

Shared Lexicon for Distributed Annotations on the Web

Paolo Avesani
ITC-irst
via Sommarive 18
38050 Povo, Italy
avesani@itc.it

Marco Cova
ITC-irst
via Sommarive 18
38050 Povo, Italy
cova@itc.it

ABSTRACT

The interoperability among distributed and autonomous systems is the ultimate challenge facing the semantic web. Heterogeneity of data representation is the main source of problems. This paper proposes an innovative solution that combines lexical approaches and language games. The benefits for distributed annotation systems on the web are twofold: firstly, it will reduce the complexity of the semantic problem by moving the focus from the full-featured ontology level to the simpler lexicon level; secondly, it will avoid the drawback of a centralized third party mediator that may become a single point of failure.

The main contributions of this work are concerned with (1) providing a proof of concept that language games can be an effective solution to creating and managing a distributed process of agreement on a shared lexicon, (2) describing a fully distributed service oriented architecture for language games, (3) providing empirical evidence on a real world case study in the domain of ski mountaineering.

Categories and Subject Descriptors

C.2.4 [Distributed Systems]: Distributed Applications;
H.3.5 [Online Information Services]: Web-based Services

General Terms

Design

Keywords

Emergent Semantics, Distributed Annotations, Language Games, Interoperability

1. INTRODUCTION

The interoperability among distributed and autonomous systems is the ultimate challenge facing the semantic web. The open issue is how to preserve the requirement of locality for representations, while at the same time enabling an effective interaction among autonomous peers.

The heterogeneity of autonomous representations is the source of the problem. Strong effort is usually spent ar-

ranging well defined ontologies whilst neglecting the issues related to their wide spread use.

In our work we conceive the problem of interoperability as the problem of supporting the emergence of a common lexicon among a community of peers. A shared lexicon enables a common denotation that allows different peers to refer to the same object using the same label. Building a shared lexicon does not necessarily require peers to share the underlying schemas designed to represent objects.

Recent studies in language evolution have developed the notion of language games. A language game is a computational model that allows convergence on a shared lexicon in a fully distributed framework. The key idea behind this model is that a shared lexicon emerges from adaptive pairwise interactions between language users and continues to evolve and adapt through repeated interactions.

Differently from the usual solution promoted by the semantic web where the official vocabulary or lexicon is preserved by a third party, usually a consortium for standard preservation, in the language game model the shared lexicon is fully distributed amongst all the peers. The main advantage is a distributed system without a single point of failure.

The original contribution of this paper is concerned with a proof of concept that language games can be an effective solution to the interoperability problem among heterogeneous and autonomous systems. In particular, we address a specific application problem that occurs in distributed annotation systems. As an example of such applications we refer to a real world case study in the domain of ski mountaineering.

The paper includes a discussion of related works, a brief introduction to the language games, a service oriented architecture to enable the interoperability among distributed annotation systems and finally the empirical results of a case study based on three real world web sites devoted to ski mountaineering.

2. SKI MOUNTAINEERING

Ski mountaineering is a very exciting outdoor activity. In ski mountaineering, both the ascent and descent of a peak are made entirely on skis, using climbing skins and perhaps ski crampons for traction on the ascent, and then descending a continuous ski route back down to the base. This sport can be very risky. Avalanches represent an ubiquitous hazard that may arise from an erroneous situation assessment.

To prevent or reduce the avalanches hazard, it is a common practice for ski mountaineers to share their experiences on the web. The typical behaviour of ski mountaineers is to

collect, the day before a ski trip, all the annotations from on-line diaries on the ski routes of interest. When the ski route is accomplished, the ski mountaineers note in their diaries the up-to-date conditions of the route.

For ski mountaineering in the Alps there are many web sites, among them *skirando*¹, *gulliver*², *moleskiing*³. These are all organized along the same pattern: a catalog of ski routes and a collection of individual diaries of ski trips where a diary entry describes a ski trip that refers to a ski route in the catalog. In such a way, given a ski route, it is straightforward to retrieve the most recent annotations, i.e. the related diary entries.

The scenario above is much more effective as many ski trips reports are collected. However, although it may appear counterintuitive, the proliferation of web sites devoted to ski mountaineering doesn't necessarily increase the accessibility of ski trip diaries. This is because a side effect of the increasing number of web sites is the partition of the ski mountaineers into smaller communities that refer to heterogeneous ski route catalogs.

3. DISTRIBUTED ANNOTATIONS

The example above is a particular instance of a more general scenario that can be referred to as distributed annotations. The habit of allowing end users to provide their opinions of a given item as annotations, e.g., about books, movies, hardware, is ubiquitous.

Blog oriented architectures for annotations introduce the notion of aggregation services, third party servers in charge of indexing all the annotations with respect to a given category of items. Such services allow a user to obtain all the recent blog annotations about a given item on demand.

Of course the tacit assumption of aggregation services is that the annotated items have to be indexed using a common reference system. While for books ISBN provides a straightforward solution, more often an agreement on a common reference system does not exist.

