

# An Intelligent Method for Inferring Information about the User/ Client

**J. F. Baldwin, T. P. Martin, and A. Tzanavari,  
University of Bristol, United Kingdom**

[Jim.Baldwin@bristol.ac.uk](mailto:Jim.Baldwin@bristol.ac.uk), [Trevor.Martin@bristol.ac.uk](mailto:Trevor.Martin@bristol.ac.uk), and [A.Tzanavari@bristol.ac.uk](mailto:A.Tzanavari@bristol.ac.uk)

## Abstract

*Having in mind today's growth of information sources, both in terms of their number and of their size, whether we are referring to the Internet, a corporate intranet, or a library information retrieval system, we can say that manipulating information is not a trivial task. The user is not often being catered for in distributed information systems. He/ she seems to be interacting with systems that do not recognize his/her uniqueness and thus do not offer an individualized treatment. As a result, User Modeling is a core, essential factor in achieving personalization. We present here an intelligent way of inferring user related information that is not available, a situation that is very likely to occur due to sparseness of relevant data. This method can be very useful in recommender systems and this is illustrated with an example.*

Keywords: Intelligent User Modeling, Recommender Systems, Profile Enhancement, Fuzzy Set Theory.

## Introduction

It is an unquestionable fact that ten years ago, software systems had not been developed for a wide range of purposes. There were a limited number of them designed and implemented to perform very specialized tasks. As a result, the people that were going to use them had to belong to the category of experts in the particular scientific area. The systems analyst could thus safely assume the prior knowledge and expertise of the potential users of his/her system. He/she could regard the group of users as uniform.

However, software systems today are being developed for almost every task and are targeting a significantly greater number of users than in the past. Furthermore these users probably do not share common characteristics; the group of users now is diverse.

Treating every user in the same way, not identifying him/her as a different user to interact with, who has different needs, preferences and interests, seems to be a limitation. The user feels he/she is dealing with a passive system that is querying a passive database. He/she doesn't

---

Material published as part of this proceedings, either on-line or in print, is copyrighted by the author with permission granted to the publisher of Informing Science for this printing. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the author.

feel any personalization, either in the form of recommendation, or individualized interaction.

Distributed Information Systems are heterogeneous collections of electronic networked information resources, targeting diverse communities of users e.g. the Internet, corporate intranets, databases, library information retrieval systems, etc. These systems seem to have the limitations mentioned above.

In order to overcome these, User Modeling (Kobsa, 1993) techniques are necessary. These will provide the system with information about individual users that is crucial for achieving personalized interaction. This information will possibly be sparse and the system might need to make inferences about a particular user. Any decisions that the system makes though will be true with a certain degree (probability) since we are dealing with human beings, who are difficult to model. Indeed, humans themselves find it difficult to understand, make assumptions or predict other humans and so we can safely claim that a computer should at least be uncertain about its decisions.

In this paper, we propose a novel way of representing the user's model with Conceptual Graphs (Sowa, 1984), a way to make inferences based on sparse data and to find the probability of truth mentioned earlier and we give an example on how the system can make suggestions to the user. Fuzzy Set Theory (Zadeh, 1965) can be proved to be powerful in this problem area, in which essentially most of the concepts are a bit abstract and relative. The inference

mechanism is based on this and Baldwin's Mass Assignment Theory (Baldwin, 1991). The example given is a food retail company and its system's suggestions to customers.

## User Modeling

We can categorize User Modeling (McTear, 1993) techniques in the following way:

- Statistical, keyword-based approaches
- Artificial Intelligence and Neural Network techniques
  - Social filtering (Shardanand & Maes, 1995) (a non-explicit approach, using the preferences of other users rather than an explicit representation of a profile)

The first two categories can be considered content-based methods (Lang, 1995). We could say that our approach mainly lies under the third category and less under the first one, combining the advantages of the two. In social filtering, the system constructs rating profiles of its users, locates other users with similar rating profiles and returns items that the similar users rated highly. Ratings sparseness is an issue in social filtering. User profiles in this case are usually sparse vectors of ratings. This problem is not present in our approach, since the user's profile is enhanced based on the prototypical user profiles, which allows the initial state of having very limited knowledge about the user.

The advantage of social-filtering techniques compared to the others, is that the user's profile does not only include topics or keywords he/she searches for - it also includes topics that other users that belong to the same "interest group" like. This can be proved to be instrumental in enhancing the user's model. Substantial work in this area and in knowledge management in general, has been done in British Telecom Laboratories (Davies and Revett, 1997; Davies, Weeks & Revett, 1995).

