In: *Proceedings of the Eleventh International Conference on Cooperative Information Systems* (CoopIS-03), pp. 93–109, Springer-Verlag LNCS Series.

Advertising games for Web Services*

Alessandro Agostini
ITC-irst Trento

Paolo Avesani
ITC-irst Trento

Abstract

We advance and discuss a framework suitable to study theoretical implications and practical impact of language evolution and lexicon sharing in an open distributed multi-agent system. In our approach, the assumption of autonomy plays a key role to preserve the opportunity for the agents of local encoding of meanings. We consider the application scenario of Web services, where we conceive the *problem of advertisement* as a matter of sharing a denotational language. We provide a precise formulation of the agents' behavior within a game-theoretical setting. As an important consequence of our "advertising games," we interpret the problem of knowledge interoperability and management in the light of evolutionary dynamics and learning in games. Our methodology is inspired by work in natural language semantics and "language games."

Keywords: Web information systems and services; semantic interoperability; negotiation protocols; peer-to-peer cooperation.

1 Introduction

The recent trend in the information technology is the promotion of distributed solutions. Web services and peer-to-peer architectures are two of the most representative examples (see for instance [CCMW01, GHMF01, BBMN02, GZ02, MSZ01] and the references cited therein). The common objective of these fast-growing research and technological areas is the design and the development of autonomous and distributed systems that overcome the restrictions of client-server architectures. As far as we know, a main challenge of the web services is how to achieve a full interoperability among distributed processes. In particular, the efficient advertisement of a service content is really a crucial task to reach full interoperability, because it addresses the actors concerned to an effective communication based on a shared knowledge representation. Advertisement is in fact the first step in the deployment of a web service. A main purpose of a service provider on the Web is the publishing of the interaction protocol by which information seekers may connect reliably with the provider. Together with the interaction protocol, the

*This work was partially supported by the *Provincia Autonoma di Trento* Project EDAMOK ("Enabling Distributed and Autonomous Management of Knowledge") under deliberation number 1060-4/5/2001. Electronic correspondence to: agostini@irst.itc.it, avesani@irst.itc.it.

service provider communicates the specification of the service content. Usually, understanding of such specification is made easier by assuming that the agents agree on a common ontology, which also fixes the semantic relations involved in the service. This centralized approach, however, is not always sustainable on the large scale (see for instance [BB02, BBT02] and the references cited there).

From the standpoint of language disambiguation, categorial indeterminacy of service advertisements is an important problem in Web services. Categorial indeterminacy is the name—historically emerged from the community of natural language semantics and situated grounded word semantics (see for instance [SK99, SKML02]), of a kind of partial incoherence of a set of languages and relations distributed over a possibly huge set of agents. In these distributed environments, most interesting situations apply to agents concerned with the management of rich categorial databases. For example, think of an information agent in the role of a provider as the holder of the directories of *Google*, and of an information agent in the role of a seeker as the holder of the directories of *Yahoo!*. In such situation, it is indeed possible and very common that a particular topic, chosen by the seeker, is conceptualised in very different ways by the provider. A possible consequence is the emergence of polysemy—that is, the same word may be used by the agents to denote different categories or topics. Successful advertisements for a web service often depend on the ability of choosing the "right" denotation among those that would be used by the agents for the same service. A similar ability is related to the problem of guessing the content of an user searching the web, where a common technique is *context guessing* (see for instance [Law00]; see also [NPSR99]). Unfortunately, today web service techniques "by guessing" are limited to cases where potential semantic contents of the required services can be identified on the basis of the keyword query. This is often not the case in the Web search.

The attempts to design solutions to categorial indeterminacy have produced the development of mapping algorithms (see for instance [MBDH02, RB01, SBMZ03]). A (schema, model, ontology, context) mapping can be effective to support the semantic interoperability between the web services, and it may be possibly used to reduce categorial indeterminacy. However, mappings may be not suitable to face with the assumption of the open World (Wide Web), where new actors join and leave the scenario continuously. A reason is that the strategy to solve the problem of interoperability based on mapping requires to find the relative mapping for each pair of actors. An undesirable consequence is that the deployment of a new actor can be very demanding, especially whenever the actor community is very large.

In this paper, we propose a general approach that exploits the meaning-to-meaning relationships found in many domains of interest, one of them being the Web services. In particular, we define a game suitable to model any domain with meaning-to-meaning relationships. Our solution concept for the problem of semantic interoperability is based on the emergence of a shared language for the advertisement in a system of agents. In other words, our idea is to conceive the problem of knowledge interoperability as a matter of sharing a denotational language. A question we investigate in this paper is whether and how language and expecially language evolution are functional to get an efficient advertising service and, as a consequence, a successful matching between an information seeker and the relevant service provider. Our methodology is inspired by work in natural language semantics and "language games" [Ste96a, SK99]. A similar approach with different assumptions has been adopted in the domain of robotics [Ste96b, Ste98], where the notions like "naming game" and "guessing games" have been proposed.

This paper is organized as follows. In Section 2 we advance and discuss some related work, expecially language games. In particular, we compare to our approach "naming games" and their major variant of "guessing games." In Section 3 we illustrate an application scenario in the Web services, which is formalized in Section 4 and further discussed through a detailed analysis of the players' behavior in Section 5. We add some final remarks to discussion in Section 6. We conclude in Section 7 with a summary of the results of this paper.

2 Related Work

Language games have been introduced to study natural language semantics by experimental use of visually grounded robots in a distributed group [Ste96b, Ste98]. The problem was to determine the systematic relations between language forms ("words"), their meanings ("concepts")—expecially local meaning assigned to words by a single agent, and their referents ("instances," "objects"). A corollary of the problem solution is the evolution of the agents' language and of their lexicon. This problem is referred to as "the grounding problem" in [Ste01].