Catalogs are usually distributed and autonomously designed. Even though they are concerned with the same category of items the heterogeneity of representations is pervasive. For this reason aggregation services need to be paired with alignment services that enable the mapping among different catalogs.

Catalogs alignment can be conceived as a typical problem of interoperability that precludes having a fully distributed annotation system. In the remaining part of the paper we will refer to the domain of ski mountaineering but it is worthwhile to note that the pattern of solution is independent from the specific application scenario.

4. RELATED WORKS

The problem of catalog alignment has been approached by many initiatives in the context of the Semantic Web. The usual strategy of these efforts consists in establishing a relationship between the local representations and a common reference encoding, namely a shared ontology. This approach requires two steps: (1) the definition of an ontology for the specific domain, (2) the definition of a mapping between a local representation and the shared ontology. While

¹<http://www.skirando.ch/>

²<http://www.gulliver.it/>

³<http://www.moleskiing.it>

intuitive, this approach is often not effective in practice. In fact, the first step raises the question of who is in charge of managing the shared ontology and also, the mapping step is generally far from being trivial and too often requires manual intervention.

Designed around this general schema, a number of initiatives have arisen, fostered by the availability of machine processable semantics expressed in meta-models such as [14, 8]. One of the most recent examples is the European project Harmonise [9, 10]. In Harmonise, a mediator is in charge of managing a shared representation of tourism concepts, while subscribers have to map their local encodings with respect to the predefined ontology. To confirm the drawbacks of this approach, it is interesting to note that Harmonise, after the conclusion of the project, is currently dealing with the problem of establishing a consortium and the related sustainability plan. Furthermore, the mapping task is at present performed manually.

Recently research is focused on the issues specifically related to the mapping between schemas [4, 11, 12, 13] or shallow representations like taxonomies [7, 22]. The automation of the mapping process would enable pervasive interoperability without the constraint of a mediator [6], that in distributed applications becomes the "single point of failure". In such a scenario, each peer would manage autonomously the mapping with respect to the other peers. Unfortunately, automatic schema matching has proven to be a complex problem and a general solution is still not available. A further problem lies in the fact that this approach does not scale well when applied to a fully distributed environment. In fact, while in centralized approaches like Harmonise only one lexicon is needed to map the local reference denotation with respect to a global one, a strategy based on pairwise mappings has a quadratic complexity with respect to the number of peers.

Gossiping algorithms [1] have been proposed to reduce the scaling problem. In this approach a (partial) solution to the schema mapping is provided only for a small portion of the complete peers set. The undefined mappings are derived through a transitive exploration of a peer's neighborhood. Although this strategy is promising, the drawback of providing manual mappings prevents a full automation of the process of interoperability.

Alternative solutions have been proposed that aim to reduce the complexity of the mapping problem by moving from the schema to the object level [21]. Object mapping differs from schema mapping. In object mapping the assessment is performed looking at attribute values of the objects. Of course, such an approach is less general than schema mapping. A good schema mapping makes possible to derive all the correct mappings for all the objects. On the contrary, the performance of an algorithm working at the object level is affected by the specific attribute value distribution.

A final research trend that is receiving increasing interest looks at the problem of interoperability in terms of a shared lexicon. A W3C initiative, SKOS [16], has been organized to explicitly manage a shared vocabulary for concepts denotation. Similarly to Harmonise, SKOS relies on the restrictive hypothesis that autonomous peers will subscribe to such a common vocabulary.

The fundamental question, then, is how to exploit the benefits of a single lexicon for each peer without incurring the restrictions of a centralized global reference that is con-

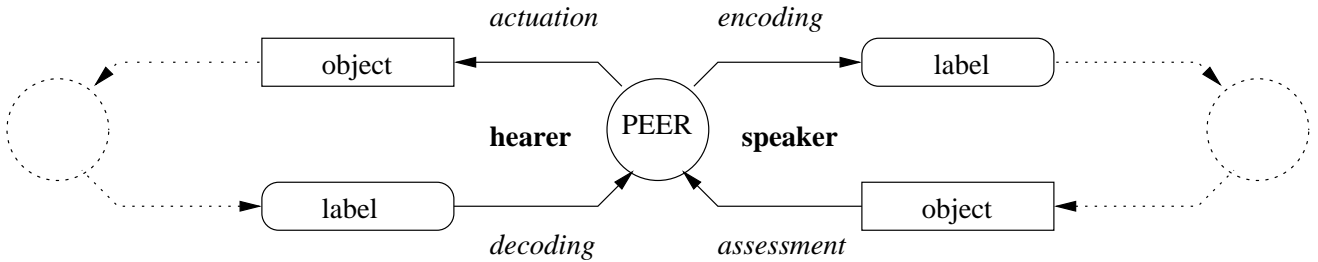


Figure 1: The pairwise roles of a language game.

trolled by a third party. We give a solution to this question by leveraging the approach of *language games*. With language games [19, 17], the language (or the ontology, the lexicon) of a community of peers is considered a system that emerges from adaptive interactions among peers. Thus, in contrast with the techniques we have described before, each peer starts with a preliminary hypothesis of lexicon and the challenge is to enable a process of interactions that brings the peers’ lexica to converge on a single common denotation system. The language games approach has been validated by simulations in different domains, e.g., bookmark taxonomies [2], robot communications [18].

