A Knowledge Based System for Person-to-Person E-Commerce

Eugenio Di Sciascio, Francesco M.Donini, Marina Mongiello,
Giacomo Piscitelli
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Via Re David, 200 – 70125 BARI, Italy
e-mail: {disciascio,donini,mongiello,piscitel}@poliba.it

Abstract

Short-term peer-to-peer (or person-to-person) e-commerce calls for an infrastructure treating in a uniform way supply and demand, which should base the matchings on a common ontology for describing both supply and demand.

Knowledge representation — in particular description logics — can deal with this uniform treatment of knowledge from vendors and customers, by modelling both as generic concepts to be matched.

We propose a logical approach to supply-demand matching in person-to-person e-commerce, which is deployed in a prototype system implemented for a particular case study but easily generalizable and is based on Classic, a well known knowledge representation system. Limits and advantages are analyzed and discussed.

1 Introduction

Electronic commerce differs from traditional commerce primarily in the way that information is exchanged and processed. Personal contacts, or at most contacts mediated through the phone or postal system, are or should be, substituted by information exchanged between computers flowing along digital networks. E-commerce can anyway mean several things.

In this paper we focus on what is known as Consumer-to-consumer e-commerce. This category, which we are more likely to call person-oriented or Person-to-Person (P2P) in accordance with the proposal in [4], is emerging slowly, as it is not backed up by the interests of large

companies, but we believe it has great potentials for widespread diffusion. We basically consider the P2P scenario an electronic marketplace, or better an agorá[1], where peer consumers may propose their goods and products and dynamically deal with counteroffers or further specifications.

Our research tries to fit an existing gap in present deployment of P2P e-commerce, by extending the technology of search and of descriptions of searched items, in order to ease and improve the free play of demand and supply.

Most of current consumer or person oriented e-commerce strives to reproduce, with limited success, the usual commercial interaction taking place in physical stores. On one hand, consumers are reached by blind advertising on store products, and brand names — which are built out of massive economic investments — fill customers' lack of information. On the other hand, (virtual) stores are visited by those consumers who are interested in buying a product they think is available at that store. Portals reproduce specialized catalogues and consumer magazines.

We believe that this situation is not the necessary picture of e-commerce; it depends on the available technology for publishing and searching on the web.

Publishing on the Web is in fact a form of blind advertising of products. Although not as intrusive as spamming, the publisher has no precise idea of who is going to access the published information. Producers rely on keywords, and on indexing on prominent search engines, to be found by potential customers. Keywords are affected by the wide-experienced problems of both synonyms, and polysemic words. Categorization by prominent index sites may help some producers to be effectively found by potential customers; however, on the customer side, relying on indexes may result in a loss of useful information about small-size producers, or short-time suppliers. This is particularly relevant for agricultural products: supplies tend to have a seasonal time window, and different producers may alternate on the market, each one without the sufficient economic strength to put up a long-term, visible and reliable site.

On the customer side, there is no way to reach a product but by active search using the available technology, which penalizes P2P in favor of B2C. The use of main portals and indexes to speed up search tends to favor long-term renowed brands, in comparison with short-term supplies by small producers. This may result in a loss of competitive offers, which could have been economically more convenient just because of either geographical proximity, or smaller advertising overhead on the overall price.

One may be tempted to solve the problem of P2P by resorting to presently available Database technology. A site specialized in particular products

[1]Agorá: An assembly; hence, the place of assembly, especially the market place, in an ancient Greek city.

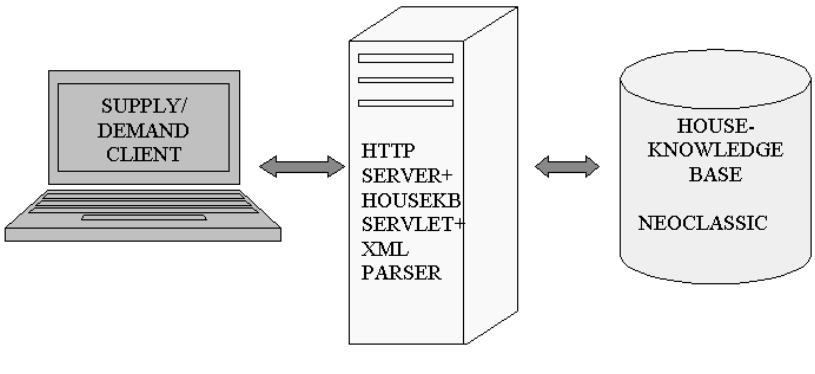

Figure 1: Three-tier architecture of the prototype system.