Furthermore, feeding back the relevance of results has appeared to be a task the user considers tedious and avoids and thus in our approach it is not required from the user. A large variety of agents that make use of machine learning techniques have been developed and presented in the literature (e.g. Pazzani & Billsus, 1997). Feedback is necessary in machine learning techniques that need it for their "learning" process. This process involves computation to learn the user's preferences, which however will not be static. This means that it will need to be repeated with a frequency that depends on the frequency that the user's model changes.

There has been work done on social-filtering methods and on content-based methods in the recommender systems framework. In the latter case the system, based on properties of the interesting items, returns those items that are most similar to the ones that were highly rated. The limitation here is that there are no new topics explored, only the ones already in the profile, or similar. This is something that does not characterize our approach where profiles can be enhanced.

## Recommender Systems

The early recommender systems provided services like recommending to the Internet user films, articles and music taking the person's personality into account (Shardanand & Maes, 1995). His/her personality in a given domain would in other words be his/her preferences. They had statistical algorithms that learnt to identify groups of users with similar interests and provided personalized recommendations based on what other users rated highly. Nowadays, on-line companies such as Amazon.com have employed this technique.

Additionally, there have been recommender systems developed for filtering documents on the Internet (Delgado, Ishii & Ura, 1998) and for sharing knowledge in an organization (Davies & Revett, 1997; Glance, Arregui & Dardenne, 1998).

## Our Approach

It is important to recognize that we will only have sparse information about a particular user and while this will build up over time, we should not expect to be able to use learning methods that require large amounts of data to provide the user model. Essentially, this means that we need to find a way to make inference when we only have sparse data available.

In human terms we do this all the time. We meet someone and make decisions about that person with reference to a set of prototypical persons, which we have built over time. For example, if we knew someone that worked in a software house, we could assume with a certain degree of confidence that he would know how to program. We should collect data for building up clusters of types of users, according to their behavior, their abilities, needs, etc. and use these clusters as prototypes (Rich, 1979). The way in which the system acquires relevant information to build them is a separate problem. It can produce them by extracting information from a database of individuals, or the user

## An Intelligent Method for Inferring Information about the User/Client

can provide it with his understanding of prototypes or refine existing ones. For our purpose, we will assume that we have a collection of prototypical people.

As for the user's model, the easiest way would be to ask him directly to pick a model (categorize himself) from an existing model database. That would leave everything to the user. Another way would be to ask the user for relevant information, possibly giving him a set of choices for each characteristic we are interested in. Finally, the most advanced method and at the same time the least demanding from the user's point of view, would be creating the model by learning the necessary information, or by using the information available.

The next question that we need to answer is how can the system acquire relevant information about the user. By observing the user's real-time interaction with the system, we can learn quite a lot about him, his preferences, his interests, his habits, etc. Additionally, all the systems that use user-profiling techniques, should give their user a certain degree of freedom as well as allow him to get involved with improving the system's performance. Consequently, the user should be able to give feedback concerning the system's actions and should be able to improve/change his profile at any time.

In the application we have developed, everything the user types is collected, processed linguistically and used to form a log file. This is where we obtain the user related information to form his/her profile. The use of a log file may potentially address the user privacy issue, if the user is given the ability to state when he/she wants his/her actions to be logged, or in other words be monitored.

The representation of the clusters/ prototypes we mentioned earlier can be done in different ways. One approach would be to represent each person in the cluster as a vector point and to find the average vector for the cluster and this would represent our prototype for the cluster. Having a collection of prototypes represented in this way, when presented with a particular user, we would match him/her to the nearest prototype vector. However, we consider this to be a rather simplistic form of representation because first of all it does not allow us to consider relationships between concepts, but only basic attributes. In addition, there is a lack of flexibility as a result of having just a single point representing a prototype. Consequently, we decided to use Conceptual Graphs (Sowa, 1984) that satisfy both of the above requirements. In this way, we will be able to represent a cluster as a set of attributes and their relations.

Furthermore, Conceptual Graphs have been proven suitable for efficiently describing attributes with an intrinsic vagueness as fuzzy sets.