In the following we will show how language games can be exploited to deal with the problem of supporting the emergence of a distributed common lexicon without the constraint of a centralized third party mediator. We first introduce the basic notions of language games with reference to the ski mountaineering domain and then we describe a service oriented architecture developed to enable the interoperability among the ski mountaineering web sites of the Alps.

5. LANGUAGE GAMES

There are many variations of language games. In the following we will focus our attention on a specific model known as *naming games* [20]. A naming game is defined by a set of peers \mathcal{P} (the game players), a set of objects \mathcal{O} (the heterogeneous representations of the ski routes), and a set of labels \mathcal{L} (the candidate names to denote the ski routes). A peer $p \in \mathcal{P}$ is then defined as a pair $p = \langle \mathcal{L}_p, \mathcal{O}_p \rangle$.

Each peer $p \in \mathcal{P}$ has its own lexicon drawn from the Cartesian product $\mathcal{L}_p = \mathcal{O}_p \times \mathcal{L}_p \times \mathcal{N} \times \mathcal{N}$, where \mathcal{O}_p are the objects referenced by p , \mathcal{L}_p is the local vocabulary of p , and \mathcal{N} are the natural numbers used to represent the strength of the association between \mathcal{O}_p and \mathcal{L}_p . The lexicon may include synonymous labels, where two labels are associated to the same objects, and homonymous labels, where the same label is associated with two different objects.

The following table illustrates a sample lexicon. From the table, for example, we can see that the association between object o_1 and label l_1 has been successfully used 8 times in 10 different language games, while the association between o_1 and label l_2 has been successful only once in 8 games.

The ultimate goal of the game is to bring the local lexica of the peers towards the same association structure. If all the peers converge to the same label to denote the same object the lexica will enable effective communication among peers.

\mathcal{O}_p	\mathcal{L}_p	Times used, u	Times successfully used, a
o_1	l_1	10	8
o_1	l_2	8	1
o_2	l_1	5	5
o_3	l_3	10	7

A naming game involves an iterative process based on pairwise sessions. The basic interaction involves two peers with different roles: speaker and hearer, and so a session of communication is not symmetric. Nevertheless each peer can play different roles in different sessions.

The interaction proceeds as follows (see Figure 1). First the speaker p_s selects an object $o_s \in \mathcal{O}_s$ from its set of objects, and encodes o_s using a label l_j . The label is chosen according to the preferences expressed in the current version of the local lexicon \mathcal{L}_s (local to speaker p_s). The encoding of object o_s is obtained by looking at the most successful label. A label l_j is more successful than a label l_k iff $\langle o_s, l_j, u_j, a_j \rangle \in \mathcal{L}_s$, $\langle o_s, l_k, u_k, a_k \rangle \in \mathcal{L}_s$, $u_j \geq u_k$ and either $a_j/u_j > a_k/u_k$ or $a_j/u_j = a_k/u_k$ and $u_j > u_k$, where u_j represents how many times the label l_j has been used and a_j represents how many times there was an agreement on label l_j with other peers. In case of a tie, a random choice is performed.

The hearer p_h decodes the label l_j and retrieves the associated object, $o_h \in \mathcal{O}_h$, by looking at its own lexicon \mathcal{L}_h . The actuation step is in charge of sending the object o_h to the speaker p_s .

The last step is concerned with assessment. The speaker has to verify that the object received from the hearer is the same as that selected at the beginning of the communication session.

If the object referred to by the hearer is the same as selected by the speaker, both of them positively reinforce their lexica by updating the corresponding label-object association as follows: $\langle o_s, l_j, u_j + 1, a_j + 1 \rangle \in \mathcal{L}_s$ and $\langle o_h, l_j, u_j + 1, a_j + 1 \rangle \in \mathcal{L}_h$. If the hearer replies with a different object $o_s \neq o_h$, it means that the communication failed and the peers’ lexica is negatively reinforced by only increasing the counters of lexical relation (while the counters of agreements on the lexical relation remain the same): $\langle o_s, l_j, u_j + 1, a_j \rangle \in \mathcal{L}_s$ and $\langle o_h, l_j, u_j + 1, a_j \rangle \in \mathcal{L}_h$.

The critical point of the game is the assessment step. Objects can refer to heterogeneous representations of the same concept and therefore the assessment step needs to be carefully implemented. Looking at our example, objects can be different instances of different schemas that refer to the same ski route. An assessment strategy could exploit the mapping between the two schemas, but as we have seen before this

task is too complex. An alternative strategy is to assess the equivalence by looking directly at the data. In Section 7 we will provide the details of the implementation choice for our case study on ski mountaineering.