A major sub-problem is "naming," that is, how vocabulary and meanings are learned individually and a shared lexicon eventually emerges in a group of agents. The problem of naming may be expressed in game-theoretical terms, and was extensively studied since the *naming games* [Ste96b]. In short, each player from a uniform set of agents has a set of words and a set of objects, and randomly associates a word to an object, called "the topic," to form a local lexicon. In a naming game, it is assumed that all the agents gain a positive payoff in cooperating, "but only if they use the same language" [Ste96b]; thus, a naming game is a coordination game in the sense of game theory (see for instance [OR94]). It is repeatedly played among randomly chosen pairs of players and involves a different couple of agents at each repetition of playing. A naming game is adaptive, in the sense that the players in the game can change their internal state. A reason for changing is to be more successful in playing future games.

There are several variations and extensions of a naming game. In relation to the problem of finding successful advertising strategies for the web services, the most interesting to us are all those which consider a player's "guessing" of the semantic content of a publicly played linguistic expression ("query"). This is the case, for example, of the query-answering systems in the Web search [Law00].

Following [Ste98], a *guessing game* is played between two agents, called "the speaker" and "the hearer." By convention, the speaker always moves first. In its simplest form the game is played in two steps. First, the speaker chooses an object (called "the topic") from other objects in a given context, the hearer attempts to guess it. The context is a fixed set of objects taken from the reality at the very beginning of the game. A context characterizes the domain of a guessing game. In contrast with the naming games, instead of playing directly the topic, in a guessing game the speaker explicitely plays to the hearer a linguistic hint. Then, the hearer guesses the topic through the verbal description ("verbalisation") provided by the speaker. The game ends in success just in case the hearer's guess is equal to the topic. The game ends in failure otherwise.

Three important points of a guessing game are that (a) topics are hidden, that is, they must be expressed through linguistic acts rather than exchanges of objects; (b) topics are the means not the end to the game playing, since the solution concept aims at discovering *sets* of topics—which are called "concepts" rather than single objects; (c) topics are always played by the speaker as a direct feedback of the hidden concept to guess. As we will see, the framework we are going to present is based on assumptions similar to (a) and (b). In contrast, a main change is about (c). In our "advertising games," in fact, the players' feedback on the meaning to guess is almost always indirect. One essential aspect of the guessing games we modify is that agents become capable of understanding each other by the fact that they are situated in a common environment and each one can detect the actions of the others in this environment. In our model the agents still communicate in a common environment, to be defined as the set of all objects the agents perceive. However, we rely on indirect communication, and the actions that the agents can mutually detect are primarly linguistic.

A remark concerns statistical learning; see for instance [Vap98]. Although our general approach is clearly comparable to statistical learning, in particular to statistical learning under unlimited amount of sample data, we do not know about any work in statistical learning literature which relates to the problem of lexicon evolution and language sharing in the context of the web services.

3 The Application Scenario

Before introducing the formal definition of an "advertising game," we sketch an example of an application scenario in the domain of web services.

Let us imagine a distributed community of actors with the common goal of sharing their effort in building web directories. We image the directories are of the same kind of *Google*, *Yahoo!* and *Looksmart*'s. The objective of the application is to define an anytime alerting service that informs an actor, who classified a topic according to some category in a given directory, of a new reviewed web page, classified under the same category but in a different directory. We depict a scenario of pairwise interactions between actors, or *peers*, who can play two roles: *the seeker* and *the provider*. The seeker looks for latest news on a given topic, the provider advertises latest news on a given topic. In this scenario, the problem is twofold. On the one hand, the seeker has to find the "right" denotation to formulate the query about a topic of interest that has to be propagated over the network of peers. On the other hand, the provider has to detect whether the broadcasted query is

```
<TopicAdvertisement>                    <TopicQuery>
   <Name>...</Name>                        <Name>...</Name>
   <Topic>...</Topic>                      <Topic>...</Topic>
</TopicAdvertisement>                   </TopicQuery>
```

Figure 1: **XML Communication Protocol.** On the left hand side, a sketch of the protocol to support the advertising of a new topic of interest, i.e. a directory to collect and to share bookmarks according to a predefined category. On the right hand side, a sketch of the protocol to support an inquiry over the network to receive recommendations on related category.

relevant to the local denotation of the news topic. In both cases, an actor faces with the needs of a shared language denotation for the different but similar topics.

The basic operations, or actions, available to an actor are two: (a) advertising for a new service on a given topic—called the "target topic," and (b) searching for news on a given topic—called the "source topic." The figure illustrates these basic operations as XML schemata (Fig. 1). In terms of role assignments, notice that advertising is primarily an action taken by a provider, while searching is primarily an action taken by a seeker. However, each actor may take both actions, since we have stated that actors are peers by definition. In case (a), the specific issue is to formulate an (extra-linguistic) advertisement by using the linguistic expression choosen by the seeker to communicate the source topic. To be effective, we require the advertisement to be relevant ("similar") to the source topic. In case (b), the issue is finding a language expression ("label," "query") to denote the source topic. This expression must be equivalent to the way the provider advertises the target topic, in a sense we will make precise below. Otherwise, a mismatch between the source and the target topics may occur. The way the provider advertises a service and the seeker formulates a search query are crucial for helping the actors to match.

We imagine that the administrator of *Google*, playing the role of the seeker, has defined a topic in a web directory (Fig. 2). The topic is a category encoded by a node identifier (the pathname `/top/home/cooking/soups_and_stews`), a mnemonic label to refer to (the syntagm "`fish and soup`"), and a collection of web locations representative of the topic (`bookmarks`). Similarly, the *Yahoo!* administrator, playing the role of the provider, defines a topic as a category in the *Yahoo!*'s directories. Now suppose that the administrator of *Google* is interested in finding whether there are other web directory administrators that have collected unknown web references for the topic "fish and soup." Unfortunately, the topic identifier is not suitable for this purpose, because the encoding (i.e., the pathname) depends on the particular web directory. However, the mnemonic label can indeed be effectively used by the administrator, because it can be revised thanks to meaning negotiation. The challenge of a successful application is to bring the administrators of the web directories to converge, by meaning negotiation, towards a common denotation without forcing the alignment of the directories' structure.

```
<topic>

<directory>
   Google:/top/home/cooking/soups_and_stews/fish_and_seafood
<\directory>
     <name>fish and seafood</name>
     <bookmark>
        <uri>
          http://www.fish2go.com/rec_0120.htm
        </uri>
        <excerpt>
          Finnan Haddie and Watercress Soup: made with smoked
          haddock, potatoes, watercress, and milk.
        </excerpt>
     </bookmark>
     <bookmark>...</bookmark>
     ...
     <bookmark>...</bookmark>
</topic>

<topic>

<directory>
   Yahoo:/top/society_and_culture/food_and_drink/cooking/recipes/
          by_ingredient/fish_and_seafood
<\directory>
     <name>beef</name>
     <bookmark>
        <uri>
          http://www.freshfish4u.com/fishnet/recipes
        </uri>
        <excerpt>
          For a range of fish and shellfish species. Alphabetically
          organized.
        </excerpt>
     </bookmark>
     <bookmark>...</bookmark>
     ...
     <bookmark>...</bookmark>
</topic>
```

