [3, 2, 1]. Shortly stated, the main coordination structure imposes constraints on resource usage and places agents in concurrence, forcing them to choose optimal solutions for document filtering. The associations between agents allow them to set up temporary interest groups for combining their computation power.
The model uses messages passing through these associations of agents so as to improve their utility functions, which have in turn a great impact on the coordination structure.
However, before going into the depth of the coordination model in Section 3, we will formalise, in Section 2, the problem of information filtering and describe the precision improvement criteria. We will also provide some implementation information in Section 4.

2 Formalisation of the problem

2.1 Queries, recall and precision

When a user submits a query, his goal may be modelled as a utility function which measures the relevance of the retrieved documents. A given document will then be considered as a correct answer if its relevance is over a certain predefined constant threshold.
In practise however, the utility function is not known and the information retrieval systems will attempt to order the document set using an approximation of it. The "quality" of this approximation may be measured by computing the recall (the percentage of relevant documents in relation to the number of documents retrieved) and the precision (ratio of the total relevant documents in the Internet retrieved by the search engine) of the search engine with respect to the query [16].
As already mentioned, the SAFIR project aims at improving the recall and the precision firstly by modifying the query before submitting it and secondly by re-filtering and re-ordering the search engine results. In this document we will focus on the second aspect.
One way to raise the precision is to minimise the difference between the utility function and its approximation. For this, we will consider several utility functions C_i instead of a single one. Each of these functions corresponds to a different way to evaluate the relevance of a document. We call them *criteria*.
In practise there are many criteria, some of them being correlated, complementary or partially redundant. Furthermore, there is no query independent formula that combines them in an optimal and relevant way.
We reconstruct a global utility function from the partial criteria, taking into account that the parameters of this reconstruction depend on each user's query:

$$C = \frac{1}{n} \sum_{i=1}^{n} \lambda_i(q) C_i$$

The main difficulty is to discover the "right" functions λ_i.

2.2 Criteria for information filtering

Constructing vector spaces. Let us now inspect some criteria for the evaluation of documents. This section does not claim to be exhaustive, not even to cover all evaluation criteria implemented in the Safir project. We have preferred, to avoid dispersion of the examples, to limit our presentation to terminological criteria. Nevertheless, the provided examples will sufficiently support our assertions on the effective complexity of the system. In particular, they will back up the conclusions that we will draw in the following section on the need for a multi-agent approach and on the type of coordination which will be necessary for the agents.
Basically, terminological criteria are founded on the idea of imposing a vector-like structure on the set of documents to evaluate.
Generally a text is considered as a stream of characters and basic retrieval techniques involve the search for exact occurrences of query string in the documents. One step forward consists in grouping characters in words, or *tokens*, and to consider documents as array of tokens[3].
However, an array of numbers would be necessary, instead of an array of strings, in order to construct a mathematically sound, measurable, vector space. A classical way to reach this point is to define a space were each axis corresponds to one word of some reference dictionary [11]. A document will then be associated with a vector such that each coordinate is, for example, the number of occurrences of the corresponding word in the document.
The sentence "gas pollutants emitted by gas turbines in cogeneration power plants" might then be represented by the vector: $(x_{gas} = 2, x_{pollutants} = 1, x_{the} = 0, x_{they} = 0 \ldots)$. Obviously, most coordinates are equal to zero.
In such a space, the proximity of two documents may be measured by the angle between the two corresponding vectors. In particular, if one considers the user's query as a very small document, it is also possible to measure the distance between the query and a document with this technique.

[3]The way to separate tokens may vary from one language to another, and there are specific linguistic software pieces, called "tokenizers" that perform this operation.

However, this construction still demands that the semantics of the coordinates is defined and this requires to decide on (1) the terminological nature of the vector space axes (tokens, terms, etc.), (2) the semantics of the vector coordinates (occurrences, frequencies, etc.) and (3) the measure used to compute "distances" in the vector space.
In addition, one should also consider variants in what is to be compared: documents versus queries or documents versus reference documents.
All these criteria may of course be combined together, and with other types of criteria, like meta-tag analysis, etc.

Possible natures of the vector spaces.

Tokens, lemmas, terms The documents may be considered at several levels of linguistic abstraction [8]:

Syntactical level The document is considered as a series of tokens, i.e. of raw words.

Example: gas, pollutants, emitted, by, gas, etc.