6. SYSTEM ARCHITECTURE

In the previous sections we have defined the problem space that we are exploring and the language games-based approach we leverage. In this section, we will illustrate the software component that we have designed and implemented to concretely realize our technique. We defer to the next section the discussion of how this component is used in a specific scenario, namely the domain of ski mountaineering.

6.1 Requirements

The reference scenario that drove the design task is similar to the one presented in section 2. Precisely, we assume that our solution is to complement a network of web applications each providing a legacy catalog system and an annotation service. The catalog system is a repository of representations of items (objects in the terminology of section 5), e.g., books, ski routes, bookmarks. The annotation service collects users’ reviews on cataloged objects. While the systems we are targeting appear to be fairly distinct, they are instances of a more general type of application that has a private catalog of objects and can share references to these objects. Such systems are indeed very common on today’s Web, with Amazon and Epinions being typical examples.

The main functional goals of our component are to allow the alignment of catalogs and to support the distributed aggregation of annotations created by related applications. The basic requirements that we identified consist in preserving the heterogeneity, the autonomy and the robustness to evolution of these applications. To support heterogeneity, we require only the minimal set of architectural constraints and allow for alternative implementations of various parts of our component. We guarantee autonomy by avoiding strong or centralized coordination among individual systems. Finally, we explicitly take into account that the system is inherently dynamic and subject to change.

In order to employ the language games technique in this scenario, we needed to realize and integrate a distributed implementation of the language games model. Some implementations of language games already exist, e.g., McIntyre in [15] describes a testbed to set up, control and visualize language games simulations. However, to our knowledge, all the available implementations consist of stand-alone simulators that lack the distributed nature of the model and do not allow it to be deployed in a real world setting. We further require that the implementation of the language games be flexible, to easily implement possible variants to the model, and easy to program and test.

Lastly, we wanted to minimize the cost of augmenting existing solutions with our component. This requirement translates into realizing a component that can be transparently plugged into an existing system, requiring as few modifications as possible to its legacy parts.

6.2 Implementation

Figure 2 represents the architecture of our component as it was shaped by the requirements described above. The *Application Server* represents the generic legacy systems we are focusing on; *DiAGRA* (*Distributed AGgRegator of Annota-*

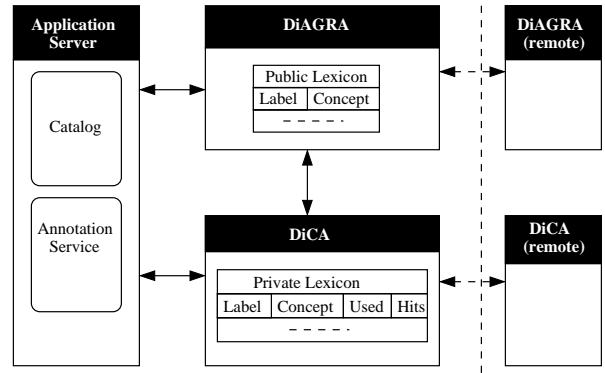


Figure 2: The architecture of the component.

tions) is the module in charge of implementing the aggregation functionalities; *DiCA* (*Distributed Catalog Alignment*) realizes the catalogs alignment feature.

6.3 DiAGRA

The DiAGRA module decouples inter-peer communications from the details of how each peer’s catalog is organized.

To do so, it maintains the current *public lexicon*, a dictionary that maps local objects to labels. All the communication is proxied through DiAGRA which performs this mapping task. DiAGRA uses this lexicon to enhance the aggregation primitives of trackbacking and crawling. Trackbacking refers to the broadcasting of new local annotations to all peers in the federation: it is used to “push” annotations to remote sites. Conversely, the crawling facility allows a peer to “pull” annotations from remote peers. Figure 3 describes the flow of operations performed when a new annotation is inserted by a local user. Figure 4 represents how remote annotations are retrieved. Note how the translation operations performed by DiAGRA are transparent to the Application Server.

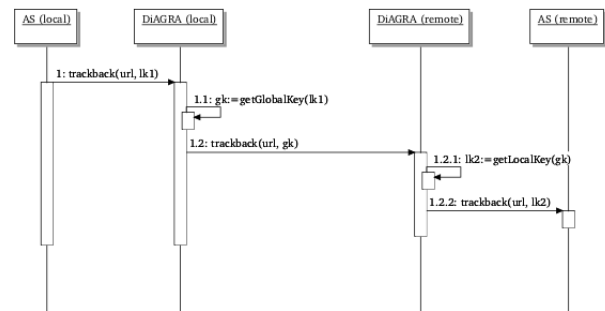


Figure 3: Sequence diagram for the trackback operation.