may help suppliers to dynamically insert and delete offers for definite time slots. The site could help also customers to issue queries regarding particular products, probably with the help of some interface to formulate queries that can be actually answered — using the right vocabulary, the right level of abstraction. To clarify with an example, an interface may help a user wondering, "Can I ask about 'merlot grapes', or should I use 'red grapes', or just 'grapes', and scrutinize retrieved answers after?". In a nutshell, the interface for queries should clarify to the user the Database Ontology (names and relations in the Database schema).

However, such a database technology solution captures only half of the matching of supply and demand in P2P e-commerce. In fact, it captures the request of a customer as a query, to be answered in a web-based database of offers by suppliers. But on the other hand, also the offer of a supplier can be treated as a query for interested customers in a database of customer requests, or even personal profiles. The choice of which is the data, and which is the query, depends here just on a point of view, and maybe, on who is more interested in actively finding the answer — and that is not necessarily the customer.

Summarizing, we believe that in small-size, short-term P2P commerce there is the need of an infrastructure treating in a uniform way suppliers and demanders, which should base the matchings on a common ontology for describing both supplies and demands.

Knowledge representation — in particular description logics — can deal with this uniform treatment of knowledge from suppliers and customers, by modelling both as generic concepts to be matched. The matching could be accomplished by a classification on the hierarchy of concepts, that forms the common ontology. Both supplier offers and customer requests could be classified, leading to exact match when they describe the same concept, and to partial match (still meaningful) when one is a subconcept of the other or concepts are compatible. For instance, an offer of 'red grapes' partially matches a request for 'merlot grapes'. The

partial match can then be further investigated, still with the help of the infrastructure.

In fact, the logical approach — which Description Logics are based upon — allows for an open-world assumption. Incomplete information is allowed (and can be filled after a selection of possible matches), and absence of information can be distinguished from negative information, allowing to discard offers/requests without the necessary properties, and to ask for missing information in the potential matches.

Here we propose an approach to P2P e-commerce based on Description Logics and describe its implementation in a prototype system for a virtual brokering agency for apartments rental. Limits and possibilities are also addressed.

2 A logical approach to matching in P2P e-commerce

We now apply the framework of Description Logics to supply-demand matching in P2P e-commerce marketplaces. We suppose for simplicity a basic P2P e-commerce setting, in which every transaction involves just one supplier and one customer.

We formalize both supplies and demands as concept names (of two distinct alphabets), which share concepts and roles of a common product ontology. Hence, we assume the existence of two disjoint alphabets of supply names $\mathcal{C} = \{C_1, \ldots, C_n\}$, and demand names $\mathcal{D} = \{D_1, \ldots, D_m\}$. Moreover, we assume that the domain of interest is modeled with concepts and roles as in a generic Description Logic.

E.g., proposing the supply of an apartment in Brooklyn Heights, NY, could be described by

$$C_1 \doteq \texttt{apartment} \sqcap \exists \texttt{location.Brooklyn-Heights} \sqcap (\geq\ 2\ \texttt{rooms})$$

while two demands in the same domain could be described by

$$\begin{aligned} D_1 &\doteq \texttt{apartment} \sqcap \exists \texttt{options.Back-Yard-Garden} \\ D_2 &\doteq \texttt{apartment} \sqcap (\leq\ 1\ \texttt{rooms}) \end{aligned}$$

The interpretation of supply and demand concepts is on a single, common domain of possible transactions; the set of transactions compatible with them. When a transaction involves a single traded good — such as apartments — supplies and demands can be interpreted directly as sets of goods that can be exchanged. However, we prefer to interpret concepts as transactions since there are exchanges involving continuous goods, such as fruits, oil, etc.