Lexical level Tokens may in turn be replaced by their base form or *lemma*. A lemma is a normalised form that represent a set of words which are different in flexions such as case, gender and number, or verbs conjugation forms [6]. This operation may be performed by a piece of linguistic software called a part-of-speech tagger.

Example: gas (NOUN), pollutant (NOUN), emit (PARTPAST), by (PREP), gas (NOUN), etc.

Terminological level Some words may be grouped together in noun phrases[4]. A *term* is a collection of such expressions, grouped together because they derive from one another by morphological transformations like flexions of case and gender, verb conjugation, acronyms, order of the elements (as in "electricity production" versus "production of electricity").

Example: gas pollutants, gas turbine, cogeneration power plant.

Starting as an array of characters, our documents became successively arrays of tokens, of lemmas and, finally, of terms. From the vector-space perspective, the same document may be considered as a vector, the coordinates of which count occurrences of either tokens, lemmas or terms.

Relations and terminological quotient spaces However, this stream of transformations is not complete yet. Indeed, the idea to switch to the higher level of terminological abstraction was motivated in the Safir project by the aim to make use of the lexical and semantical relations which exist between terms, and which are of two kinds: linguistic relations like synonymy, hyponymy[5] and meronymy[6] on the one hand and domain specific relations like "use", "produce" on the other hand.
Each of these relations may be used to impose a structure on the set of terms and to define corresponding quotient-sets, either by switching to equivalency classes (e.g. for synonymy) or to transitive classes (e.g. for hyponymy).
For example, the terms "cogeneration" and "combined cycle" might be identified in the quotient set defined by the synonymy. Similarly, "gas turbine", "steam turbine" and "turbine" belong all to the same class in the quotient space induced by the hyponymy relation.
Hence, the number of dimensions of the vector space may be reduced using the following algebraic transformation.

Projection Frequent words (stop-words, articles, prepositions, etc.) may be removed and the corresponding dimensions in the vector-space disappear.

Quotient One may consider the quotient space with respect to some relationship between the coordinates. Things appear as if the dimensions associated to some related words were merged. One practical advantage in our example is that occurrences of "trigeneration" would transparently count as occurrences of "cogeneration".

So, we end up with as many possible definitions of the terminological nature of the vector space dimensions as there are combinations of these transformations.

Possible semantics of the coordinates. Once a reference set has been selected, the next step is to define the quantification of the coordinates of the vector that represent a document.

Straight occurrence count The coordinates may represent the number of occurrences of the items in the document: $(x_{gas} = 2, x_{in} = 1, \ldots)$

Frequency The coordinates may be the frequencies of the items in the document: $(x_{gas} = 0.2, x_{in} = 0.1, \ldots)$

Weighted frequency Before computing the frequencies, the occurrences of each term are multiplied by a factor which depends on the context (titles, headings, lists, strong text, anchors and plain text [14, 7]).

Distinct occurrence count In order to compensate the bias introduced by documents with many occurrences of one single term of the query, one will use boolean coordinates which will denote the presence or the absence of a given word: $(x_{gas} = 1, x_{in} = 1, x_{the} = 0, \ldots)$

Inverted documentary frequency If one wants to reduce the weight of frequent words, which are not very discriminating, one may use the Tf*Idf coordinates system where the frequency of an item in a document is divided by the frequency of the item in all the documents [15]. Common word will hence have very low coordinates.

[4] Noun phrases are parts of sentences that play the role of a noun in the sentence. For example "production of electricity" in the sentence "The production of electricity has grown...".

[5] Hyponymy is the relation between two lexical units in which the meaning of the first is included in that of the second [13]. It may be compared to the type/subtype relations.

[6] Meronymy is the "part-of" relations.

When terminological quotients are used, several variants of the above basic quantification techniques may be used.

- Occurrences of multi-word terms might have more weight than occurrences of single word terms;
- Occurrences of synonyms (or any other related term) may be counted as occurrences of the original term, but with a lower weight.

Measuring the similarity. A given quantification system over a given reference dictionary defines a vector space where each document may be represented as a vector. Each such vector space may then be measured in a variety of ways, called "similarity measures", among which are the cosine, the minimal loss of mutual information and other information theoretic similarity measures [4, 9, 12, 10].
For example, the cosine calculus provides a measure of the "angle" between to documents. It may be easily computed using the scalar product of vectors. It has the advantage over distance calculus that the length of the documents does not introduce a bias and, consequently, two documents will be considered similar if they share the same frequencies of words, lemmas or tokens. In particular this allows for a meaningful comparison of the user's query with the retrieved documents.