In the trackback scenario, when a user posts a new comment with respect to a local item, the Application Server notifies DiAGRA of the new annotation available. DiAGRA looks up its lexicon and retrieves the label associated with the annotated object. It substitutes this label for all the ref-

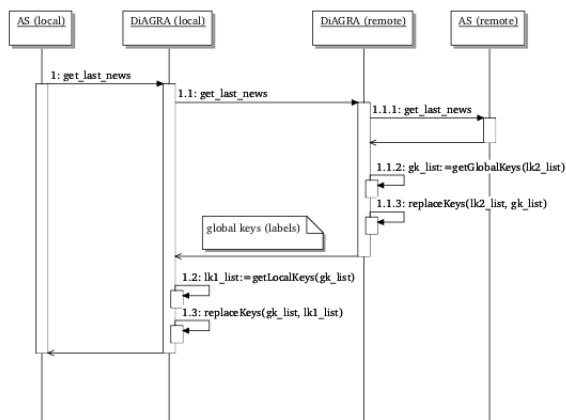


Figure 4: Sequence diagram for the crawling operation.

ferences to the local item in the annotation and then sends the annotation to the other peers in the federation. A peer receiving the annotation performs the opposite translation, from the label to the local item. The translated annotation, now containing only local references, is made available to local users as if it was produced locally.

In the crawling scenario, a peer asks remote peers to provide the most recent annotations inserted by their users. At the remote peer, the crawling request is handled by the DiAGRA component. It fetches the annotations from the Annotation Server, translates references to local objects to the associated labels defined in the lexicon, and sends back the translated annotations. The first peer is now able to perform the opposite translation and presents the received annotations to local users.

The translation processes that we have described may fail in two cases: when there is no mapping available for an object - meaning that it is not shared by other peers of the federation - or when a received label has no mapping - that is, it is encoding an unknown object. In both cases, DiAGRA simply drops the faulty annotations.

In summary, DiAGRA makes possible to dynamically parametrize communication with a desired lexicon. Therefore, it solves the problem of how an Application Server, given a lexicon common to a community of peers, can start using the lexicon in its communications without modifying the legacy catalog. Next we will see how such a common lexicon can be provided. This is the task of the DiCA module, described in the following section.

6.4 DiCA

The DiCA module encapsulates the language games technique by implementing the model outlined in section 5.

DiCA's primary task is to use the language games approach to adaptively build and refine a common lexicon. Specifically, its responsibilities are threefold. Firstly, it defines the choreography of the distributed system, by specifying the possible inter-peer communication methods. This is achieved by providing primitives to send a peer a label, a label and a set of objects, a set of objects or a feedback message. Secondly, it stores a variety of implementations

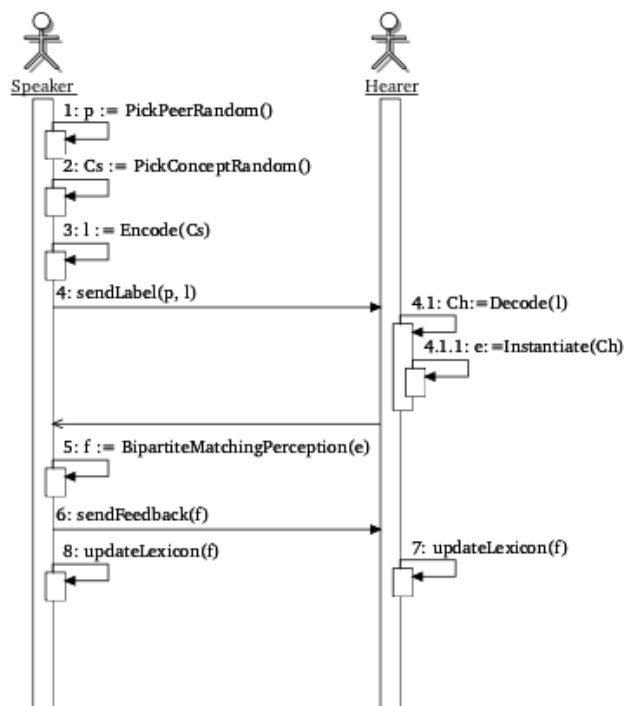


Figure 5: A language game.

for each step of a language game. For instance, it contains different primitives to select the next peer to play with, such as randomly selected or in a round robin fashion. Finally, it provides the mechanisms to create and run game strategies, i.e., to express how the communication and game primitives should be combined to play a language game.

Figure 5 shows two DiCA components engaged in a language game interaction. In this case, the choreography consists of the exchange of three messages: the speaker sends the hearer a label; the hearer sends an object to the speaker; the speaker sends the result of the game to the hearer. Each self-message in the diagram represents the invocation of a primitive operation. By selecting different primitives one can define new strategies. For example, a strategy where peers are contacted in a round robin fashion can be obtained by substituting the invocation of the *PickPeerRandom* primitive with the *PickPeerRoundRobin*.

A complete description of the DiCA module is out of scope of this paper. We invite the interested reader to refer to [3] for further details.