Definition 1 *An offer C and a request D are [in]compatible iff the concept $C \sqcap D$ is [un]satisfiable.*

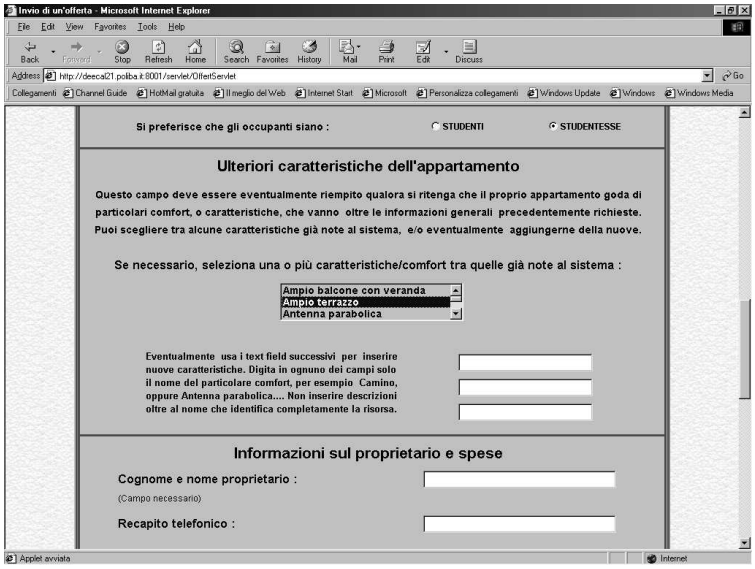

Figure 2: The form specifies the basic features usually requested to describe an apartment, such as price, number of rooms and so on. Further features/concepts, originally not foreseen and now part of the system knowledge, can be selected from the scrollable window. New concepts can be inserted in the textfields to extend the description of the apartment. After the insertion, the knowledge base is automatically updated. The new concepts appear in the list of features in the scrollable window and are available for further searches or insertions.

For the above examples, D_1 is compatible with C_1, and further inquiries could be issued to both parts to see if the transaction can be concluded. Instead, C_1 is incompatible with D_2, so this matching can be discarded in a first filtering phase.

There are cases, however, in which we can establish from known data that the transaction is already possible: *e.g.*, a demand of exotic fruits is fulfilled by a supply of bananas. Obviously, to establish this conclusion the system must have knowledge (through an inclusion assertion) that bananas are exotic fruits.

Definition 2 *A demand D is* fulfilled *by a supply C in a TBox T if C is subsumed by D in T.*

Unmatched demands and supplies can be "stored" as new concepts, and classified accordingly in the TBox. As soon as a matching supply/demand enters the system, the old request can be reexamined. In this sense, the approach based on Description Logics is balanced between supply and demand, and much different from usual B2C portals where a customer

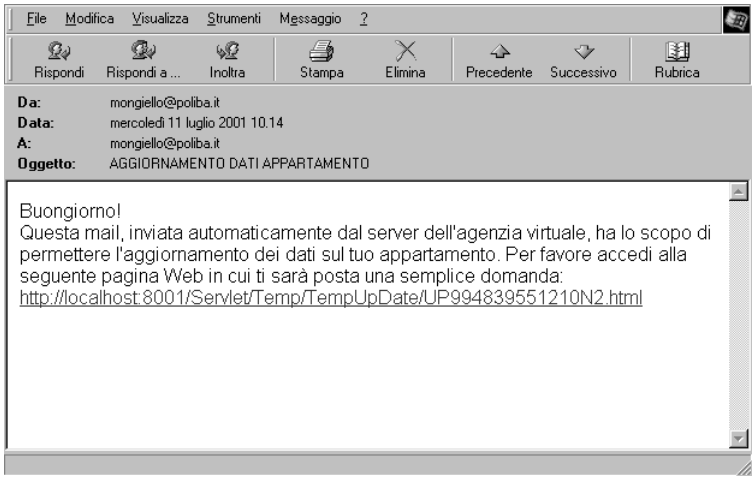

Figure 3: The text of an e-mail automatically sent to the owner of an apartment. The message points to a URL where the owner is asked to answer about the presence of features not included in the original description.

has to make active efforts to find the goods s/he looks for, and if she does not find them now, s/he has to re-enter data next week.

