A decentralized approach for establishing a shared communication vocabulary

Jurriaan van Diggelen, Robbert Jan Beun, Frank Dignum, Rogier M. van Eijk, John-Jules Meyer

Institute of Information and Computing Sciences
Utrecht University, the Netherlands
{jurriaan, rj, dignum, rogier, jj}@cs.uu.nl

1 Introduction

A fundamental problem of communication in open multi agent systems is caused by the heterogeneity of the agent's knowledge sources, or more specifically of the underlying *ontologies*. Although ontologies are often advocated as a complete solution for knowledge sharing between agents, this is only true when all agents have knowledge about each others' ontology. The most straightforward way to establish this is to develop one common ontology which is used by all agents. However, this scenario is highly unlikely in open multi agent systems, like those on the internet, as it requires all involved system developers to reach consensus on which ontology to use. Moreover, a common ontology is disadvantageous for the problem solving capabilities of the agents as different tasks typically require different ontologies [3].

To deal with this problem, we deliberately diverge from the definition of an ontology as an explicit specification of a shared conceptualization [5]. We adopt private ontologies, which are not shared, and an intermediate ontology (or interlingua [12]) which is (partially) shared and not explicitly specified. The private ontology of an agent is used for storing and reasoning with *operational knowledge*, i.e. knowledge relevant to a particular problem or task at hand. The intermediate ontology is used for the communication of operational knowledge; we therefore refer to it as *communication vocabulary* (or cv). Communication proceeds by translating from the speaker's private ontology to the communication vocabulary which the hearer translates back again into its own private ontology. Contrary to the agent's private ontology, no operational knowledge is dependent on the communication vocabulary. It can therefore be treated as a *dynamic ontology* [6]. The agent's private ontologies, on the other hand, are *static*, i.e. they do not change over time.

Initially, the communication vocabulary is empty. To enable communication between agents, the cv is built. This raises the question: how is it built? Most approaches that deal with these issues use some centralized service which facilitates in publishing and making decisions about the the communication vocabulary. For example, the FIPA ontology agent [1] publishes information about ontologies and thereby facilitates communication by mediating between heterogeneous agents.

Also, the Ontolingua server [4] is intended as a service to achieve consensus on a shared ontology in heterogeneous groups. However, these techniques are not straightforwardly applicable in systems which lack a clear organization. Consider, for example, the group of agents on the world wide web. If there would be one ontology agent that aligns every ontology in the system, it would be very difficult to make every agent accept its authority. Furthermore, this agent runs a high risk of getting overloaded. If, on the other hand, there would be a number of ontology agents on the web, communication problems would arise between agents that use different ontology agents.

In this paper, we try to overcome these problems by exploring ways to establish a communication vocabulary in a fully *decentralized* way. This means that, instead of making one agent pursue the goal of building a cv, we design *every* agent to pursue this common goal. This way, a shared cv emerges in the system due to the interactions of individual agents. We describe several communication strategies the agents can follow, and evaluate their quality according to specified criteria.

Although this paper is primarily intended for the ontology integration community, we follow an approach which is related to research done in the language evolution community. Following Luc Steels [10], we assume that the meaning of a concept can be conveyed to another agent by pointing to shared instances. Furthermore, we also approach a language (or an ontology) as a complex adaptive system which can be studied by means of simulation [9, 7]. However, whereas most research on language evolution has an explanatory goal, our goals are purely constructive. Our work can thus be viewed as an application of language evolution to ontology integration.

In the next section, we present the communication protocol. Because the communication protocol is non-deterministic, a communication strategy is required to guide the agents in making the right choices. Section 3 presents four possible strategies. The strategies are compared in three simulation experiments which serve to evaluate how the strategies contribute to the agents' mutual understanding, what kind of cv the strategy gives rise to, and how well the strategy performs in a sparsely connected network. We conclude in section 4 and give directions for future research.