The decomposition of the system into DiAGRA and DiCA allows us to keep clearly separated the tasks of building a common lexicon, which is performed by DiCA, and the task of using the lexicon, performed by DiAGRA, to provide interoperable services. The main functional effect of this separation is that the stable lexicon is less sensitive to fluctuations in the process of agreeing on a common lexicon.

From the technological point of view, the interfaces between the Annotation Server, DiAGRA and DiCA are REST interfaces. Communication between peers is performed using web services technology.

7. A REAL WORLD APPLICATION

The architecture we have presented in the previous section is general enough to fit many different scenarios. However, in order to practically assess it, we grounded the architecture in the ski routing scenario presented in section 2.

The web applications we are targeting are gulliver, skirando and moleskiing. Each of these web sites is the center of a ski mountaineering community to which it offers the services of a ski route catalog and a ski trip annotation list.

This scenario constitutes a fine example of the semantic interoperability problem: because these communities are completely autonomous and heterogeneous, they use different schemas to describe ski routes and denote the same routes using different names. As a consequence, annotations on trips performed along the same routes cannot be shared among communities. Furthermore, in the ski mountaineering domain there is currently no effort leading to the formation of a shared ontology nor is it foreseeable in the future. Thus, it represents an ideal scenario for the application of the language games approach.

Let us, then, examine how the advertising model maps to the ski mountaineering domain. Ski mountaineering web sites play the role of peers and ski routes map to objects. They are private to each peer, in the sense that a peer is free to model a route according to the schema it prefers. The role of objects is played by concrete representations of ski route models. A convenient way to represent a ski route is to provide an XML linearization of the information available for the route. The following are the linearizations for the same route as modeled by two different ski mountaineering web sites:

```
<route>
  <id_route>3002</id_route>
  <top>Altissimo di Nago</top>
  <route>da S.Giacomo</route>
  <area>Trentino</area>
  <municipality>MORI</municipality>
  <valley>
    Valle dell'Adige (Alto Garda - Baldo)
  </valley>
  <difficulty>MS</difficulty>
  <exposure>SE</exposure>
  <start_height>1150</start_height>
  <top_height>2078</top_height>
  <gap>930</gap>
  <start_point>S. Giacomo</start_point>
</route>

<route>
  <id>3940</id>
  <top>Altissimo di Nago</top>
  <region>Adamello</region>
  <title>Da San Giacomo</title>
  <global_difficulty>AD</global_difficulty>
  <ski_difficulty>S4</ski_difficulty>
  <base_height>1194</base_height>
  <top_height>2079</top_height>
  <gap>900</gap>
  <exposure>E</exposure>
</route>
```

A typical use case scenario is the following. The DiCA module on moleskiing is activated and starts playing lan-

guage games with gulliver and skirando. When the shared lexicon starts emerging, a snapshot of the lexicon is provided to DiAGRA. At this point, DiAGRA holds a mapping for some of the routes that are common to other web sites. For these route, moleskiing can retrieve and present to its users annotations inserted on gulliver or skirando, and conversely share annotations produced locally.

The current prototype uses Tomcat and the axis toolkit to support the SOAP protocol, and Java for the DiAGRA and DiCA implementations. The language used to define game strategies is BPEL4WS, a composition language normally used to perform web service orchestration. We chose it for its built-in coordination features, relatively high level of abstraction and the availability of tools to do quick, graphical programming.

8. EXPERIMENTAL RESULTS

We are currently testing the system locally before deploying it on the real web sites. A number of factors influence the outcome of a game. Previous works on language game simulations show that the number of peers, the cardinality of the object and label sets of each peer, the mutual overlapping of the object sets and the particular strategy adopted to play a game are critical factors.

Our test bed has the following setup. It reproduces the federation composed of the skirando, gulliver and moleskiing web sites. The following table summarizes the main characteristics of the datasets we are using:

	<i>gulliver</i>	<i>moleskiing</i>	<i>skirando</i>
Total items	38	179	69
gulliver overlap	-	22 (12%)	8 (11%)
moleskiing overlap	22 (57%)	-	51 (73%)
skirando overlap	8 (21%)	51 (28%)	-
Complete overlap	6 (15%)	6 (3%)	6 (8%)

The overlapping objects in the test datasets were found manually.

