



UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
<http://www.dit.unitn.it>

IMPLICIT CULTURE: AN APPROACH FOR EXPERIENTIAL REASONING

Aliaksandr Birukou and Paolo Giorgini

April 2006

Technical Report DIT-06-020

Implicit Culture: an approach for Experiential Reasoning

Aliaksandr Birukou and Paolo Giorgini¹

Abstract. Usually, people tend to use knowledge acquired with the experience for their daily activities. Very often this knowledge is implicit and highly personalized, and it is very hard to make it explicit in information bases. Transferring knowledge to new community members or making it available for another community is a challenging problem. In this paper, we describe the Implicit Culture approach that uses learning techniques to facilitate the transfer of implicit knowledge and the reuse of previously accumulated experience. We briefly describe current applications of Implicit Culture and show how it relates to Experiential Reasoning.

1 Introduction

Groups, or *communities*, of people tend to assume specific behaviors in their daily activities. These behaviors are very often induced by the specific knowledge acquired with the experience, and independently of the changes, it is very important to preserve them. For example, the process of software release should not depend on the people currently working in the company [9]. On the other hand, groups of agents can benefit from using knowledge of other groups. For example, PhD students usually would use the knowledge of their advisors about the state-of-the-art in the research field. As reported in [6], the implicit knowledge transfer in communities of practice is an important problem, which, unfortunately, becomes very hard since very often knowledge is highly personalized and hardly representable explicitly by means of information bases (knowledge is usually embodied in the skills and the experience of the community members).

Implicit Culture [5] deals with the problem of knowledge transfer between two communities of human or artificial agents acting in similar conditions. This problem can be also seen as the problem of transferring experience between community members. Systems for Implicit Culture Support (SICS's) observe situations faced by the agents and memorize which actions are adopted in these situations. This information is then analyzed using learning techniques, which can be considered as a particular case of reasoning on data, and the information is exploited to create suggestions for the agents which face situations similar to those faced previously by the community members. Implicit Culture is intended to be used as a framework for computer-human, or, more generally, computer-agent interactions, that supports, preserves, and encourages desired behavior of the members of communities according to the extracted knowledge.

The paper has the following structure. In Section 2, we describe the Implicit Culture approach. Section 3 shows where it has been applied and, finally, Section 4 describes how it is related to Experiential Reasoning.

2 Implicit Culture in short

Implicit Culture ideas have been introduced [5] as a generalization of Collaborative Filtering [7]. Implicit Culture combines multi-agent

systems and learning techniques (data mining or machine learning). The main steps of the Implicit Culture approach are the following: the behavior of a group of people in different situations is observed; the behavior is then analyzed and some behavioral patterns are discovered; patterns are used to support another (or the same) group of people to behave similarly to the observed group. All this allows a person to use information about others' behavior in similar situations.

The Implicit Culture approach is implemented by Systems for Implicit Culture Support (SICS's). A SICS consists of the three modules: the *observer*, which records actions and situations where the actions are performed; the *inductive module*, which induces a theory about the actions and the situations where actions are performed by applying learning techniques to the data stored by the observer; and, the *composer*, which analyzes observations to find similar situations and to suggest actions consistent with the theory learnt by the inductive module. The theory can represent desired patterns of behavior.

Implicit Culture is defined as the relation between two groups of agents such that the agents belonging to a group behave consistently with the "culture" of the agents belonging to another one [5]. Implicit Culture takes its name from the idea of learning hidden laws of the community behavior, its "culture", non-obvious sequences of actions which execution can involve the use of implicit knowledge and experience of the community members. The application of the patterns of behaviors to the second group is also done implicitly, since agents of the second group do not need to know anything about the first group and its behavior.

3 Current Applications

The Implicit Culture ideas have been applied in several information systems [2, 3, 4, 8]. For instance, the agent-based recommendation system for supporting communities of people in searching the web is presented in [4]. The system aims at providing support to the communities of people whose web search is guided by specific interests, e.g. related to their professional activities.

Another application of Implicit Culture is the agent-based system that facilitates scientific publications search [2], where users interacting with their personal agents produce a transfer of knowledge about relevant publications from experts to beginners. Each personal agent observes how publications are used and induces behavioral patterns that are used to create more effective recommendations. Feedback exchange allows agents to share their knowledge and virtual communities of experts can be created to support novice users.

The concepts of Implicit Culture are also applied to support the work of biologists in their laboratories [8]. Adopting this approach the system described in the paper tries to make the knowledge, mainly stored in biologists' notebooks, more useful.

Finally, Implicit Culture is adopted in the system that supports programmers in choosing the design pattern suitable for the given problem [3]. Personal agents in the system produce knowledge transfer among users, allowing for the reuse of experience in choosing design patterns.