Final construction. In the previous sections, we have presented several ways to construct a measurable vector space, by combining an axis selection, with various semantics of the coordinates and the choice of a measure. By combining only the mentioned elements, we end up with about forty ways to compute a "similarity" of documents.
However there are still several possibilities to apply these techniques.

Query relevance The similarity of the whole documents may be measured with respect to the user's query. Alternatively, long documents may be cut in paragraphs beforehand so as to isolated interesting sections.

Domain relevance The similarity of the document with documents from a corpus of representative documents from the domain may be measured so as to verify that a document is not off-topic. Note that other statistical methods are also applicable for this goal.

Sub-domain identification The domain of electricity production has been divided into four sub-domains covering the technical, legal, ecological and economical aspects of the subject. Four sets of representative documents have been collected which are used to guess the subject of the documents. For this aim, vector calculus is combined with other statistical techniques.

Group detection Documents collected by search engines may also be compared with one another in order to group them by similarity. This is done by measuring the average angle and by grouping together the documents the angles of which are a fraction of this average. Similar documents tend to share the same relevance.

Each combination of the above elements defines a utility function, which have called *criteria* in Section 1 and denoted by C_i. The computation of these criteria involves several linguistic and terminological techniques the description of which is out of the scope of this document. The corpora and the terminological reference data are stored in a database the structure of which has been designed for this project. The computation of each criterion may require more or less computing power, raising critically efficiency issues.
In the next section we will see how the diversity of criteria calls for a multi-agent approach to the issue of their combination in order to combine them in a meaningful and efficient way.

3 Multi-agent modelling of Safir

In this section, we will present the Safir multi-agent system and the coordination model that we propose for the agents and we will give the algorithmic behaviours of the agents during the coordination process. Then we will define more formally the concept of association in multi-agent systems and we will compare it with the concepts of coalitions, teams and aggregations.

3.1 Requirements and constraints

There are two main reasons why the SAFIR project is a non-trivial application which intrinsically requires a multi-agent approach.
Firstly, the various complexity factors combine with one another: the variety of user query types, the number of filtering and sorting criteria (several tens), their complexity and their indirect correlation and relative relevance.
Secondly, the empirical factor has an important role: some criteria that seem conceptually close may have very different results, discriminating power, efficiency or computation speed, etc. Each criterion may be used either for filtering or sorting, with diverse quality. Some have side effects, like computing some information about the documents that might be useful for the computation of other criteria. Finally, the multiple possible combination of criteria may be more or less precise and efficient. But we do not possess any reliable means to determine in advance the precision or the efficiency of such or such combination with respect to a given query.
Here are the obstacles to the formalisation of rules that might guide the global filtering process in a deterministic way.
For example, newspapers articles tend to use less technical and less varied vocabulary than scientific ones. This calls for binary coordinates, but the system has very few clues for guessing the nature of the retrieved documents. Hence whether a binary space based criterion should be used will depend not only on the documents, but also on previously

retrieved documents of the same author, or from the same site.
In these issues lies the origin of the idea to relate the combination of the filtering criteria with the cooperation between software agents. The key of this comparison resides in the pairing of a criterion and an agent, the combination of criteria being solved in a cooperation between agents. But it must be as of now clear that it is not enough to encapsulate the criteria in agents to solve the problem, since the absence of algorithm implies also the inapplicability of those protocols which define the roles of the agents in a static way. One needs a dynamic coordination of the agents.

3.2 The Safir multi-agent architecture

We consider herein that an agent is an autonomous entity which interacts with others through protocols. With regard to the filtering of the documents, three kinds of agents were defined.

- The *criteria-agents* C_j encapsulate a criterion of evaluation C_j, like those presented in section 2.2. The service that they are likely to provide consists in evaluating and grading the documents which are presented to them. They all are thus different from one another.
- The *document-agents* A_j are associated to the documents brought back by the search engines. Such an agent is associated to each document and its role is to find criteria to evaluate it. The document-agents are thus, a priori, identical between them and their lifespan is limited.
- The *query-agents* Q_i are associated with the queries of the users. They are those to which document-agents and criteria-agents must give their final evaluation.