Fuzzy Set Theory (Zadeh, 1965) is based on the notion of a fuzzy set. Zadeh argued that humans reason not in terms of discrete symbols and numbers, but in terms of fuzzy sets. An element in Set Theory either belongs to a given set or not. However, the same element can belong to a fuzzy set with a certain degree. There are three properties that should be mentioned. The domain, which is the range of values that the fuzzy set, is mapped on. The degree of membership axis, from 0 to 1, that measures a value's membership in the set and the surface of the fuzzy set (Fig. 1).

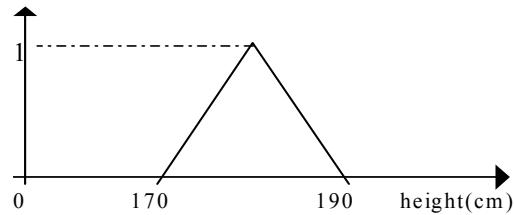


Figure 1. Fuzzy set “tall”

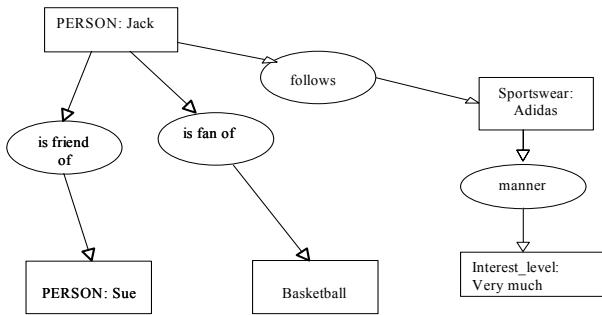
The philosophy of our approach follows, with the possibility of variations depending on the application area. We will collect information on users based on their interaction with the computer and will divide them into clusters. The latter will be generic categories of people with similar behavior and characteristics. Those characteristics will be captured in the cluster's definition. Each cluster will be defined by the relevant fuzzy set attributes and relationships between them and will be represented by a conceptual graph. This will form an individual prototype. In the case of non-numerical attributes, these will form a concept node on their own in the graph, with no instantiation. A new person will be checked against this prototype to decide how closely it matches it. He/she will also be checked against all the other prototypes that include information that is of interest. By doing this we assume that a user that partially matches a certain prototypical person's behavior, will probably possess other features of that specific prototype as well. The computer will be able, when missing some information about its user, to deduce it from prototypical users with similar behavior. This form of reasoning will be inductive or even analogical and will have no truth guarantee.

We must mention at this point that the whole inference mechanism described has been implemented in FRIL (Baldwin, Martin & Pilsworth, 1995; Baldwin, 1993) and all the core operations and functions have been integrated in a top-level coordinating C program.

## Conceptual Graphs

In the human mental process of perception, information can be saved in a number of forms like images, language and concepts. When seeing an image, perception only identifies the key concepts of that image. The way in which an English person and a Greek person will perceive an image is the same. In the case where they want to speak about what they saw, the concepts are translated in words – different to each language.

Concepts can be organized in a graph in order for them to be meaningful and can also be arranged to form sentences. This form of knowledge representation is called a Conceptual Graph (Sowa, 1984). Concepts are linked with relations and arcs point the flow of the sentence's meaning (Fig. 2). Concept nodes have one or two parts: a type, which is the concept's name and possibly a referent, which instantiates the concept to a value, or a set of values. There are a number of operations that can be performed and rules that need to be followed.



**Figure 2. A Conceptual Graph**

It is obvious that this is a logical formalism that closely relates to Natural Language. This can be instrumental when needing an efficient processing paradigm for a computer. The task of dealing with the semantics of vague linguistic terms, led to the exploration of using Fuzzy Logic (Zadeh, 1983) in Conceptual Graphs (Baldwin & Morton, 1985; Morton, 1987; Wuwongse & Manzano, 1993; Cao, 2000). The imprecision of words like “most”, “high”, “little” can be handled in the graphs by instantiating concepts to fuzzy sets.