2 Communication Protocol

The communication protocol in figure 1 defines the possible dialogues that may happen when an agent informs another agent about something. After Ag_i has sent its message to Ag_j, Ag_j may respond in two ways. When it knows the terms in the message, it translates the terms to its own ontology and responds "OK". When it doesn't know a term, it responds with "ConceptUnknown", which leaves no option for Ag_i but to *teach* the meaning of the term to Ag_j. For the purposes of this paper, it suffices to say that the teaching agent presents a number of examples of the term which enables the other agent to discover its definition in terms of its private ontology. For a more elaborate discussion on this teaching

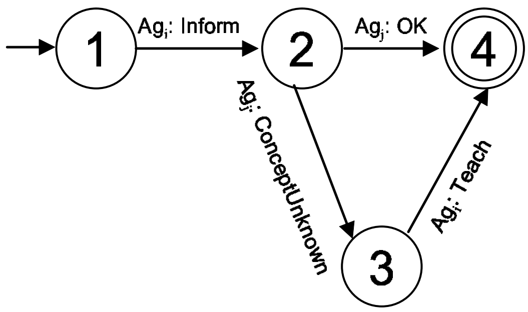

Fig. 1. Communication Protocol

process, the reader is referred to [11]. After Ag_j has learned the meaning of the term, it translates the message to its own ontology after all, and the dialogue finishes.

We now discuss the inform action in further depth. In doing so, we distinguish the following agent-components:

- ONT : The set of concept-names defined in the agent's private ontology.
- AKB : The assertional knowledge base, consisting of statements of the form $x(a)$, where $x \in$ ONT and a is an element from the domain of discourse.
- VOC : The set of terms constituting the agent's knowledge of the communication vocabulary.
- DEF : $\{\langle x, y\rangle | x \in \text{VOC} \wedge y \in \text{ONT}\}$: A set of concept pairs, defining every concept in VOC as a concept in ONT.
- SCR : VOC $\rightarrow \mathbb{N}$ is a data structure which assigns a score to every term. This score is used by the agent's strategy.

The communicative abilities of the agents are specified as actions. During the execution of actions, messages are sent through the instruction send(Ag$_j$, $\langle$SpeechActType$, p_1, .., p_n\rangle$), where Ag$_j$ is the addressee of the message, the SpeechActType specifies what the message is about, and $p_1..p_n$ are parameters of the message. The effect of this instruction is that Ag$_j$ is able to perform a Receive(Ag$_i$, $\langle$SpeechActType$, x_1, .., x_n\rangle$) action, where Ag$_i$ is the sender of the message and $x_1..x_n$ are instantiated to $p_1..p_n$. In the specification of actions we will adopt Ag$_i$ as the sender and Ag$_j$ as the receiver of messages. The Inform-action is specified as follows:

Action `SendInform`$(\text{Ag}_j, x(a))$
where $x(a) \in AKB$
Candidates $:= \{y' | \langle y', x\rangle \in \text{DEF}\}$
$y :=$ `Select`(Candidates), *where* `Select` *is implemented in the agent's strategy*
Send$(\text{Ag}_j, \langle \text{Inform}, y(a)\rangle)$

Action `ReceiveInform`
Receive$(\text{Ag}_i, \langle$ Inform,$x(a)\rangle)$

If x $\in$ VOC Then Add $y(a)$ to AKB, s.t. $\langle x, y\rangle \in$ DEF
Else Send(ConceptUnknown)
`UpdateScore`(Ag$_i$,x), *where* `UpdateScore` *is implemented in the agent's strategy*

We assume that initially, the set VOC = ONT, and DEF = $\{\langle x, x\rangle | x \in \text{VOC} \cap \text{ONT}\}$. This means that the agents start communication by using their private concept-names. The SendInform action therefore always sends a message because there is always a way to translate a concept in ONT to a term in VOC. The ReceiveInform action does not always translate a received term to the private ontology because there is not always a translation available in DEF. In these cases, the agent responds with "ConceptUnknown" after which the unknown term will be taught to him. When Ag$_i$ teaches a term x to Ag$_j$, Ag$_j$ adds the term to VOC, adds the definition of the term to DEF and sets SCR(x) to 0. For this reason, an agent soon ends up having multiple ways to translate a concept in ONT to a term in VOC: one possible translation is its private concept-name (a translation which is available due to the initialization of VOC), other translations are terms it has learned from other agents. All possible translations are listed in the set "Candidates" in the `SendInform`-action. Although all terms in the Candidate-set are allowed translations, the agents follow a strategy to select the *best* candidate. The agents base their decision on the score of the term, given by SCR. The agents may update the score of a term each time they receive a message with that term from other agents, i.e. in the `UpdateScore` action which is performed at the end of the `ReceiveInform` action. Different implementations of `Select` and `UpdateScore` give rise to different communication strategies. This is the topic of the next section.