The founding principle of the so-called "pyramidal" cooperation model is to carry out the application of the criteria in several passes (Figure 1). Thus, at a moment t, the evaluation of a document d with respect to a query q is defined as:

$$C(d,t) = \frac{1}{n}\sum_{i=1}^{n} \lambda_i(q,t) C_i(q,d)$$

The parameter $\lambda_i(q,t)$ measures the weight of the i^{th} criterion and will be equal to zero if it is not activated yet.
The activation of the criteria is done according to a negotiation model between criteria-agents and document-agents, of which a detailed description is the subject of Section 3.3.

3.3 The dynamic pyramidal coordination structure

Principle. The goal of the document-agents is to select criteria-agents so as to maximise the evaluation of their documents while respecting some constraints related to the usage of computer resources. To a document-agent A_i and a criterion-agent C_j one may associate a partial utility function $U_{ij}(t)$ which measures the interest of A_i to require an evaluation from C_j at time t. A strategy of A_i may hence be evaluated as a global utility function $U_i(t) = \sum_j U_{ij}(t)$.

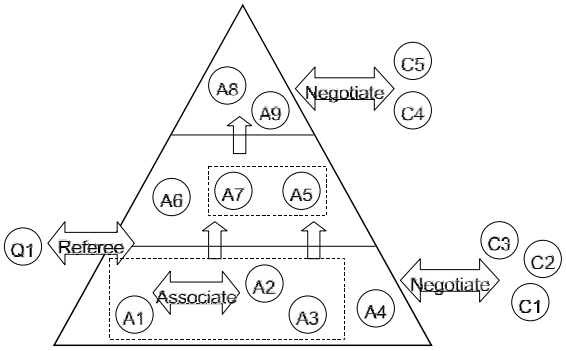

Figure 1: Dynamic Pyramidal Coordination Structure

This process can be performed using a structure called a "Dynamic Pyramidal Coordination Structure" of agents that we use both to impose constraints and to enforce synergy between agents. This pyramidal coordination structure is built up of associations of agents whose utility functions match, at least partially, with each other.

Definition 1 *A dynamic pyramidal coordination structure for a set of agents with respective utilities $U_1 \dots U_m$, with m denoting the number of criteria, is a levelled partition of $\mathcal{A}$ whose components $\{L_1 \dots L_p\}$ contain each a set of agents $\{A_{k1} \dots A_{kx}\}$ and a set of associations $\{S_{k1} \dots S_{ky}\}$ of agents.*
Each component of the partition is called a level.

An example of a pyramidal coordination structure of three levels with nine agents and 2 associations is shown in Figure 1. On this diagram, agents belonging to a same association are grouped in rectangles.
When a new document is to be evaluated, a new document-agent is created and placed at the lowest level of the pyramid. It then starts collecting evaluations from criteria-agents. If at some moment its utility function has reached a certain threshold, then it is entitled to raise to the next level. At each level, the amount of computing power available to a single agent increases but remains limited so as to limit its possible choice of a criterion. This principle enforces resource optimisation. However, this optimisation should not be inflexible, and this is where the organisation of the agents in associations comes into play.
A first and simplified informal definition of associations is given below. This definition is completed in Section 3.3.

Definition 2 *An* association *is a set of agents whose utilities interact with each other due to some shared features.*

In our application, these features are features of the document that the document-agents manage. Such features might be, for example, the author of the document, or the web site they come from, or some linguistic feature. The

document-agent has some knowledge of the features of its document, and this knowledge may increase as a result of the application of some criteria-agents.
On entering in a level of the pyramidal structure, a document agent announces itself as well as the features of its document it already knows about. On this basis, it starts forming associations with existing agents at that level. It also collects from criteria agents information about their possibilities, their usage constraints and their requirements in terms of computing resources.
The utility U_i of each agent A_i is now depending only on the actions that A_i chooses. To make this choice the document-agent exploits the established associations by using the knowledge collected by the other agents belonging to the same associations. The document-agent can then decide its own strategy taking all the implications into consideration. The choice of optimal criteria and the insurance they will "pay back" will, of course, depend on the past decisions of the agents of the same associations, whose strategies have already provided feedback. Afterwards, the agent will communicates the results of the application of its strategy to other agents of the associations it belongs to, so that they can be informed about possible decisions to take in their own strategies.
So, this pyramidal coordination structure, limits the use of computer resources by document-agents, preventing them to apply resource intensive criteria while they have not proved that the document they carry is worthwhile. At the same time, this structure helps document-agents to exploit their relations through the associations so as to increase their utilities. The query-agent takes this decision according to the result of the utility functions that document-agents obtained during their processing.
Each document-agent proceeds repeatedly:

1. A_i broadcasts a query for identifying criteria-agents and receives from them the information $< C_k, t_k : v_k >$ where that v_k is the range of the utility value returned by the criterion-agent and t_k is an estimate of the computing time needed.