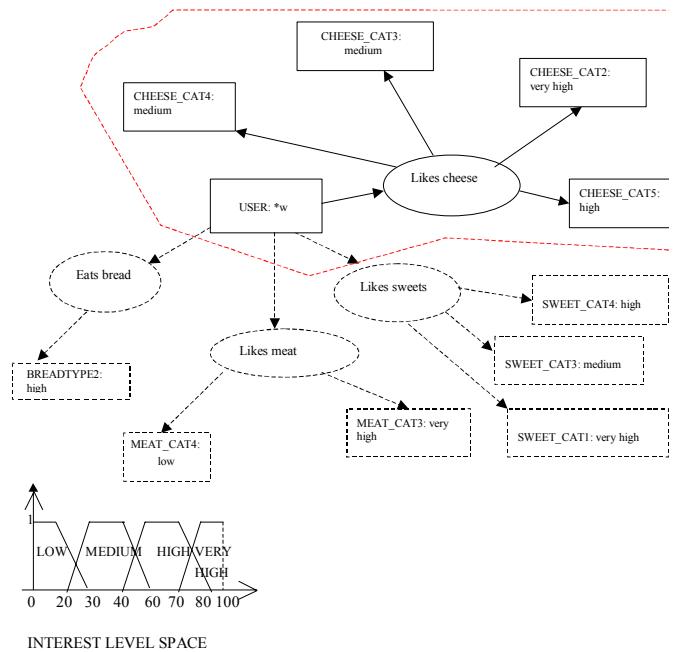
## Applications

The detailed outline of the inductive inference mechanism can be found in (Baldwin, Martin & Tzanavari, 2000a) along with an example and its application to the Forum project (McGrath 1998). Its application to British Telecom Laboratories' Search Engine has been presented in (Baldwin, Martin & Tzanavari, 2000b) with a description of the theoretical aspects behind the approach.

This novel inference method has been used in a search application, but can naturally be used in several other domains. In Customer Relationship Management (CRM) this can be proved to have advantages in recommending new products, or new services that the user/client would not have explored yet. The prototypes in this case will be groups of users/ clients that share common interests in the particular company's services' context. Here we present a relevant example to illustrate the method itself.

## Example

Let us consider a company that deals with food retailing and that also allows its customers to shop online. We will assume that after studying people in this context, we derived a set of prototype conceptual graphs. We decided for the purpose of this example to categorize clients according to their preferred cuisine (e.g. English, French, Greek,



**Figure 3. Prototype graph P and P1 (within the dotted line area) and the Interest Level Space**

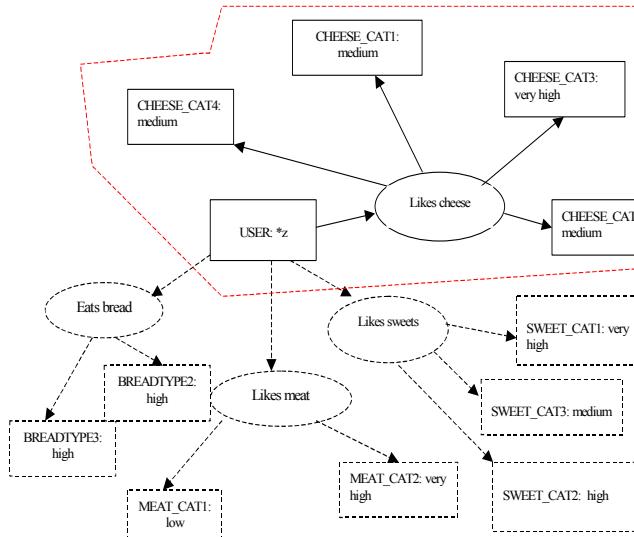
## An Intelligent Method for Inferring Information about the User/Client

etc.). People with similar preferences and tastes were clustered to the same graph. Furthermore, the user's graph is developed after collecting relevant information about him/her. This information can be acquired in several ways: observation of the real-time interaction between him/her and the system (what products he/she purchases), some direct user-feedback concerning the system's actions, or by other means which will not be dealt with in this paper. The mechanism that was used in order to reach the conceptual graph format from the data available to us in the Search Engine application is presented in (Baldwin, Martin & Tzanavari, 2001).