3 A case study: renting apartments to students

3.1 Scenario

Differently from most other countries, Italian universities do not have dormitories for their students, which are only provided to a minimum percentage of them. Also, often faculties are spread in different parts of towns and not concentrated in campuses.

This situation leads to a flourishing activity of privately owned apartments rental. Furthermore largest part of students do not rent an entire apartment or house, instead they are more likely to ask for a room and often for just a bed within a room.

Supply and demand are extremely dynamic and currently there are no brokering agencies, at least in most italian towns. Therefore, although there is a noteworthy aggregate turnover, typically supply meets demand in naive ways, *e.g.*, advertisements pinpointed on show-cases, word passed round, etc.

This business scenario can hence be considered an ideal candidate for a web-based system designed for person-to-person small business e-

commerce whose objective is becoming a virtual brokering agency endowed of innovative services able to ease use and interaction, and closer to user needs than most of business-to-consumer e-malls are.

Main services we considered objective of our case study are:

1. Support to the user in the data insertion and query submission. The user is incrementally guided in the definition of a query or an offer. We remark here that though we distinguish for simplicity the supplier and the demander they are treated in a uniform way by the system.
2. Automatic construction and verification of satisfiability of the query.
3. Deduction of new knowledge on the basis of available data. As an example, the system automatically detects apartments available in a convenient area once the faculty has been entered.
4. Ability to provide conceptually approximate answers in the presence of unsatisfiable queries. Notice that this the way a human clerk behaves: when a request can not be satisfied he/she will propose the closest alternatives to the client, will not answer "no match".
5. Ability to manage incomplete queries and possibility to ask for unforeseen (hence not immediately available) services and features to the supplier.
6. Storage of satisfiable queries/demands that were still unmatched, with automatic reexamination when new supplies are provided, and notification on succesful match between supply and demand. The same service is available for unmatched supplies.

The remaining of this section describes main features of the implemented system. It is anyway noteworthy that, though the system has been actually designed as a virtual brokering acengy for apartments rental, similar issues exist in several other business scenarios. A nice property of our approach is in fact that reverting to other scenarios is quite simple, provided that basic concepts and a simple ontology are available. The system will extend its knowledge as more information becomes available without having to modify the structure of the system.

3.2 The virtual brokering agency

The proposed prototype system is based on a NeoClassic engine interfaced to the web via Java servlet technology. Classic is a member of the KL-ONE family of knowledge representation languages [2]. NeoClassic, a Classic implementation, is basically a knowledge server to build, manage, infer about, and query a classification hierarchy. NeoClassic is a frame-based knowledge representation system that differentiates between

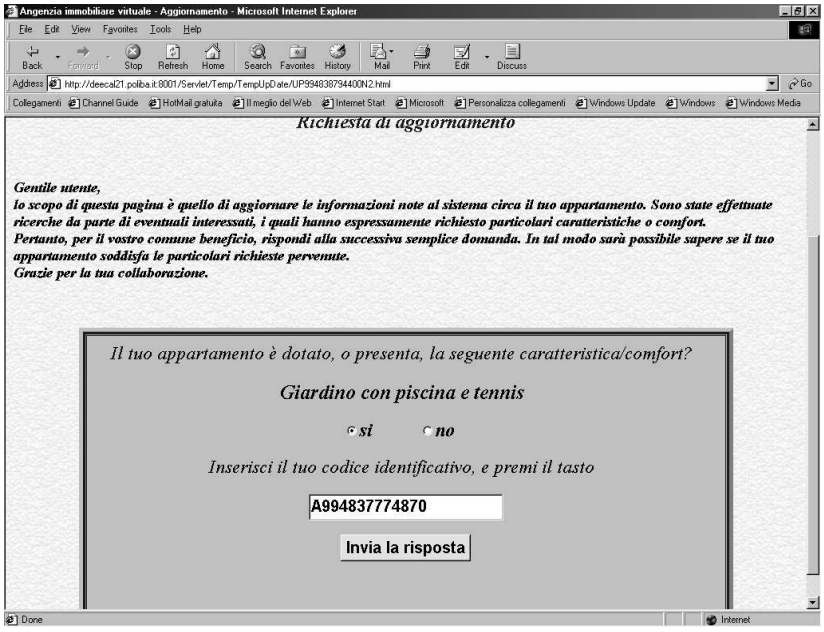

Figure 4: The form specifies a new feature, requested by an interested demander, not available in the apartment description. The owner can answer the question by clicking on the radiobutton and fill the textfield for the authentication procedure with his password.

terminological and assertional aspects of knowledge representation, and which focuses on the key inferences of subsumption and classification, typical of description logics.