2. A_i broadcasts an announce of its presence and receives in exchange proposals for joining existing associations in the form $< S_x, A_y >$ where A_y is the document-agent that A_i can contact to join the association S_x (cf. definition 2).

   After this step, A_i can join in one or more associations.

3. A_i builds its own maximisation strategy with respect to the criteria-agents which it will contact to analyse its document.

4. A_i performs the maximisation of its utility function U_i according to its strategy which it updates according to the messages that it receives from the documents-agents sharing common associations.

5. A_i distributes the information about the results obtained with the criteria-agents it has chosen to the document-agents of the associations to which it belongs.

Once this process is completed by a document-agent at the lowest level of the pyramidal structure, it may or not be entitled to reach a higher level, where the process repeats. Each level L_k of the Dynamic Pyramidal Coordination Structure has a minimal utility threshold U_k^{min}. To reach this level, a document-agent A_i need to have $U_i \geq U_k^{min}$.
The messages exchanged in step 4 are the base information needed by a agents from the same associations to build their own strategies. Let's consider an association S_x, constituted, for example by documents sharing the same author, then if criterion C_k has given the best results for one agent in S_x then the other agents of S_x will tend to use the same criterion C_k to increase their utility.
In a different situation, the agents may coordinate their actions differently. Let's imagine two documents-agents $A_{i'}$ and A_i in a same association because they have obtained good results with the criterion C_k. These agents need not to remain in the same association for ever. Indeed, relationships that hold before can disappear as agents apply other strategies by choosing another criteria. For instance, if at a later time t', $U_{i'm}(t') = 10$ and $U_{im}(t') = 0$, then an optimal strategy for document-agent A_i would be to leave the association with $A_{i'}$. In this example, the previous confidence between A_i and $A_{i'}$ disappears due to differences in their respective strategies and therefore A_i and $A_{i'}$ do not need to communicate anymore their results.

Advantages of the coordination structure. This pyramidal coordination structure exhibits several important properties.

1. It allows to dispose at anytime of a temporary result of the filtering of the documents simply by examining how the corresponding agents are distributed in the pyramid, which reflects the current value of their utility functions.

2. Thanks to associations, it enforces cooperation between agents sharing common properties.

3. Associations reduce the time lost in unnecessary computations since document-agents avoid choosing criteria having proved to be inefficient on similar documents.

4. The coordination of the document-agents is managed thanks to the organisation of the dynamic pyramidal structure in several levels. These levels set dynamically the priority of the document-agents and make them evolve with their utility U_i.

5. This structure is flexible. It can be adapted incrementally and easily since new criteria may be introduced in the system simply by adding new criteria-agents.

6. There is no implicit central algorithm, no implicitly shared behaviour, no central ruling agent nor plan: the agent behaviour and usage of the resources is imposed by the coordination model constraints.

What is an Association?. In this section, we propose a more precise definition of what is an "association" and present the features required from associated agents.
An association of agents is a group of agents, but not characterised by a group rationality. The agents have of course their individual rationality but do not receive any direct payoff as a result of the group's performance. Such characteristics are more relevant for coalitions [5, 19, 20] and teams [18, 17].
As in congregations, each agent has its own utility function that it maximises. To do so, it takes in consideration the benefits it has of joining associations. Agents join only associations with which they share features. They are free to leave these associations if during their processing they find that their utility does not evolve within these associations. They join associations in order to satisfy their needs in better conditions.
Even if the concept of association presents similarities with that of congregation defined in [2] such as those described above, nevertheless they remain two different concepts.

1. [2] defines the concept of congregation as:

   > "A group of agents that has come together for some mutually beneficial purpose, exchange of goods, services or informal accomplishing of tasks or an aggregation in order to accomplish goals which could not be met separately."