Figure 2: **Topic XML Schema.** A couple of examples of directories extracted from *Google*'s web directories. Each directory is defined by an identifier (in this case the full path), a name that provides a mnemonic support to detect the related category associated to the directory (the local denotation), and a collection of bookmarks.

4 The Framework

We model the foregoing application scenario together with the pairwise interactions of the actors by a kind of language game. More specifically, we introduce the notion of an "advertisement game," where the local representation of a topic is hidden, that is, it is not shared among the actors. In an advertising game, the only way to assess whether a seeker's query about some topic matches a provider's advertisement on the same or on a similar topic is to go through an inductive trial and error process. From the standpoint of the game design, our ultimate goal is to reduce the search failures which are the result of a provider's misunderstanding of a seeker's queries. In short, we aim to minimize categorial indeterminacy of service advertising and search.

4.1 Basic Components

In this subsection, we present and discuss the basic elements and properties of an "advertising game." An *advertising game* is defined over a countable (nonempty) collection $\mathcal{D}$ of *objects* shared among a (recursive) set Λ of agents—we call the set $\mathcal{D}$ *game domain*. Intuitively, $\mathcal{D}$ denotes the class of objects in the reality ("universal set") that all players perceive in the same way according to the nature of the game. For example, if the game is used to model information extraction or retrieval, $\mathcal{D}$ is a set of achievable documents.[1] For another example related to the application scenario of Web services (Section 3), $\mathcal{D}$ is the set of documents contained in the nodes of a web directory.

Each agent has a set of words and a set of meanings. In this paper we assume that a word (also "label") is any finite string of symbols from an alphabet of symbols *Sym* without further specification. Intuitively, a meaning is a proper subset of the game domain. For example, if the game domain is information retrieval then a set of documents is a meaning. (We do not discuss here how these are related to each other, since we assume meanings to be primitive elements of our model.) Each agent has a "lexicon." Informally, a lexicon is a set of pairs of the form $\langle \mathit{word}, \mathit{meaning} \rangle$. An agent's lexicon may be either empty (no associations) or incomplete, that is, there may be some word with no meaning or some meaning with no word associated. Polysemy and synonymy, that is, the same meaning may be expressed by the agents through different words, may occur. Notice that there is not one lexicon for all agents, but each agent has his own local lexicon. A general lexicon as a system's component (that is, a global element of the game) can indeed be defined by the union of the agents' lexicon, and we do it below. However, this general component is not necessary in our game construction. A motivation is that we want to deal with locally computable elements, and a global lexicon cannot be computed by any agent, because the set-theoretical operation of union runs over all the agents.

[1] The *content* of the documents, however, may be perceived in different ways by the players, and in fact this is the main point of this paper: how the agent can compare documents whose content is locally assigned by each player independently?

Given a player i, let Lx_i denote the player's *lexicon*. Let $pow(\mathcal{D})$ denote the power set of $\mathcal{D}$. We define Lx_i to be any subset of $\mathcal{L}_i \times \mathcal{C}_i$, where $\mathcal{L}_i$ denotes the language (i.e., a set of "words") of player i, and $\mathcal{C}_i \subseteq pow(\mathcal{D})$ denotes the set of meanings of i (given $\mathcal{D}$). We define the *game lexicon* (written: Lx) by the union of Lx_i over all $i \in \Lambda$. To provide easy starting of the game, for simplicity we assume that

(1) there is at least a player $i \in \Lambda$ such that $Lx_i \neq \emptyset$.

In particular, $\mathcal{L}_i \neq \emptyset$ and $\mathcal{C}_i \neq \emptyset$. Only a player i satisfying (1) is allowed to play the role of the seeker at the beginning of the (repeated version of) the advertising game. In other words, (1) is the condition we require to a player to be able to start the game.

Each agent has a *sampling function* that transforms every meaning in the agent's set of meanings into an infinite sequence ("ω-sequence") of nonempty subsets of objects in the game domain $\mathcal{D}$. To define an agent' sampling function we introduce some notation. We write $length(\sigma)$ for the length of a finite sequence. Let N denote the set $\{0, 1, 2, ...\}$ of natural numbers. We write σ_n for the nth element of σ, $0 \leq n < length(\sigma)$.

(2) Definition: Let game domain $\mathcal{D}$, player i and meaning $m \in \mathcal{C}_i$ be given. A *sample of m in $\mathcal{D}$ by player i* is an infinite sequence over $pow(\mathcal{D})$ (written: $\mathsf{Sample}^i(m \mid \mathcal{D})$) such that:

(3) $\bigcup_{n \in N} \mathsf{Sample}^i(m \mid \mathcal{D})_n = m$.

We say that the nth element $\mathsf{Sample}^i(m \mid \mathcal{D})_n$ of $\mathsf{Sample}^i(m \mid \mathcal{D})$ is the *sample* ("instance," "example,"...) by player i of m in $\mathcal{D}$ *at n*.

There are "good" and "bad" samples. For goodness, a necessary (but not sufficient) condition to efficiency is that for every $n \in N$, $\mathsf{Sample}^i(m \mid \mathcal{D})_n$ is finite. For badness, an example is the sample $\langle \emptyset \ m \ \emptyset \ \emptyset \ \emptyset \ldots \rangle$ for m being a meaning with an infinite extension.