In regards to the details of peers interactions, we are experimenting with a number of basic language game strategies. An especially critical step in the game is the assessment task, in charge of evaluating whether two route linearizations represent the same ski route. In the current setup, the DiCA primitive that performs this step uses a string comparison technique based on a bipartite matching algorithm. It works as follows: The linearizations are divided in tokens and schema information is dropped. This leaves two sets of tokens, in our example $\{3002, \textit{Altissimo di Nago}, \textit{da S.Giacomo}, \textit{Trentino}, \textit{MORI}, \textit{Valle dell'Adige (Alto Garda - Baldo)}, \textit{MS}, \textit{SE}, 1150, 2078, 930, \textit{S. Giacomo}\}$ and $\{3940, \textit{Altissimo di Nago}, \textit{Adamello}, \textit{Da San Giacomo}, \textit{AD}, \textit{S4}, 1194, 2079, 900, \textit{E}\}$. A bipartite graph matching algorithm is then used, given a distance function, to find the optimal matching of tokens. Working on attribute values overcomes the drawback of high variance in schema design by taking advantage of the redundancy of the data values. The fundamental assumption underlying this assessment method is that different representations of the same object share a significant part of their textual content and that, on the contrary, the contents of different objects are significantly different. For example, in our sample routes, the tokens “Altissimo di Nago” and “Giacomo” are present in both representations. Of course, this method is not applicable

where the assumption does not hold. It is important to stress, however, that the assessment primitive, like all the other game primitives, is a parameter of a peer’s strategy. In other words, peers are free to choose the assessment method that best suits the characteristics of objects representation.

The primitive then, given an object, ranks local routes according to the matching weight. On our data, this simple method effectively detects the correct matching, if one exists. If a matching does not exist, it still provides an answer, which can be interpreted as the best approximation for the given example route, i.e., the “nearest” ski route, according to some distance metrics.

We tested the system running various game sessions. In every game, either two or three peers were employed, playing the roles of gulliver, skirando and moleskiing. Every peer contained the routes in common with all the other peers participating in the game. Dynamic modifications to the federation or to individual peers’ object sets will be included in future trials.

Figure 6 shows the plot of four sample game sessions. It shows the percentage of lexica convergence as a function of the number of games played by peers. 0% convergence means that there is no common associations among the peers: every peer is using different labels to encode the same object. Thus, the common lexicon is empty and inter-peer communication will fail. Conversely, 100% convergence indicates that all peers have reached an agreement on how to reference all shared objects. The common lexicon contains one entry for each shared object and thus inter-peer communication is always successful. Translating these findings into our application, 0% convergence implies that no ski routes contained on remote sites have been associated to local routes. Therefore, no annotation produced on a remote site is available on the local site. 100% convergence means that all remote routes have a local correspondent and that all annotations produced in the federations are available to all web sites, independently of the production site.

Some observations are possible about these results. Firstly, all games end in a 100% convergence state. This happens because there is a complete overlapping of the object sets of all playing peers. In a more realistic situation there would be peers equipped with objects that are not common to other peers. In this case, the convergence process would stop before reaching 100%.

Secondly, the increase in the convergence rate is not strictly monotonic. This can be explained as follows: As a result of a series of games among a subset of all peers, an association between an object and a label might be chosen that maximizes communicative success in this subset. However, this association might represent a suboptimal choice at the level of the whole federation. Hence the decrease in the overall convergence percentage.

Manual checking has shown that the routes that end up being denoted by the same label in the shared lexicon are actually the same. However, this result depends essentially on the efficacy of the assessment module: in this case, it was able to correctly identify matching routes.

In tests, the convergence speed is most strongly dependent on the object sets cardinality. The smaller the cardinality, the lower the number of matchings to be established and, consequently, the faster the convergence. With regard to the peer numbers, it is worth remarking that in the test setup, we deal with a very limited federation, composed of either

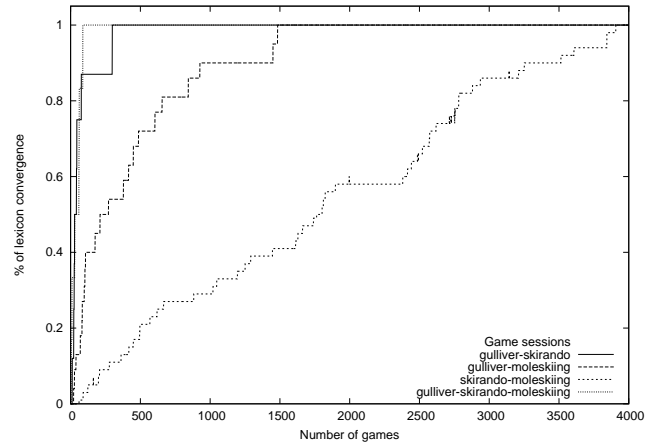


Figure 6: Evolution of common lexicon formation.

two or three peers. Therefore, it is impossible to meaningfully assess the impact of this factor on the lexicon formation. Nonetheless, on the basis of previous simulations, we expect the peers number to also be a key factor in the convergence speed.

9. CONCLUSIONS

In this paper we investigated the problem of building the distributed common reference systems needed to enrich current web applications and allow for their meaningful interoperability. We introduced a novel approach to this problem based on the language games technique. To our knowledge, this paper is the first to describe a general architecture that can be used to deploy the technique in real applications on the web. Lastly, we presented our experience with a concrete example of both the technique and the architecture in the field of ski mountaineering.