   As an example they propose that of "clubs" in human society. Thus they associate to the concept of congregations a cost for joining them. Agents should pay a fee to join the congregation and to have access to its services. Such a fee can be monetary for some congregations like clubs. On the contrary, in associations, agents do not need to pay any fee.

2. There is no substantial investment to create an association, while this is needed in congregations.

3. Contrary to a congregation, an agent joining an association does not seek a particular partner with whom it will interact directly but rather a group of agents providing knowledge.

4. An association is initially characterised by the agent that took the initiative to create it. Its characteristics do not change with time. For example, in our application, an association might be created to group agents that operate on documents written by the same author.

5. The integration of an agent into an association is done only on the basis of the objective features associated to the association and not on the basis of the agents already belonging to it as it would be the case in congregations [2].

6. An agent does not need to know all members of an association it belongs to. One "entry point" is enough to share its results with other agents and, of course, to take benefit of other agent's "know-how".

7. The belonging of an agent to an association is not necessarily a long-term contract. In the case of our application, it might be just the duration of a user's query.

8. Associations do not have any kind of central control.

A formal Model of Associations. In this section, we provide a formal model of associations.

Definition 3 *An association S_i is represented by each agent A_j as a triple $< A_i, F_i, K_i >$, where:*
A_i is the "entry point" of 'agent A_j in the association, i.e. a set of agents that A_j will have as interlocutors in the association.
F_i: is the list of the shared features of the agents in the association. This list is initially complete as it serves as a basis for integrating new agent in this association.
K_i: is the knowledge acquired by agent A_j from the association S_i. This knowledge is in a form of rules that the association gathered from its different participants.

The knowledge pertaining to an association is updated as members communicate to each other their observations on the operations they perform. In the case of our application, this knowledge is related to the results of document evaluation. For example, the vector analysis using the Tf*Idf measure might have good results for all documents of a given author.

Definition 4 *A rule R_k of an association S_i is represented as a triple $< C_j, Dom_k, p_k >$, where:*
C_j: is the criterion concerned by the rule R_k.
Dom_k: is the application domain of the rule R_k.
p_k is the probability that R_k has a positive result.

4 Implementation

At the agent level, the coordination protocol is realised as per-role finite-state machines implemented in java and that agents may reuse at will, provided they implement the necessary methods for negotiation, etc. From the interaction point of view, KQML has been selected as communication language.
However, the implementation of such a coordination model is confronted with two issues: firstly the system mixes persistent agents, like criteria-agents, with volatile ones, like document-agents, which have a limited time to live. Secondly, the association protocol relies on message broadcasting which may be an issue with volatile agents and is

also an issue with respect to bandwidth when hundreds of agents are involved.
To cope with this we introduced the notion of a *class-agent*, a kind of specialised facilitator the role of which is to instantiate new agents of a particular class, to keep track of their existence and to broadcast them relevant messages.
Class agents are programmed at the same time as the class of the agents they will represent. They have a knowledge of the "competence" of the instances of the class in the form of the set of messages they are able to reply to. This is declared at the time of the programming of the agents.
The behaviour of a class-agent follows a limited number of rules. Let m be the message received by a class-agent, with a set of competences $\mathcal{M}$.

1. If m is a brokering message[7] and $content(m) \in \mathcal{M}$, then a new instance of the class is created, and the message $content(m)$ is forwarded to it, either directly (brokering) or through a pipe (recruiting).

2. If m is a `broadcast` and $content(m) \in \mathcal{M}$ then m is forwarded to all instances.

As an example, let's examine how new document-agents are managed.

1. The query-agent `Q1` requires to the document class-agent `DCA` the help from a document-agent, which instantiate a new document-agent `A1` and forwards the content of the message (Figure 2).

2. `A1` looks for associations for some features of the embedded document (Figure 3).

3. `A1` looks for available criteria. On the one hand it broadcast a message asking for criteria that fit in `A1`'s time limits and, on the other hand, it inspects its associations. (Figure 4).

```
(broker-one // Q1 calls for a document-agent
  :sender Q1
  :receiver DCA
  :content (perform
             :content evaluate(<document>, <query>)"))

(forward // DCA instantiates one and forwards
  :from DCA
  :to A1
  :content (perform
             :sender Q1
             :content evaluate(<document>, <query>)"))

// A1 is now created and gets credit
(ask-one :sender A1 :receiver Q1 :content "credit(0)")
(tell :sender Q1 :receiver A1 :content "20")
```

Figure 2: Creation of a document agent.