(4) *Remark:* A question is why we resort to infinite sequences to define a meaning's sample. The answer is in fact fundamental to understand our approach. We define sampling "in the limit," in the sense that we assume that no finite sampling can capture completely any (sufficiently interesting) meaning. Of course, a sample may be an infinite sequence of a *finite* number of sets (subsets of $\mathcal{D}$) even for meanings with denumerable extension.

We note that the infinite sequence $\mathsf{Sample}^i(m \mid \mathcal{D})$ defines a preference order over samples for m in $\mathcal{D}$. Also observe:

(5) Lemma: For all players i and all game domains $\mathcal{D}$, and for every meaning $m \subseteq \mathcal{D}$, $\mathsf{Sample}^i(m \mid \mathcal{D})$ exists if and only if $m \in \mathcal{C}_i$.

Proof: The "only if" direction follows immediately from Definition (2). To prove the "if," suppose that $m \in \mathcal{C}_i$. Observe that the cardinality of m is countable.

Then m is a r.e. set and we can list all the members of m according to a recursive procedure. For example, we can produce the first element of the list, say o_1, randomly from m, the second element of the list, say o_2, randomly from $m - \{o_1\}$, the third element of the list, say o_3, randomly from $m - \{o_1, o_2\}$, and so on. For all $n \in N$, define $\mathsf{Sample}^i(m \mid \mathcal{D})_n = o_n$. It is immediate to verify that (3) holds. ■

Each agent has a *preference relation* ("utility function") over the lexicon. Let Z denote the set of integers (positive, negative, and zero).

(6) DEFINITION: Let player i be given. A *utility function* of i is a total computable function from Lx_i to Z.

In other words, a utility function decides what utility a player gets for his choice. Intuitively, negative numbers are bad outcomes, positive numbers are good outcomes, and 0 is the neutral outcome. Big positive outcomes are nicer than small positive outcomes, and similarly for negative numbers. Observe that a utility function is locally computable, that is, each agent computes the utiliy of assigning a meaning to a word over its own lexicon.

We refer to values of a utility function as *payoffs* (or "utilities"). For all pairs $\langle w, m \rangle \in Lx_i$, let $\mathsf{Pref}^i(w, m)$ denote the preference, or utility, of player i to use a word w to communicate a meaning m. If $\mathsf{Pref}^i(w, m) = n$, we say that w is the n-preferred word by player i for m or, equivalently, that m is the n-preferred meaning by player i for w. Since we are interested in the repeated playing of a game, the updating of the players' preferences over time is important. In order to deal with time, we extend Definition (6) over N as follows. Given $t \in N$, let $\mathsf{Pref}^i(w, m, t)$ denote the payoff $\mathsf{Pref}^i(w, m)$ *at time* ("move") t of the game history. To shorten notation, we write $\mathsf{Pref}^i(w, m)$ for $\mathsf{Pref}^i(w, m, 0)$.

(7) *Remark:* An infinite sequence $\mathsf{Sample}^i(m \mid \mathcal{D})$ defines a preference ordering over the samples of m. As a consequence, we have two related kinds of preferences, respectively over meanings with respect to a given word, and over the samples of a given meaning. In an advertising game, the former preferences are properly used by the seeker, who decides what meanings to play; while the latter preferences are properly used by the provider, who uses sampling as a way to communicate the seeker's requests; see the protocols in Section 5 for details.

We are now ready to define a "local similarity relation."

(8) DEFINITION: Let player i and game domain $\mathcal{D}$ be given. A *local similarity relation* of i on $\mathcal{D}$ is a recursive relation on $\{i\} \times pow(\mathcal{D})^2$ such that the restricted relation on $pow(\mathcal{D}) \times pow(\mathcal{D})$—called *similarity relation*, is a reflexive, symmetric and transitive binary relation.

Intuitively, given $m, m' \subseteq \mathcal{D}$, $\mathsf{Sim}(i, m, m')$ is true if m and m' are "similar" from the i's viewpoint. Otherwise, $\mathsf{Sim}^i(m, m')$ is false. The simplest example of similarity is equality. Notice that this definition of similarity emphasizes the local perspective of individual agents. For notational convenience, from now to the end of this paper we write $\mathsf{Sim}^i(m, m')$ in place of $\mathsf{Sim}(i, m, m')$.

(9) *Remark:* Various aspects of meaning can be used to determine the exact content of similarity, usually depending on the application domain and the appropriate definition of similarity for that domain. When meanings are sets, and this is the case in an advertising game, a similarity relation is identical to an equivalence relation in discrete mathematics. However, we prefer to use a similarity relation rather than an equivalence relation, because one main goal of our research agenda is to lead off to experimental work on the impact of language evolution and lexicon sharing in the Web services. Similarity relations play an important role in our foreseeable experimental work [AA03b]. For a similar reason, notice that often similarity is a continuous measure, rather than a binary relation, so a question arises on motivations. On the one hand, we may extend our definition, for example to integers, denoted by Z, and define the local similarity measure of a player i on game domain $\mathcal{D}$ to be a recursive *function* from $\{i\} \times pow(\mathcal{D})^2$ to Z. (Recall that the relation $\mathsf{Sim}^i(\cdot,\cdot)$ is equivalent to a function with codomain $\{0, 1\}$.) On the other hand, at the present stage of our work we do not need to deal with the additional power but the additional complexity of continuous similarity metrics.

Whatever $\mathsf{Sim}^i(\cdot,\cdot)$ is defined, we require it satisfies the following property. We rely on some notation. Given any (finite, infinite) sequence σ over $pow(\mathcal{D})$, let $content(\sigma)$ denote the union set of elements in σ. For example, take $\mathcal{D}$ be the set of natural numbers and $\sigma = \langle \{0\}\ \{0, 2, 6\}\ \{2, 74, 8, 1\}\ \{1\} \rangle$. Then $content(\sigma) = \{0, 1, 2, 6, 8, 74\}$. Observe that $content(\sigma) \subseteq \mathcal{D}$, so $content(\sigma)$ is a meaning. The announced property follows:

(10) For all players i, j and for all $m \in \mathcal{C}_i$, $\tilde{m}' \in \mathcal{C}_j$ such that $\mathsf{Sim}^i(m, \tilde{m}')$ is true, there are meaning $m' \subseteq \mathcal{D}$ and $k \in N$ such that:
  (a) $\tilde{m}' = \mathsf{Sample}^j(m' \mid \mathcal{D})_k$,
  (b) $m \cap content(\mathsf{Sample}^j(m' \mid \mathcal{D})) \neq \emptyset$, and
  (c) $\mathsf{Sim}^i(m, \mathsf{Sample}^i(m \mid \mathcal{D})_n)$ is true for all $n \in N$.