Figure 1 shows the three-tier architecture of our prototype system. The web server passes requests to a Java servlet that communicates with the NeoClassic engine running as a daemon in the background. The main user interface for both supply and demand submission is a Java applet. Classic output is parsed in XML; the system is also interfaced to a mailer and to a SMS (Small Message System) server for automatic news notification.

Submission of a new query or offer is carried out incrementally. The user can introduce basic elements, such as price, type of the apartment, number of rooms, etc. A distinguishing aspect is the possibility to ask for new services or features not foreseen at the design stage. The system extends its knowledge and new concepts are added to the knowledge base. Figure 2 shows the form that includes a text area for specification of new concepts. The scrollable window in the upper part of the figure shows other concepts added and consequently part of the system knowledge. The system includes a small thesaurus and a predictive text input mechanism to avoid repeated insertion of similar concepts.

Our basic service is the matching of compatible descriptions. A query

that matches a description will also subsume all other descriptions that in the system hierarchy are classified below the first matched description. Supplying a new apartment is also a description matching. The position of the new supply in the system hierarchy is determined considering all the descriptions that the new description is subsumed by. Once the position has been found, a reclassification takes place to determine the descriptions that satisfy it, which are then tied in the system hierarchy.

Each request, a supply or a demand, is given a unique code for information tracking. A supply is always an apartment, which can be rented as a whole or room by room or also on a "bed" basis. A demand can be any of the above. Test functions allow several deduction services including: reachable faculties given the address of an apartment and vice versa; computation of prices per room or per "bed" given the cost of an apartment; computation of spare rooms or spare beds. It is noteworthy that when a new concept is introduced by a user looking for a new service/feature, the system automatically contacts owners of apartments that fulfill all other user requirements but the new one through an e-mail or a SMS, see Figure 3 for an example.

The message points the owner to a web page where the question on the newly requested feature is posed, as it is shown in Figure 4. If he/she confirms the availability of the requested service/feature the system stores the new information and consequently informs, with the same procedure (an e-mail or a SMS), the demander about the fulfillment of his/her request.

The system also correctly handles requests that cannot be fulfilled for lack of apartments offers satisfying all needs expressed. The system answers declaring the reason why the query could not be completely satisfied, so that the user may eventually release the constraint, that is, turning to the hierarchical structure of the knowledge base, raise towards a less specified description.

It also proposes those apartments whose description, though not perfectly matched by the request are closer to it. We remark that this is the way we believe a human clerk would behave: he/she would propose something that, though not exactly in agreement with the customer request is close enough to be of interest. As an example Figure 5 shows the system answer, with the negative answer (and the reason for it) and alternative solutions.

If a user refuses alternatives his/her request is not lost. The system stores it; on insertion of new supplies matching the query the system automatically informs the user that a new offer is available satisfying his/her requests.