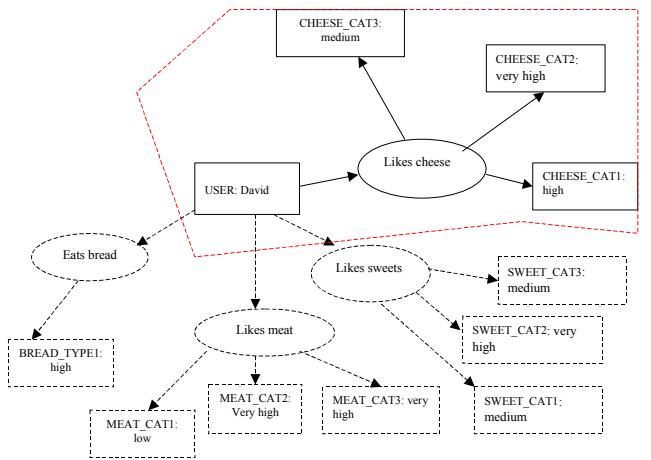
We will refer to the prototype graphs as P1, P2, ... Pn and to the user's graph as S. Examples of these can be seen in figures 3, 4 and 5. The graphs include the different categories of sweets, cheese, meat and bread the customer prefers, depending on the cuisine he/she likes. The instantiations denote the level of that interest/preference in terms of fuzzy sets (Fig. 3).

Prototype graphs P1, P2 will be representative of the prototype graph category. For example they can represent English and Greek cuisine. In our example we will only use two graphs for illustrative purposes. However, in reality there will be a significantly greater number.

As we mentioned above, the computer will not always have all the necessary information on a particular user in order to take action on his behalf. Its knowledge will be limited, so it should make inferences from the information



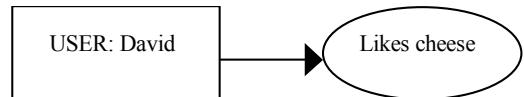
**Figure 4. Prototype graph P2 and P2' (within the dotted line area)**



**Figure 5. User's graph S and S' (within the dotted line area)**

it has available. If the user were one of the prototypes, then the answer would be given by part of the respective graph.

The information that the computer will be looking for could be expressed in a query/question format when using natural language. For our example, let us suppose we would like to identify more topics that would be interesting for our customer based on the prototypical users he/she resembles and in particular topics related to the "likes cheese" domain. We will need a query graph Q like the one in figure 6. To construct Q, we must use concept and relation node labels that exist in the prototypes as well, in order to achieve some matching.



**Figure 6. Question/ Query Graph Q**

We will match graph Q to each of P1, P2...Pn – this will correspond to a maximal join operation – all nodes in Q should match corresponding nodes in P1, P2...Pn. If nodes exist in Q that do not find matching nodes in a certain prototype graph, then that graph is not capable of providing us with the information we are looking for and so we do not consider it any further. For each graph, we identify the subgraph that corresponds to the domain that is relevant to our query and strip the remaining concept and relation nodes (these appear in dotted line on the figures). The new

graphs obtained will be called P'1, P'2...P'n and can be seen within the dotted line area on the figures.

The graph S corresponding to the specific user is also treated in this way. Q is mapped onto S and S is stripped of non-relevant nodes to give S'. We now take S' and match onto P'1, P'2...P'n by means of projection. This results to graphs P"1, P"2...P"n. These matchings will not be complete and some measure of completeness will be used to give a support for how well S' matches each of P'1, P'2...P'n. When we talk about matching two graphs, we mean matching one's relations and concepts against the other's. Two relations match when their type is the same. Two concepts match with a support  $s=1$  when their types and referents are identical and with a support  $s<1$  when their types are the same, but their referents different. Since we consider the case where referents can be fuzzy sets as well, in that specific case we perform a matching between fuzzy sets. Point Semantic Unification (Baldwin, 1993), which is based on the Mass Assignment Theory (Baldwin, 1991), is used to obtain the support of matching. Because we get a support from each pair of concepts that is matched, we accept the overall support to be the mean of the individual supports. Consequently, we end up with a support for each pair S' and P'n. Let these supports be given by s1, s2...sn. We can now pick out the parts of P"1, P"2...P"n that correspond to our answer. We do this by projecting P"1, P"2...P"n on to our user graph and identifying the part of the former that does not project on anything. In our example the answers will be CHEESE\_CAT4 from Prototype1 and CHEESE\_CAT5 and CHEESE\_CAT4 from Prototype2, instantiated to fuzzy sets on the Interest Level space.

At this stage we have a set of answer concepts, their instantiations and an accompanying support for each one (s1, s2...sn, mentioned previously). It is now necessary to fuse these to obtain a final answer because answers from different prototypes may overlap (e.g. CHEESE\_CAT4). This will be done with a combing scheme, based on the Mass Assignment Theory that will take into account the supports as well.

We will end up with concept node names e.g. CHEESE\_CAT5, CHEESE\_CAT4 with linguistic variables assigned like "very\_high", "medium". These will constitute the answer to our problem of enhancing the user graph's "likes cheese" branch, or in other words identifying new topics of interest based on the prototypical users he/she resembles.