Observe that (10) implies that a strategy to succeed in the game playing is genuinely an inductive process. For example, it may happen that $\mathsf{Sim}^i(m, \tilde{m}')$ is true but $m \cap \tilde{m}' = \emptyset$.

5 Game Protocols—A result

We now analyze in more detail the behavior of the players in an advertising game. This is controlled by two related kind of rules. We call *game protocol* the computable "rules of the game." In an advertising game there are two kind of rules. First, the rules to govern the mutual inter-actions of the players. Second, the rules to update the players' preferences. We present each set of rules in turn. Notice that a game protocol is shared by all the players, but individual strategic components are present. An individual strategy is basically a set of rules which tells

Algorithm 1 "One-shot" playing of an advertising game.

proc *OneShotGame*($\mathcal{G}(\,Sym, \{i, j\}, \mathcal{D}, \rho\,), t) \equiv$
  i "thinks" to $m \in \mathcal{C}_i^t$ at time t;
  i "plays" $w \in \mathcal{L}_i^t$ to j at time t s.t. $\mathsf{Pref}^i(w, m, t)$ is the maximum over Lx_i^t;
  if $\exists m' \in \mathcal{C}_j^t \, \langle\, w\,, m'\,\rangle \in Lx_j^t$
    then $M_w^t := \{m \in \mathcal{C}_j^t \mid \langle\, w\,, m\,\rangle \in Lx_j^t\}$;
      j "thinks" to $m' \in M_w^t$ s.t. $\mathsf{Pref}^j(w, m', t)$ is the maximum over Lx_j^t;
      $\tilde{m}' := \mathsf{Sample}^j(m' \mid \mathcal{D})_t$;
      j "outputs" $\tilde{m}'$ to i at t;
      if $\mathsf{Sim}^i(m, \tilde{m}')$ **then** (* reinforcement *)
        i "outputs" True to j at t;
        PrefUpdating+($\mathcal{G}(\,Sym, \{i\}, \mathcal{D}, \rho\,), t, w, m)$;
        PrefUpdating+($\mathcal{G}(\,Sym, \{j\}, \mathcal{D}, \rho\,), t, w, m')$
      else i "outputs" False to j at t;
        PrefUpdating-($\mathcal{G}(\,Sym, \{i\}, \mathcal{D}, \rho\,), t, w, m)$;
        PrefUpdating-($\mathcal{G}(\,Sym, \{j\}, \mathcal{D}, \rho\,), t, w, m')$
      fi
    else
      j "outputs" ? to i; (* "call for help" - "?": special symbol *)
      $\tilde{m} := \mathsf{Sample}^i(m \mid \mathcal{D})_t$;
      i "plays" $\tilde{m}$ to j at t; (* direct feedback *)
      PrefUpdating-($\mathcal{G}(\,Sym, \{i\}, \mathcal{D}, \rho\,), t, w, m)$; (* adaptation i *)
      $\mathcal{L}_j^{t+1} := \mathcal{L}_j^t \cup \{w\}$; (* adaptation j *)
      $\mathcal{C}_j^{t+1} := \mathcal{C}_j^t \cup \{\tilde{m}\}$;
      $Lx_j^{t+1} := Lx_j^t \cup \{\langle\, w\,, \tilde{m}\,\rangle\}$;
      $\mathsf{Pref}^j(w, \tilde{m}, t) := 0$;
      $\mathsf{Sample}^j(\tilde{m} \mid \mathcal{D})_t = \tilde{m}$ **fi.**

the player who uses it how to move. A strategy may depend on earlier decisions and the opponent's moves taken in the game. A strategy is said to be winning if a player wins every game in which he or she uses it. We will see that the strategies in the protocols below are winning to supply the agents in the repeated playing of the game with a shared lexicon.

5.1 Communication Protocol

We assume that the seeker is chosen from a subset of players whose lexicon is nonempty at time $t \in N$. This guarantees the easy starting of the game, as we have said in (1). In the following, we identify the seeker with player $i = 1$ and the provider with player $j = 2$. We illustrate Algorithm 1 by dividing it into four main steps. We call the actions taken by the agents in these steps a *round* (or "play") of the advertising game. Without loss of generality, we describe the algorithm for $t = 0$.

Step 0. The players agree *by convention* to play on a common set of objects $\mathcal{D}$. Moreover, the players agree on the reinforcement parameter $\rho > 0$ to use to update their own preferences.[2]

Step 1. The seeker outputs a word $w \in \mathcal{L}_1$ while internally he plays (interpretes, "conceptualizes" it by) a meaning $m \in \mathcal{C}_1$ with positive preference $\mathsf{Pref}^1(w, m)$. We assume that $\mathsf{Pref}^1(w, m)$ is the maximum over the seeker's lexicon. In this case, we say that m is the most preferred meaning for w or that w is the most preferred word for m.
Comment: If the maximum of $\mathsf{Pref}^1(w, m)$ is not unique, then a user-internal procedure runs to pick out the final best choice. For simplicity, we assume that the seeker internal-procedure to rule out optimal preferences among equally desiderable alternatives is the simplest "take it randomly."