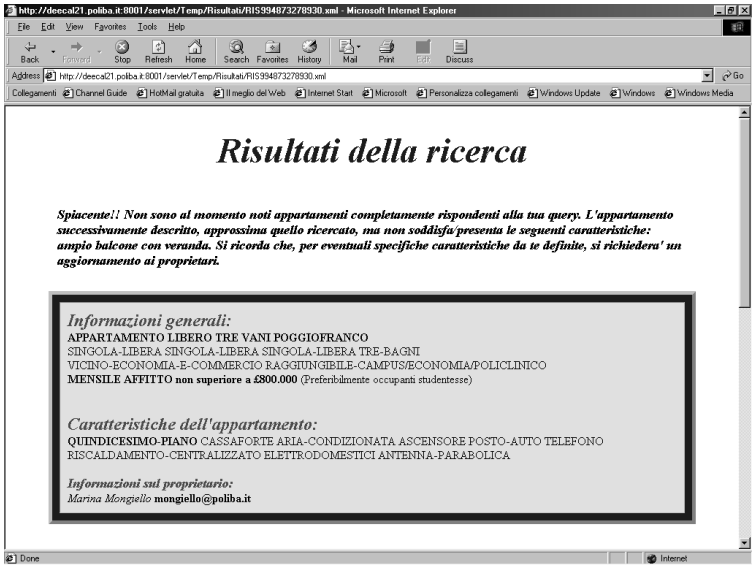

Figure 5: Response to a query not completely matched. In the upper part the system provides information about the requirements not fulfilled. It points out that one requirement, "balcone con veranda" (i.e. porch), could not be fulfilled by any available supply. It also proposes in the lower part apartments that fulfill all other requirements of the query.

4 Conclusion and discussion

We have proposed the application of the Description Logics framework to supply-demand matching in a P2P scenario, modelling both as generic concepts to be matched. We have also presented as a realistic case study a prototype of system that implements a virtual brokering agency for apartments rental. The system is fully functional and is currently in alpha test stage.

Our conclusion is that Description Logics can be almost an ideal candidate for scenarios where peer entities interact for supply-demand matching, providing a level of interaction with user-information needs which is not available with current database technology. We end the paper discussing some pros and cons of our choices.

We had several reasons that led us to choose Classic instead of other implementations of Description Logics. First, since we are working on a years-long project, we need to rely on a stable system. Second, we need *system* functions like "list all roles attached to this concept", or "list the *derived* subsumers of this concept".

Moreover, we need a language escape for requirements that are not expressible in the Description Logic. Although weak as a *reasoner*, one of the strongholds of Classic is exactly the possibility to use computed fillers and test functions in the host language.

Third reason, but not less important, there is a large documentation on Classic, which is not just the system manual, but also tutorial-like papers [2] that allow newcomers to grasp very rapidly the system capabilities, and how to exploit them.

Nevertheless, we experienced also some problems in using Classic, such as the fact that computed fillers required almost continuous recompilation of the code. Also, Classic appears to have some strange behavior in the removal of "told" information, which requires a careful handling. Furthermore, lack of the disjunction operator (OR), and qualified existential quantification ("an apartment with a single room") required some work to simulate their behavior. We also needed small computing capabilities to express things as "an apartment with five rooms, three of which are rented, is an apartment with two free rooms".

To circumvent this problem in Classic, we had to code in concept names all combinations of available and rented rooms, and use rules; this appears a somewhat cumbersome compilation of subtraction.

Finally, we would have find useful some epistemic capabilities to express that, *e.g.*, to enter a new apartment in the system, its location must be entered too, or that the owner already told that smokers are allowed.

Acknowledgements

We wish to acknowledge help of our former students P. Bacco and M. Ranieri, and of P.F. Patel-Schneider who clarified various points of NeoClassic implementation.

This work has been supported by project MURST-CLUSTER22 subcluster *"Monitoraggio ambiente e territorio"*, workpackage: *"Sistema informativo per il collocamento dei prodotti ortofrutticoli pugliesi"* and by EU-POP project *"Negotiation Agents for the Electronic Marktplace"*.

References

[1] Alexander Borgida. Description logics in data management. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):671–682, 1995.

[2] R. J. Brachman et al. Living with CLASSIC: When and how to use a KL-ONE-like language. In John Sowa, editor, *Principles of Semantic Networks: Explorations in the representation of knowledge*, pages 401–456. Morgan Kaufmann, 1991.

[3] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the decidability of query containment under constraints. In *Proceedings of the Seventeenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[4] P. Devambu, S.G. Stubblebine, and M. Uschold. The next revolution: Free, Full, Open Person-2-Person (P2P) E-commerce. In *Proc. of DL 2000 Workshop*, 2000.

[5] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Reasoning in description logics. In Gerhard Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.

[6] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[7] William A. Woods and James G. Schmolze. The KL-ONE family. In F. W. Lehmann, editor, *Semantic Networks in Artificial Intelligence*, pages 133–178. Pergamon Press, 1992. Published as a special issue of *Computers & Mathematics with Applications*, Volume 23, Number 2–9.