3 Strategies

The decision to be solved by the agent's strategy involves choosing a term among the possible candidates when sending an inform message. We assume that `Select` always returns the term with the highest score. The differences between the strategies lie in the way the scores are updated after a message is received. Strategy s1 only attributes a score to its private concept-name. Therefore, in systems with agents that use strategy s1 (s1-systems), every agent holds on to its own private concept-names when it comes to speaking. To understand other agents, they learn each other's concept-names. This strategy is analogous to ontology alignment ([8]): every agent has a mapping to every other agent's ontology. Strategy s2 attributes a score of 1 to the most recently received term; all other terms with the same definition are attributed a score of 0. In s2-systems, an agent chooses the candidate term it has most recently received from another agent. Strategy s3 increases the score of a term each time a message with that term is received. This way, the agents in a s3-system choose the candidate term which they have most frequently received. Strategy s4 is similar to s3 but also takes into account *which* agents have used the term. The score of a term is increased more when an agent with many acquaintances uses the term than when an agent with few acquaintances uses the term. We assume that the number of acquaintances of an agent can be known by the agents. In systems where every agent has an equal amount of acquaintances, s4 gives rise to the same behavior as s3. We therefore only discuss s4 in section 3.3, where we consider networks in which the agent's number of acquaintances differ. The four communication strategies are specified in figure 2.

We evaluate the performance of these strategies using simulation experiments. To obtain a clear picture of the properties we are interested in, in the experiments, we abstract away from as many irrelevant aspects as possible. This way, the agents only exchange one meaning (but may use different terms to do so). An experiment consists of t steps at which randomly a speaker and a receiver is selected from the set of connected agents. The agents follow the communication protocol as described in the previous section. Because the agents only exchange one meaning, the elements in an agent's set VOC all have the same definition in DEF. Therefore, all elements in VOC are allowed candidates to use in a message. We distinguish between the terms that are understood by an agent (the *understandable terms*), and those that are spoken by an agent (the *spoken terms*). The understandable terms of Ag_i are those in the set VOC. The spoken terms are those that are selected from VOC by the agent's strategy. Using s1 and s2, there is only one spoken term per agent, because there is only one term with a highest score. s3 and s4 may give rise to several terms with an equal highest score, amongst which the the strategy randomly chooses one. In these cases we say that there are several spoken terms, i.e. those with the highest score. The model-components are summarized below:

- A multi agent system is a network of agents, where:
  - n is the number of agents.
  - $\mathrm{AG} = \{\mathrm{Ag}_1..\mathrm{Ag}_n\}$ is the set of agents
  - $\mathrm{CAG} \subseteq \mathrm{AG} \times \mathrm{AG}$ is the set of pairs of *connected agents*, i.e. those that can communicate with each other.
- UT_i is the set of understandable terms of Ag_i. This is equal to VOC of Ag_i
- $\mathrm{ST}_i \subseteq \mathrm{UT}_i$ is the set of Spoken terms of Ag_i. ST_i is equal to the set of terms with a maximum score.

We evaluate the communication strategies using the following criteria:

- How many steps does it take before every agent understands each other?
- How many different terms are used in the system?
- How does the strategy perform in sparsely connected networks?

These criteria are described in the following sections.

3.1 Understandings rate

We define that agent Ag_i understandable for agent Ag_j when every spoken term of Ag_i is understandable for Ag_j, i.e. $\mathrm{ST}_i \subseteq \mathrm{UT}_j$. We define the understandings rate to be the number of connected agent pairs that are understandable to each other (without having to teach each other new concepts) divided by the total number of connected agents. This is formalized as follows

// The Select action is the same in every strategy
// Select returns the concept from Candidates with the highest score
// If several concepts have an equal highest score, it randomly picks one from them
Action `Select`(Candidates)
Randomly choose y from $\{y'|y' \in$ Candidates and
for all $z \in$ Candidates: $\mathrm{SCR}(z) \leq \mathrm{SCR}(y')\}$
return y

Strategy s1:
// Always speak your own concept-name.
The scores of the agent's private concept-names is kept equal on 1.
// The score of every other term therefore remains 0.
Action `UpdateScore`(Ag_i, x)
For all $y \in \mathrm{VOC} \cap \mathrm{ONT}$ Do $\mathrm{SCR}(y) := 1$

Strategy s2:
// Speak the term which most recently another agent used
Action `UpdateScore`(Ag_i, x)
SCR(x) := 1
For all $y \neq x$ with $\mathrm{DEF}(y) = \mathrm{DEF}(x)$ Do $\mathrm{SCR}(y) := 0$