Step 2. The provider inputs w. Suppose that w is an entry in the provider's lexicon Lx_2. In other words, suppose that there is $\langle w', m' \rangle \in Lx_2$ such that $w' = w$.[3] Hence, the set $M_w = \{m \in \mathcal{C}_2 \,|\, \langle w, m \rangle \in Lx_2\}$ is nonempty. The provider picks a meaning $m' \in M_w$ with positive preference $\mathsf{Pref}^1(w, m')$. Again, we assume that m' is the most preferred meaning in M_w by the provider. So we assume that $\mathsf{Pref}^2(w, m') > \mathsf{Pref}^2(w, m)$ for all $m \in M_w$. We call m' the provider's *guess* of m (at some time t—we assumed $t = 0$ here). The provider applies his sampling function $\mathsf{Sample}^2(\cdot \mid \mathcal{D})$ to compute $\mathsf{Sample}^2(m' \mid \mathcal{D})$. Observe that $\mathsf{Sample}^2(m' \mid \mathcal{D})$ exists by Lemma (5) and the definition of Lx_2. The provider outputs $\tilde{m}' = \mathsf{Sample}^2(m' \mid \mathcal{D})_0$.[4] If either the provider's lexicon is empty or w is not an entry in the provider's lexicon, then the provider outputs the special word "?". Intuitively, this means that the provider publicly signals misunderstanding, as a "call for help" to the seeker. The game ends in failure. Then the seeker plays $\tilde{m} = \mathsf{Sample}^1(m \mid \mathcal{D})_0$ directly to the provider—"direct feedback" intervenes—and updates her preferences in order to reduce the probability that she uses word w to denote m in playing future "rounds" of the game. The provider updates his language: $\mathcal{L}_2 = \mathcal{L}_2 \cup \{w\}$,[5] adds $\tilde{m}$ to $\mathcal{C}_2$—we refer to $\tilde{m}$ as to the provider's ("special") *guess* of m, and $\langle w, \tilde{m} \rangle$ to his lexicon and sets $\mathsf{Pref}^2(w, \tilde{m}) = 0$ and $\mathsf{Sample}^2(\tilde{m} \mid \mathcal{D})_n = \tilde{m}$ for all $n \in N$.
Comment 1: If m' is not unique, then a provider-internal procedure runs to pick out the final best choice. For simplicity, in this paper we assume that the provider internal-procedure to rule out optimal preferences among equally desiderable alternatives is the simplest "take it randomly."
Comment 2: If the game ends in failure, then there is a different behavior by

[2]Variants of the game are obtained by requiring the players to exchange objects and to communicate explicitely the reinforcement parameter (Step 0).

[3]The model here presents a way of generalization to linguistic analysis, for example by parsing. In other words, the identity between words required at this step of playing may be replaced by an appropriate linguistic equivalence, whose definition, however, is out of the scope of this paper.

[4]In the terminology of infinitely repeated games, we consider one-shot playing of a game to be equivalent to playing the first repetition of the infinitely repeated version of the game. Here, we denoted such first repetition with "0."

[5]For the sake of simplicity, we omitted time-dependent decorations like supscripts on $\mathcal{L}_2$. However, it is important to emphasize that language evolves along time.

the seeker as if the game would succeed. Someone could then wonder whether this asymmetry is truly motivated, expecially in the scenario of the web services. According to the protocol, in fact, communication between the agents is pairwise, linguistic and extra-linguistic. In particular, both linguistic queries and objects samples are eventually played by the seeker to the provider. For example, the seeker gives the provider some documents or URL links related to the meaning he is referring to when doing a linguistic query. This would happen in case of game failure, for instance because of the misunderstanding of the query by the provider. Two special words "True" and "False" are allowed. These special words are assumed to be universally understood by the players in the game and provide the seeker with the way to communicate to the provider the feedback about the meaning he used to interpret the query.

Step 3. The seeker inputs $\tilde{m}' \subseteq \mathcal{D}$. (If the seeker inputs "?" see the adaption mechanism in Step 2.) Two cases arise.

Case 1. Suppose that $\mathsf{Sim}^1(m, \tilde{m}')$ is true.[6] Then the seeker outputs True—the seeker publicly signals agreement. The game ends in success. Observe that it is not necessary for the seeker to compute $\mathsf{Sim}^1(\mathsf{Sample}^1(m \mid \mathcal{D})_0, \tilde{m}')$ rather than $\mathsf{Sim}^1(m, \tilde{m}')$, since we assumed that the seeker "knows" m in full. So, using a sample of m (at step 0) is not easier from the standpoint of the seeker than using the whole meaning m.

Case 2. Suppose that $\mathsf{Sim}^1(m, \tilde{m}')$ is false. Then the seeker outputs False—the seeker publicly signals disagreement. The game ends in failure.

5.2 Preferences Updating Protocol

Both game success and game failure imply learning by updating. The procedures we present are based on simple positive reinforcement of successful moves and on negative reinforcement of competing associations $\langle\, word\,, meaning\,\rangle$. The players' preferences are updated according to the following cases (see Algorithm 2 below). Recall that $\mathsf{Pref}^a(,,t)$ refers to the player a utility function at time $t \in N$, w is a word, and m is a meaning.)

Suppose that the *nth* round of the repeated game has been played.

Case 1. $\mathsf{Sim}^1(m, \tilde{m}')$ is true. Then the seeker increases the value of her preferences over the winning association $\langle\, w\,, m\,\rangle$ she played by a fixed amount $\rho \in N$ (the "reinforcement parameter" of the game), namely, $\mathsf{Pref}^1(w, m, n) = \mathsf{Pref}^1(w, m, n-1) + \rho$. At the same time n, the seeker decreases by ρ her payoff over competing associations, namely, $\mathsf{Pref}^1(\hat{w}, m, n) = \mathsf{Pref}^1(\hat{w}, m, n-1) - \rho$ for every pair of the form $\langle\, \hat{w}\,, m\,\rangle$, $\hat{w} \neq w$, in the seeker's lexicon Lx_1. The seeker's motivation to adapt preferences is to play a different word for the same meaning in future playing against the same provider. Similarly, the provider increases the value of his preferences over the winning association $\langle\, w\,, m'\,\rangle$ he played by the same amount ρ, namely, $\mathsf{Pref}^2(w, m', n) = \mathsf{Pref}^2(w, m', n-1) + \rho$, and decreases by the same amount ρ the value of his preferences over competing associations: $\mathsf{Pref}^2(w, \hat{m}, n) = \mathsf{Pref}^2(w, \hat{m}, n-1) - \rho$ for every pair of the form $\langle\, w\,, \hat{m}\,\rangle$, $\hat{m} \neq m'$,

[6] Recall that m is the meaning for the word played by the seeker at Step 1.