We could now suggest these new topics/ categories as some that might be of interest to the user. This can take the form of new products or services suggestions when talking about a CRM context, or of profile enhancement when talking about User Modeling. An example taken from the Search Engine application can be seen in figure 7.

**Figure 7. Example of Profile Enhancement**

In the case where the system will be missing a particular piece of information about the user, possibly in order to take some action, it can form a query, which would be more complete than the one in figure 6. It might for example include another concept node whose instantiation will represent the missing information. In this case the same inference mechanism will be applied, however the difference is that there will be a single answer (Baldwin, Martin & Tzanavari, 2000a).

At all times, the user will be able to view his/her profile in the Conceptual Graph format and have the option to edit it. This is another reason why Conceptual Graphs were selected. Pictorial representation of the user's model is more attractive to users and easier to manipulate.

## Conclusions

The strength of the method lies in the fact that it gives a solution to the problem of having sparse information about the user. This is a quite common situation especially when dealing with users/customers who are new. A system must have a mechanism for inferring the information it is missing about the user in order to reach conclusions and potentially make decisions. There are several ways that it can be applied, like for example to recommend something new to the user, or to identify other users with similar in-

## An Intelligent Method for Inferring Information about the User/Client

terests, or to enhance his/her profile. In the Search Engine's User Modeling component that we have developed, the user's model is created from a log file, which includes information that is collected from a number of different sources. As a result, the user's profile can never be empty, even at the start of the application (cold-start).

On the knowledge representation side, Conceptual Graphs (Sowa, 1984) are chosen because of their strength of being a graphical language that has a logical formalism. Their origin in human perception, along with their close relationship to Natural Language are only some of their advantages (Cao, 2000).

Finally, Fuzzy Set Theory (Zadeh 1965) is fundamental in order to capture the uncertainty that is involved in every aspect of the system, whether that is the linguistic query terms, or the suggestions to the user. The inductive inference method itself employs computational models that are based on the Mass Assignment Theory (Baldwin, 1991), which is widely accepted for its strength in approximate reasoning.

In personalization systems it is hard to determine how good their performance is, due to the fact that this procedure of evaluating will involve purely subjective judgments. The users would be asked to state how well the system was able to describe them by commenting on its actions, its recommendations. This will form the next phase of our research, along with further experimentation to point out that incorporating User Modeling in areas like search can improve performance.