Strategy s3:
// Speak the most frequently used term
Action `UpdateScore`(Ag_i, x)
$\mathrm{SCR}(x) := \mathrm{SCR}(x) + 1$

Strategy s4:
// Speak the most frequently used term weighted by the user's number of acquaintances
// The number of acquaintances of Ag_i is denoted by k(Ag)
Action `UpdateScore`(Ag_i, x)
$\mathrm{SCR}(x) := \mathrm{SCR}(x) + \mathrm{k}(\mathrm{Ag}_i)$

Fig. 2. Implementations of different strategies

Definition 1. $UR = \frac{\#\{\langle Ag_i, Ag_j\rangle | \langle Ag_i, Ag_j\rangle \in CAG \wedge ST_i \subseteq UT_j\}}{\#CAG}$

All communication strategies eventually result in an understandings rate of 1, a situation which we call *common understanding*. However, the strategies differ in the number of steps it takes before common understanding is achieved. Following strategy 1, common understanding is achieved only after *every* pair of connected agents have communicated with each other. Given that at each time step a random pair of communicating agents is selected, probability theory predicts that the expected number of steps before UR = 1 is given by:

$E(X) = q \sum_{r=1}^{q} \frac{1}{r}$
where
- X is the number of steps it requires to reach common understanding.
- $q = \#$CAG

Consider an agent system where n=20 and CAG = AG $\times$ AG, i.e. every agent speaks to every other agent. In this system, the formula above predicts that common understanding is reached after approximately 2400 turns. This prediction is confirmed by the experimental results shown in figure 3. The experiment also

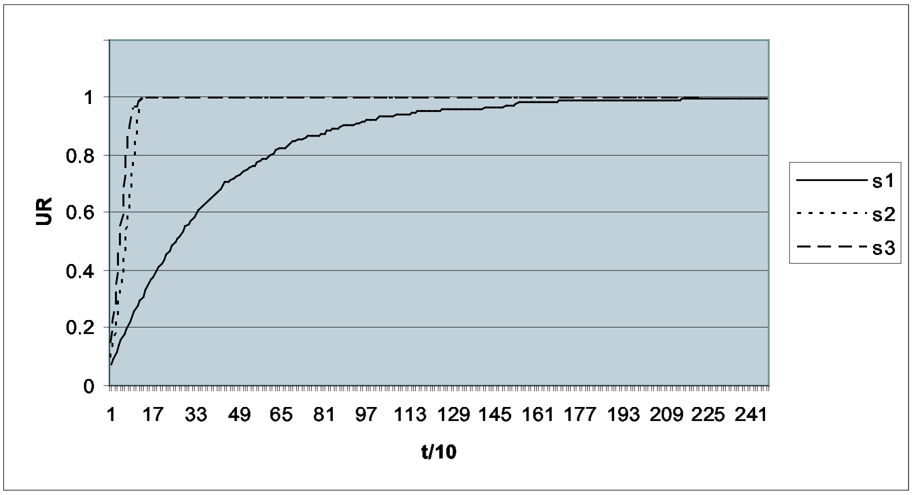

Fig. 3. Strategy performance w.r.t. understandings rate

reveals that s2 and s3 require much fewer steps than s1 to reach common understanding. We argue that this is because s2 and s3 incites the agents also to speak each other's terms which gives rise to groups of agents that speak the same term. Therefore, two agents from the same group that never communicated before, are able to understand each other nevertheless. This explains the fast increase of UR in s2- and s3-systems.

Besides the speed of increase in UR , another important issue is the *size* of the communication vocabulary. s1 gives rise to a communication vocabulary of size n, i.e. every agent's private concept-name eventually becomes part of the communication vocabulary. Besides the fact that this is the main cause for s1's slow increase in UR, another disadvantage is that newcomers in the system would have to learn a large number of terms to be able to understand everyone. It is therefore desirable that the total number of different terms which are spoken by the agents is as small as possible. This aspect is studied in the following section.

3.2 Number of terms

We refer to the number of terms that are used in the system with NT which is defined as follows:

Definition 2. $NT = \# \bigcup_i ST_i$

Obviously, in an s1-system the NT remains equal on n. For s2, it holds that, eventually, the NT becomes 1 in which case we say that the communication vocabulary has *converged*.