Algorithm 2 Preferences updating.

<u>**proc**</u> *PrefUpdating+*$(\mathcal{G}(\,Sym, \{a\}, \mathcal{D}, \rho\,), t, w, m) \equiv$
$\mathsf{Pref}^a(w, m, t+1) := \mathsf{Pref}^a(w, m, t) + \rho;$ (* winning association *)
<u>**if**</u> $a =$ "`the seeker`"
<u>**then**</u> <u>**foreach**</u> $\langle\, \tilde{w}\,, m\,\rangle \in Lx_a^t$ with $\tilde{w} \neq w$ <u>**do**</u> (* competing associations *)
$\mathsf{Pref}^a(\tilde{w}, m, t+1) := \mathsf{Pref}^a(\tilde{w}, m, t) - \rho$ <u>**od**</u>
<u>**elsif**</u> $a =$ "`the provider`"
<u>**then**</u> <u>**foreach**</u> $\langle\, w\,, \tilde{m}\,\rangle \in Lx_a^t$ with $\tilde{m} \neq m$ <u>**do**</u>
$\mathsf{Pref}^a(w, \tilde{m}, t+1) := \mathsf{Pref}^a(w, \tilde{m}, t) - \rho$ <u>**od**</u>
<u>**fi**</u>.
<u>**proc**</u> *PrefUpdating-*$(\mathcal{G}(\,Sym, \{a\}, \mathcal{D}, \rho\,), t, w, m) \equiv$
$\mathsf{Pref}^a(w, m, t+1) := \mathsf{Pref}^a(w, m, t) - \rho.$ (* wrong association *)

in the provider's lexicon Lx_2. The provider's motivation to adapt preferences is to play a different meaning for the input word in future playing of the game againts the same seeker.

Case 2. $\mathsf{Sim}^1(\tilde{m}, \tilde{m}')$ is false. Then the players decrease the value of the preferences over their selected associations. In particular, the following updating are computed. For the seeker, $\mathsf{Pref}^1(w, m, n) = \mathsf{Pref}^1(w, m, n-1) - \rho$. For the provider, $\mathsf{Pref}^2(w, m', n) = \mathsf{Pref}^2(w, m', n-1) - \rho$.

6 Discussion

By the foregoing protocols of communication and preferences updating, a shared, preferred lexicon eventually emerges from a system of agents as the result of an "infinite horizon" repeated game. The idea of infinitely repeated advertising game is that players will play the same basic game $\mathcal{G} = \mathcal{G}(\,Sym, \Lambda, \mathcal{D}, \rho\,)$ over and over again (see Algorithm 3 below). Each player bases his next move on the prior history of the game to that point. The history in a repeated advertising game is recorded as lexicon evolution and preference updating.

For a repeated advertising game whose players behave according to the communication and preferences updating protocols we have presented so far, categorial indeterminacy of advertising and search is minimized. Although we have omitted a full theoretical development to support our claim in this paper, we have shown in some detail how we conceive the real setting of an experimental scenario. Experimental work along the direction presented in this paper is available in preliminary form [AA03b], where we have continued the study of the application scenario presented in Section 3.

If an advertising game $\mathcal{G}$ is repeatedly played the resulting infinitely repeated game models a genuine limiting process, say in the spirit of inductive inference [JORS99]. The seeker's feedback might never imply the correct matching of the samples played by the two players at each step of the game history, and the use of game history becomes fundamental. But success in our infinitely repeated adver-

Algorithm 3 An "infinite horizon" repeated advertising game.

foreach $a \in \Lambda$ **do** (* initialization *)
$\mathcal{L}_a^t := \mathcal{L}_a$;
$\mathcal{C}_a^t := \mathcal{C}_a$;
$Lx_a^t := Lx_a$ **od**;
$t := 0$;
Take $i \in \Lambda$ s.t. $Lx_i^t \neq \emptyset$;
$i :=$ "the seeker";
for $t = 0$ **to** $t = l$ **do** (* repeated playing starts *)
foreach $j \in \Lambda \setminus \{i\}$ **do**
$j :=$ "the provider";
$OneShotGame(\mathcal{G}(\,Sym, \{i, j\}, \mathcal{D}, \rho\,), t)$ **od** **od**.

tising games may nevertheless be possible.

In Steels' guessing games, words are associated with *single objects*. In contrast, we associate words with *sets of objects*, namely, meanings. As an important consequence, an advertising game captures a guessing game under the constraint of "complete sampling" of the target meaning at each step of the game. More precisely, this means that if $\tilde{m} = \mathsf{Sample}^i(m \mid \mathcal{D})_t$ is equal to m for each $t \in N$, then for every guessing game there is an equivalent advertising game with the same effects on language and lexicon of the game players. The converse is false.[7]

Related to the previous remark is the cardinality of a single-step sampling $\tilde{m}$, that is, the number of objects in the game domain eventually played at some time by the provider in response to the seeker's query. In Steels' work on guessing games, the learning feedback at some time t in the game history about the hearer's (hidden!) choice of meaning m in a context $\mathcal{D}$ is, when traslated into our framework, $\tilde{m} = \mathsf{Sample}^2(m \mid \mathcal{D})_t$. The strong hypothesis by Steels is that $\tilde{m} = m$. Of course, in the case of a meaning m with very high cardinality, such hypothesis is at least uncomfortable (ineffective). So, our approach differs from Steels' with respect to the ability we allow the players to sample a meaning, in order to obtain a *strictly proper* subset of the target meaning at ever step in the game history (i.e., $\tilde{m} \subset m$) and, possibly, a set of minimal cardinality.