## References

- Baldwin, J.F., Martin, T.P. & Pilsworth, B.W. (1995). FRIL - Fuzzy and Evidential Reasoning in AI. Taunton, England: Research Studies Press, John Wiley.
- Baldwin, J.F., Martin, T.P. & Tzanavari, A. (2000)a. User Modeling Using Conceptual Graphs for Intelligent Agents. In: B. Ganter and G.W. Mineau (eds.): *Conceptual Structures: Logical, Linguistic, and Computational Issues*. LNAI 1867, Springer Verlag, pp. 193-206.
- Baldwin, J.F., Martin, T.P. & Tzanavari, A. (2000)b. Fuzzy User Modeling for Agents. Submitted to *User Modeling and User-Adapted Interaction*.
- Baldwin, J.F., Martin, T.P. & Tzanavari, A. (2001). Fuzzy User Modeling for Query Expansion. Midwest conference in Artificial Intelligence and Cognitive Science, Oxford, OH, USA.
- Baldwin, J.F. & Morton, S.K. (1985). Conceptual Graphs and Fuzzy Qualifiers in Natural Language Interfaces. Research Report ITRC-85, University of Bristol.
- Baldwin, J.F. (1990). Computational models of uncertainty reasoning in expert systems. *Journal of Computers and Mathematics with Applications*, 19(11), pp. 105-119.
- Baldwin, J.F. (1991). A Theory of Mass Assignments for Artificial Intelligence. In: Driankov, D., Eklund, P.W., Ralescu, A.L. (eds.): *Fuzzy Logic and Fuzzy Control*.
- Baldwin, J.F. (1992). Fuzzy and probabilistic uncertainties. In: Shapiro, S.A. (ed.): *Encyclopedia of AI*, John Wiley (2nd ed.), pp. 528-537.
- Baldwin, J.F. (1993). Probabilistic, Fuzzy and Evidential Reasoning in Fril. *Two Decades of Fuzzy Control*, IEE London, pp. 1-20.
- Cao, T.H. (2000). Fuzzy Conceptual Graphs: A Language for Computational Intelligence Approaching Human Expression and Reasoning. In: Sincak, P. et al. (eds.): *The State of the Art in Computational Intelligence, Advances in Soft Computing*, Physica-Verlag (a Springer-Verlag company), pp. 114-120.
- Davies, N.J. & Revett, M.C. (1997). Networked information management. *BT Technology Journal*, 15(2).
- Davies, N.J., Weeks, R. & Revett, M.C. (1995). An Information Agent for WWW. International Conference on the World Wide Web, Boston, USA.
- Delgado, J., Ishii, N. & Ura, T. (1998). Content-based collaborative information filtering: Actively learning to classify and recommend documents. In: Klusch, M., Weiss, B. (eds.): *Cooperative Information Agents II. Learning, Mobility and Electronic Commerce for Information Discovery on the Internet*. Springer-Verlag, LNAI 1435.
- Glance, N., Arrengui, D. & Dardenne, M. (1998). Knowledge Pump: Supporting the Flow and Use of Knowledge. In: Borghoff, U. & Pareschi, R. (eds.): *Information Technology for Knowledge Management*. New York: Springer-Verlag.
- Kobsa, A. (1993). User Modeling: Recent Work, prospects and hazards. In: Schneider-Hufschmidt, M., Kuhme, T., Malinowski, U. (eds.): *Adaptive User Interfaces: Principles and Practice*. Amsterdam: Elsevier Science Publishers B.V., pp. 111-128.
- Lang, K. (1995). NewsWeeder: Learning to filter netnews. Twelfth International Conference on Machine Learning, Lake Tahoe, California.
- McGrath, A. (1998). The Forum. SIGGROUP Bulletin 19(3).
- McTear, M. (1993). User modelling for adaptive computer systems: A survey of recent developments. *Artificial Intelligence Review* 7, pp 157-184.
- Morton, S. (1987). Conceptual Graphs and Fuzziness in Artificial

## Baldwin, Martin and Tzanavari

- Intelligence, PhD Thesis, University of Bristol.
- Pazzani, M. & Billsus, D. (1997). Learning and Revising User Profiles: The identification of interesting web sites. *Machine Learning* 27, pp. 313-331.
- Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science*. 3, pp. 329-354.
- Shardanand U. & Maes P. (1995). Social Information Filtering: Algorithms for Automating "Word of Mouth". CHI-95 Conference, Denver, Colorado, USA.
- Sowa, J.F. (1984). Conceptual Structures: Information Processing in Mind and Machine. Massachusetts: Addison-Wesley.
- Wuwongse, V. & Manzano, M. (1993). Fuzzy Conceptual Graphs. In Mineau and Moulin (eds.): *Conceptual Graphs for Knowledge Representation*, LNAI No.699, Springer-Verlag, pp. 430-449.
- Zadeh, L.A. (1965), Fuzzy Sets. *Information and Control*. 8, pp. 338-353.
- Zadeh, L.A. (1983). A fuzzy-set-theoretic approach to fuzzy quantifiers in natural languages. *Computers and Mathematics with Applications*. 9, pp 149-184.

## Biographies

Jim Baldwin is Professor of Artificial Intelligence and Director of the AI Research Group at the University of

Bristol. He has been active in fuzzy sets since the early days of the subject, and his research in AI covers fundamental aspects of knowledge representation, inference under uncertainty and belief theories, fuzzy control and fuzzy set theory, and machine learning. He is the originator of support logic programming and the mass assignment theory. He has published over 250 papers, is a member of the editorial board for a number of journals and has served on many conference program committees.

Trevor Martin is a Reader in the AI Research Group at the University of Bristol. His research interests focus on uncertainty in AI, and he is co-developer of the FRIL language and the fuzzy data browser. He has published over 100 papers in refereed journals and conferences, and is a joint organizer of several international fuzzy and fuzzy logic programming workshops.

Aimilia Tzanavari received her BSc in Information Engineering from the Technical Educational Institute of Athens, Greece, and her MSc in Advanced Computing from the University of Bristol. She is currently in the final stage of writing up her PhD thesis in the Artificial Intelligence Group of the same university. Her PhD project is sponsored by British Telecom Laboratories. Her research interests include User Modeling, Intelligent User Interfaces and Artificial Intelligence.