Since the outermost envelope, the broadcast, has no specified destination, it is intercepted by `DCA`, the class-agent

[7] In KQML, brokering messages are `broker-one`,`recommend-one`, `recruit-one`.

```
(broadcast
  :content (ask-all :content "association(<features>)"))

(tell :sender A2
  :receiver A1
  :content "association(<features>)")

(tell :sender A1 :receiver A2
  :content "member(A1,association(<features>))")
```

Figure 3: Document agents associating.

```
(broadcast
  (recommend-all :sender A1
    :receiver CCA
    :content (ask-if
               :content "evaltime(<document>)<20")))
```

Figure 4: Looking for criteria

of the document-agent, which broadcasts the query for recommendation to all class-agents (not to all agents). Those the instances of which may understand the message will forward the innermost query.
The advantages of this implementation are:

1. The pyramidal coordination model relies heavily on broadcasting and thanks to our implementation, the effective broadcast is limited to relevant agents.

2. Document and query agents may be implemented without any knowledge about criteria, and the list of criteria may evolve dynamically.

5 Conclusion

In the presentation we made, we voluntarily have limited the discussion on the criteria themselves. However, a number of issues remain open on this topic. In particular we envision to apply machine learning techniques combined with "rewards" from the user to improve the weighting of the criteria with respect to one another. This behaviour should impose a revision of the coordination protocol.
Nevertheless the pyramidal multi-agent coordination model we have presented offers an important degree of flexibility since agents may integrate such a structure, or leave it dynamically. We have also introduced the notion of multi-agent association on which the optimisation of the structure relies. This concept was compared to related ones like coalitions, teams and congregations.
Through the Safir project, we have shown the practical advantages of such a coordination model.
As a continuation of these principles, the intermediate results of the Safir project are being tested in the context of a European project aiming at the discovery and filtering out of racist web pages. This project is pursued with linguistics labs in France, Germany and Ireland. The linguistic issues addressed provide us with an occasion to validate our approach in a more complex configuration.

References

[1] C.H. Brooks and E.H. Durfee. Congregation formation in multiagent systems. *Journal of Autonomous Agents and Multi-agent Systems*, 2001.

[2] C.H. Brooks and E.H. Durfee. Congregating and market formation. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS, 2002.

[3] C.H. Brooks, E.H. Durfee, and A. Armstrong. Introduction to congregating in multiagent systems. In *Proceedings of the Fourth ICMAS*. ICMAS, July 2000.

[4] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 1992.

[5] Philippe Caillou, Samir Aknine, and Suzanne Pinson. A multi-agent method for forming and dynamic restructuring of pareto optimal coalitions. In *Proceedings of the first AAMAS*, pages 1074–1081. ACM Press, 2002.

[6] Claire Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4):65–80, 1997.

[7] M. Cutler, H. Deng, S.S. Maniccam, and W. Meng. A new study on using html structures to improve retrieval. 1999.

[8] Harald Lngen Dafydd Gibbon. Consistent vocabularies for spoken language machine translation systems. In Jost Gippert, editor, *Multilinguale Corpora. Codierung, Strukturierung, Analyse*, pages 169–178, Prague, 1999. Enigma Corporation.

[9] Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, 1994.

[10] Schutze H. Dimensions of meaning. *Supercomputing*, 1992.

[11] T.A. Letsche and M.W. Berry. Large-scale information retrieval with latent semantic indexing. *Information Sciences - Applications*, 100:105–137, 1997.

[12] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the COLING-ACL, Montreal, Canada*, 1998.

[13] P. H. Matthews, editor. *The Concise Oxford Dictionary of Linguistics*. Oxford University Press, 1997.

[14] M.Cutler, Y. Smith, and W. Meng. Using the structure of html documents to improve retrieval. 1997.

[15] G. Salton and C.Buckley. Term weighting approaches in automatic text retrieval. In *Information Procession & Management*, volume 24, pages 513–523, 1988.

[16] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[17] T.W. Sandholm and V.R. Lesser. Coalitions among computationally bounded agents. *AI*, 94:99–137, 1997.

[18] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.

[19] M. Tambe. Towards flexible teamwork. *Journal Of AI Research*, 7:83–124, 1997.

[20] M. Tsvetovat, K. Sycara, Y. Chen, and J. Ying. Customer coalitions in the electronic marketplace. In *AAAI*, 2000.