As we have seen, in Steels' guessing games, the feedback allows the hearer to learn, and it implies lexicon evolution (in short, the game evolution). The feedback is played by the speaker through a single object o for which a conceptualization (i.e., a distinctive feature set) has been created (by the speaker). Such feedback is *direct*, in the sense that o is assumed to completely explain the hidden conceptualization (i.e., a distinctive feature set) taken by the speaker in order to play an associated, most preferred word in the lexicon. In contrast, we perceive the seeker's feedback to be *indirect*. This means that the "True/False" response by the seeker on the sample $\widetilde{m'} = \mathsf{Sample}(m' \mid \mathcal{D})_t$ refers to elements which do not characterize uniquely the target meaning.

[7] A formal treatment of this topic is out of the scope of this paper. See [AA03a, AA03b].

7 Conclusion

We have argued for the evidence that the proposed model is suitable to capture the dynamics of new services advertising in a community of agents. We have viewed communication as a general way of managing a seeker/provider relationships, and we have applied the resulting model to an information retrieval scenario of the Web services. We have been concerned about how to make the meaning relevant to a seeker's needs usable by a potential provider in order to fulfill the seeker's requests. More precisely, we have studied the problem of how to successfully coordinate a seeker's needs and a provider's ability to fulfill these needs by means of lexicon sharing. How can the agents establish a common language semantics over a shared domain that allows them to communicate reliably? How do a common language eventually emerge by meaning negotiation? Hereby, we have proposed to these related questions an answer important *per se* that might also be useful to address the more general problem of knowledge management. In fact, we strongly believe that knowledge management is, essentially, a process whereby knowledge seekers are linked with knowledge sources, and knowledge is transferred.

To summarise, in this paper we have advanced and discussed a framework suitable to study both theoretical implications and experimentally grounded impact of language evolution and lexicon sharing within an open distributed multi-agent system. In our approach, local meanings of eventually shared linguistic expressions and the positive benefits of the agent cooperation played an important role. We have considered the application scenario of Web services, where we defined two main operations on services that each agent may perform: advertising and search. For these basic actions, we have provided a precise formulation within a game-theoretical setting. We have conceived the problem of advertisement as a matter of sharing a denotational language. As an important consequence of our "advertising games", we have interpreted the problem of knowledge interoperability and management in the light of evolutionary dynamics and learning in games.

In the context of future work, we plan to work the scenario of Web services up into new experiments and theoretical work. The advertising games we have defined and discussed in this paper provide, we believe, a fruitful starting point.

Acknowledgments

We thank Alessandra Zecca for proofreading.

References

[AA03a] A. Agostini and P. Avesani. Advertising games for the web services. Technical report, ITC-irst, Trento, Italy, April 2003.

[AA03b] A. Agostini and P. Avesani. A peer-to-peer advertising game. Technical report, ITC-irst, Trento, Italy, June 2003.

[BB02] M. Bonifacio and P. Bouquet. Una visione distribuita del sapere organizzativo: il ruolo dell' Intelligenza Artificiale. *Sistemi e Impresa*, 653(6), 2002.

[BBMN02] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. KEx: A Peer-to-Peer solution for distributed knowledge management. In *Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-02)*, pages 490–500, Heidelberg, 2002. Springer-Verlag LNAI 2569.

[BBT02] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.

[CCMW01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language. Electronically available at `http://www.w3.org/TR/wsdl`, 2001.

[GHMF01] M. Gudgin, M. Hadley, J.-J. Moreau, and H. Frystyk. SOAP messaging framework. Electronically available at `http://www.w3.org/TR/`, October 2001.

[GZ02] F. Giunchiglia and I. Zaihrayeu. Making peer databases interact: A vision for an architecture supporting data coordination. In *Proceedings of the Sixth International Workshop on Cooperative Information Agents (CIA-02)*, Heidelberg, 2002. Springer-Verlag LNAI 2446.

[JORS99] S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems That Learn - An Introduction to Learning Theory, 2nd edition*. The MIT Series in Learning, Development, and Conceptual Change, v. 22. The MIT Press, Cambridge, MA, 1999.

[Law00] S. Lawrence. Context in Web Search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.

[MBDH02] J. Madhavan, P. A. Bernstein, P. Domingos, and A. Y. Halevy. Representing and reasoning about mappings between domain models. In R. Dechter, M. Kearns, and R. Sutton, editors, *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02) and of the 14th Innovative Applications of Artificial Intelligence Conference (IAAI-02)*, pages 80–86, Menlo Park, CA, 2002. AAAI Press/The MIT Press.

[MSZ01] S. A. McIlraith, T. C. Son, and H. Zeng. Semantic Web Services. *IEEE Intelligent Systems*, 16(2):46–53, 2001.

[NPSR99] R. Navarro-Prieto, M. Scaife, and Y. Rogers. Cognitive strategies in web searching. In *Proceedings of the Fifth Conference on Human Factors & the Web (HFWEB-99)*. NIST, 1999.

[OR94] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, Cambridge, MA, 1994.

[RB01] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[SBMZ03] L. Serafini, P. Bouquet, B. Magnini, and S. Zanobini. An algorithm for matching contextualized schemas via SAT. Technical Report 0301-06, ITC-irst, Trento, Italy, January 2003.

[SK99] L. Steels and F. Kaplan. Situated grounded word semantics. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 862–867, San Francisco, CA, 1999. Morgan Kaufmann.

[SKML02] L. Steels, F. Kaplan, A. McIntyre, and J. Van Looveren. Crucial factors in the origins of word-meanings. In A. Wray, editor, *The Transition to Language*, pages 214–217. Oxford University Press, Oxford, UK, 2002.

[Ste96a] L. Steels. Perceptually grounded meaning creation. In Y. Demazeau, editor, *Proceedings of the Second International Conference on Multi-agent Systems (ICMAS-96)*, pages 338–344, Los Alamitos, CA, 1996. IEEE Computer Society.

[Ste96b] L. Steels. Self-organizing vocabularies. In C. Langton and T. Shimohara, editors, *Proceedings of the V Alife Conference*, Cambridge, MA, 1996. The MIT Press.

[Ste98] L. Steels. The origins of syntax in visually grounded robotic agents. *Artificial Intelligence*, 103:133–156, 1998.

[Ste01] L. Steels. Language games for autonomous robots. *IEEE Intelligent Systems*, 16(5):16–22, 2001.

[Vap98] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.