There is wide scope for future work. The model underlying the language games technique is still fairly unsophisticated and we plan to use the experience gained from practical experimentation to improve it. Along the same line, we expect to design more refined strategies to guide the games, in order to improve the lexicon building process. The use case we have shown here has some limitations. In particular, there is a one-to-one mapping between objects and their linearization. We plan to test our approach on other, more complete, domains, e.g., the blogosphere.

Finally, it is interesting to observe that the problem we considered in this paper is only one small subset of the larger problem of providing an extension of the current web so that “information is given well-defined meaning, better enabling computers and people to work in cooperation” [5]. It is our hope that the approach we present here, although orthogonal to the ones employed by the mainstream Semantic Web initiative, can be a useful piece in the solution to this larger problem. We are encouraged by the results we present in this paper, since they show that this research trend, even at its beginning, can successfully be applied to real problems in real world scenarios.

10. ACKNOWLEDGEMENT

We would like to thank Roberto Tiella and Arun Sharma for their collaboration to implement the work presented here. Andrew Webb helped us to revise this paper.

This work was partially funded by the DiCA project, thanks to a grant of INRM (Istituto Nazionale Ricerca Montagna) and PAT (Provincia Autonoma di Trento).

11. REFERENCES

- [1] K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. Start making sense: The Chatty Web approach for global Semantic agreements. *Semantic Web Journal*, 1(1), 2003.
- [2] P. Avesani and A. Agostini. A Peer-to-Peer Advertising Game. In M. Orlowksa, M. Papazoglou, S. Weerawarana, and J. Yang, editors, *First International Conference on Service Oriented Computing (ICSOC-03)*, pages 28–42, Berlin Heidelberg, 2003. Springer-Verlag LNCS 2910.
- [3] P. Avesani, M. Cova, R. Tiella, and A. Sharma. A Service Oriented Architecture for Advertising Games. In S. Weerawarana, editor, *RA221:W0411-084 ICSOC'04 (forum)*. IBM, 2004.
- [4] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
- [5] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.
- [6] M. Bonifacio, P. Bouquet, G. Marni, 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)*, volume 2569 of *Lecture Notes in AI*, pages 490–500. Springer Verlag, 2002.
- [7] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In *Second International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 130–145. Springer Verlag, September 2003.
- [8] J. Broekstra, M. C. A. Klein, S. Decker, D. Fensel, F. van Harmelen, and I. Horrocks. Enabling knowledge representation on the Web by extending RDF schema. In *Proceedings of the tenth international conference on World Wide Web*, pages 467–478, 2001.
- [9] M. Dell’Erba, O. Fodor, A. Formica, et al. HARMONISE: Towards Interoperability in the Tourism Domain. In *Proceedings of ENTER Conference on Information Technology and Travel & Tourism*, 2003.
- [10] M. Dell’Erba, O. Fodor, W. Hoepken, and H. Werthner. Exploiting semantic web technologies for harmonising e-markets. *Journal of Information Technology & Tourism (JITT), Special Issue on “Semantic Web Technologies and Applications”*, 2005.
- [11] H. H. Do and E. Rahm. COMA - A system for flexible combination of schema matching approaches. In *Proceedings of the 28th International Conference on Very Large Databases*, pages 610–621, 2002.
- [12] A. Doan, J. Madhavan, P. Domingos, and A. Halvey. Learning to map between ontologies on the semantic web. In *Proceedings of the 11th International Conference on World Wide Web*, pages 662–673, 2002.
- [13] F. Giunchiglia and P. Shvaiko. Semantic matching. *Knowledge Engineering Review*, 18(3):265–280, 2004.
- [14] J. Hendler and D. McGuinness. DARPA Agent Markup Language. *IEEE Intelligent Systems*, 15(6), 2000.
- [15] A. McIntyre. Babel: A testbed for research in origins of language. In *Proceedings of COLING-ACL 98*, Montreal, 1998. ACL.
- [16] SKOS. <http://www.w3.org/2004/02/skos/>.
- [17] L. Steels. Grounding symbols through evolutionary language games. In A. Cangelosi and D. Parisi, editors, *Simulating the evolution of language*, pages 211–226. Springer Verlag, London, 2001.
- [18] L. Steels. Language games for autonomous robots. *IEEE Intelligent systems*, pages 17–22, October 2001.
- [19] L. Steels and F. Kaplan. Bootstrapping Grounded Word Semantics. In T. Briscoe, editor, *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, chapter 3, pages 53–73. Cambridge University Press, Cambridge, 2002.
- [20] L. Steels and A. McIntyre. Spatially Distributed Naming Games. *Advances in Complex Systems*, 1(4), January 1999.
- [21] S. Tejada, C. A. Knoblock, and S. Minton. Learning Object Identification Rules for Information Integration. *Information System*, 26(8):607–633, 2001.
- [22] Y. Tzitzikas and C. Meghini. Ostensive Automatic Schema Mapping for Taxonomy-based Peer-to-Peer Systems. In *Seventh International Workshop on Cooperative Information Agents (CIA 2003)*. Springer-Verlag LNAI, vol. 2782, 2003.