Property 1. In every s2-system: $\lim_{t \to \infty} P(\text{NT=1 after t steps}) = 1$

Proof: In every s2-system, the probability that NT becomes 1 after n steps is greater than 0. This happens when n times a speaker is selected with SC = x, and every agent with $SC \neq x$ is selected at least one time as a hearer. Furthermore, each "trial" (the execution of n steps) is independent of the other trials: the failure of one trial to result in NT= 1, does not systematically influence the probability that the next trial results in NT =1. Therefore, by the definition of chance, as the number of steps approaches infinity, the probability that NT=1 approaches 1.

□

Although the property described above is a nice theoretical result, in practice we are interested in the speed at which NT decreases. To obtain statistical significance, in the next experiment we use n=1000. Again, we set the structure as a fully connected network, i.e. CAG = AG × AG. Figure 4 shows the decrease of NT in this experiment. The results of this experiment show that the s2-system gives rise to a great decrease of NT at first, but does not lead to a fully converged cv. After 100000 steps, there were still 9 different terms in use, and it would have taken a very long time before the cv would have converged. S3, on the other hand, performs relatively poorly at first, but gives rise to a converged cv after approximately 35000 steps. By that time, every agent spoke 35 times on average, which is not a bad result given the large size of the system.

The reason why the cv does not converge within reasonable time in an s2-system, is that in these systems NT decreases only by coincidence. At the end,

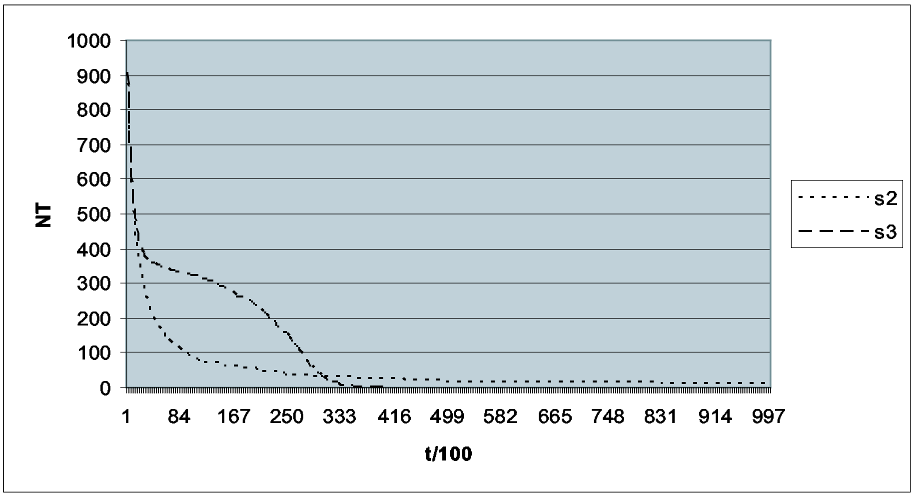

Fig. 4. Strategy performance w.r.t. number of terms

the probability becomes very small that one of the terms "dies out", because each term is used by many agents. Agents that follow s3, keep an approximation up to date of which terms are most frequently used. Because every agent uses the term which they believe to be most frequently used, NT not only decreases by coincidence, but is guided by the beliefs of the agents.

Because s3 gives rise to a stable and converged cv, s3 is preferable over s2. In the next section we discuss the strategy's performance in other network structures.

3.3 Network structure

In the previous sections, we have evaluated the strategies in fully connected agent networks. In this section we discuss the strategy's performance in a more sparsely connected network. In doing so, we adopt some terminology from graph theory. We assume that the network is a non-directed graph, i.e. $\langle x, y\rangle \in \text{CAG} \rightarrow \langle y, x\rangle \in \text{CAG}$. We call the number of acquaintances of an agent, the *degree* of an agent, denoted by k:

Definition 3. $k(Ag_i) = \#\{Ag_j | \langle Ag_i, Ag_j\rangle \in CAG\}$

Many networks, amongst which the world-wide-web, are structured as a *scale-free network* [2]. Networks of this type are characterized by a large number of nodes with a relatively small k. A few nodes, however, are stars in the network and have a relatively high degree. Stated more precisely, the degree distribution follows a power law: $P(k) \sim k^{-\gamma}$.

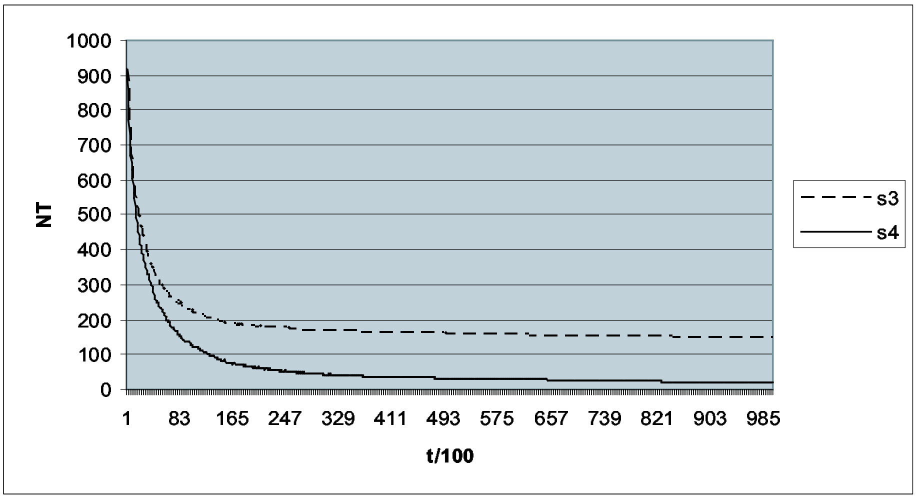

Fig. 5. Strategy performance w.r.t. NT in a scale-free network

The next experiment describes the results of s3 and s4 in a scale-free network, with n=1000, an average k of 3.11, and a maximum k of 50.

Although s3 gave rise to a converged communication vocabulary in fully connected networks within reasonable time, the NT does not go below 140 in the experiment described above. This happens because most agents have a low degree and are not capable to form a realistic approximation of which terms are most frequently used. To overcome this problem, s4 using agents take into account the degree of the speaking agent when they update their scores. Agents with a high degree have a more realistic approximation of the most frequently used sign, and are therefore taken more seriously than agents with a low degree. This explains why s4 performs better than s3 in this experiment, although it still does not give rise to a fully converged communication vocabulary.

4 Conclusion and future research

In this paper, we have described four strategies which, when used by individual agents, give rise to a shared communication vocabulary on a global level. While evaluating these strategies, the following issues were demonstrated. Firstly, a strategy which incites the agents to adopt each others' concept-names for speaking (such as s2 and s3) has considerable advantages over a strategy in which every agent holds on to its own concept-names for speaking (such as s1 or ontology alignment). Secondly, the strategy of adopting the most frequently used concept-name (s3) is usable in a network with a simple interaction pattern, i.e. when everyone speaks to everyone with equal probability. Thirdly, it was demonstrated that in a scale-free network, the performance of the strategy of adopting the most frequently used concept-name is improved when the agents' number of acquaintances is taken into account (as is done in strategy s4).

We continue this line of research by testing the strategies in networks with more complicated interaction patterns and by exploring adjustments which could lead to better performance in those environments. Furthermore, we intend to explore combinations between these communication strategies and centralized services for ontology integration as mentioned in the introduction of this paper.

References

1. FIPA Ontology Service Specification. http://www.fipa.org/specs/fipa00086/.
2. Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509-512, 1999.
3. T. Bylander and B. Chandrasekaran. Generic tasks for knowledge-based reasoning: the right level of abstraction for knowledge acquisition. *Int. J. Man-Mach. Stud.*, 26(2):231–243, 1987.
4. Adam Farquhar, Richard Fikes, and James Rice. The ontolingua server: a tool for collaborative ontology construction. *Int. J. Hum.-Comput. Stud.*, 46(6):707–727, 1997.
5. T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
6. J. Heflin and J. Hendler. Dynamic ontologies on the web. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 443–449, Menlo Park, CA, USA, 2000. AAAI Press.
7. S. Kirby. *Syntax without Natural Selection: How compositionality emerges from vocabulary in a population of learners*, pages 303–323. Cambridge University Press, 2000.
8. N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. *In Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.
9. L. Steels. The origins of syntax in visually grounded robotic agents. *Artificial Intelligence*, 103:1–24, 1998.
10. Luc Steels. Language as a complex adaptive system. In M. Schoenauer, editor, *Proceedings of PPSN VI*, Lecture Notes in Computer Science, Berlin, Germany, September 2000. Springer-Verlag.
11. Reference to technical report to appear in the final version of this paper.
12. M. Uschold and M. Gruninger. Creating semantically integrated communities on the world wide web. *Semantic Web Workshop Co-located with WWW 2002 Honolulu*, 2002.