¹ University of Trento, Italy, email: {birukou,pgiorgio}@dit.unitn.it

The preliminary evaluation of the system has been performed and the results obtained so far look promising. For the recommendation system for web search it has been shown that the increase of community members causes the increase of the recall of recommendations produced by agents, keeping the precision on a rather high level. It proves that our way of complementing search engine with suggestions, produced as a result of collaboration, makes sense and allows performing web search in a more qualitative way. As for the agent-based system for the search of scientific publications, the experimental results obtained have proven that the SICS is able to adapt suggestions to the user's preferences.

4 Relation to Experiential Reasoning

Consistently with Case-Based Reasoning (CBR) [1], where previously experienced situations are called *cases*, Implicit Culture also operates with situations and the specific knowledge of these situations contributes to the solution of new problems. Implicit Culture also implements one of the most important feature which makes CBR different from other major AI approaches, namely the fact that it is “[...]an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems[...]” [1]. The main differences between the two approaches are the following: while CBR solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation, Implicit Culture helps to solve a problem by using information and knowledge about previous similar situations, but not produces the solution directly; in CBR, the problem is represented explicitly (the case), while Implicit Culture does not deal with an explicit representation of the problem, but only with the implicit information about it.

The reuse of the experience about actions performed in similar situation makes Implicit Culture related to Experiential Reasoning. The essential feature of our approach is that it allows to avoid acting “from scratch”, rather it allows to exploit the knowledge about others' behavior in similar situations. In particular, in the application of Implicit Culture to the selection of design patterns [3], the system offers to an unexperienced programmer to describe the design problem and provides a list of similar problems encountered previously and solutions used. In the application of Implicit Culture to the search of scientific publications [2], users such as novice researchers benefit from the exchange of information about the relevance of scientific articles, specifically, permitting to find influential papers. In the recommendation system for web search [4] community members share the experience on using web links when performing a web search related to the specific interests of the community. All this shows that Implicit Culture allows community members to exchange their experience, in particularly, performing the transfer of implicit knowledge.

5 Conclusion

We have described Implicit Culture — the approach dealing with the problem of knowledge transfer in the communities of agents. The applications of the approach in several domains are described and the relation to Experiential Reasoning is expressed.

We are currently working on the formalization of Implicit Culture, where we would like to synthesize our experience in developing and using applications based on these ideas.

ACKNOWLEDGEMENTS

We would like to thank Enrico Blanzieri, one of the founders of Implicit Culture, for his valuable comments on the paper.

This work is funded by research projects EU SERENITY “System Engineering for Security and Dependability”, COFIN “Artificial Intelligence Techniques for the Retrieval of High Quality Information on the Web” and by Fondo Progetti PAT, MOSTRO “Modeling Security and Trust Relationships within Organizations” and QUIEW (Quality-based indexing of the Web), art. 9, Legge Provinciale 3/2000, DGP n. 1587 dd. 09/07/04.

REFERENCES

- [1] Agnar Aamodt and Eric Plaza, ‘Case-based reasoning: Foundational issues, methodological variations, and system approaches’, *AI Communications*, **7**(1), 39–59, (March 1994).
- [2] Alexander Birukou, Enrico Blanzieri, and Paolo Giorgini, ‘A multi-agent system that facilitates scientific publications search’, in *AAMAS '06. To appear*, (2006).
- [3] Aliaksandr Birukou, Enrico Blanzieri, and Paolo Giorgini, ‘Choosing the right design pattern: an implicit culture approach’, in *Proceedings of the Workshop on Multi-Agent Systems and Simulation (MAS&S) at the Industrial Simulation Conference 2006 (ISC-2006)*. To appear. Available at <http://eprints.biblio.unitn.it/archive/00000960/>, (June 2006).
- [4] Alexander Birukov, Enrico Blanzieri, and Paolo Giorgini, ‘Implicit: An agent-based recommendation system for web search’, in *AAMAS '05*, pp. 618–624. ACM Press, (2005).
- [5] Enrico Blanzieri and Paolo Giorgini, ‘From collaborative filtering to implicit culture: a general agent-based framework’, in *Proc. of the Workshop on Agents and Recommender Systems*, Barcellona, (2000).
- [6] Patricia Gongla and Christine R. Rizzuto, ‘Evolving communities of practice: IBM global services experience’, *IBM Syst. J.*, **40**(4), 842–862, (2001).
- [7] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl, ‘GroupLens: an open architecture for collaborative filtering of netnews’, in *CSCW '94*, pp. 175–186. ACM Press, (1994).
- [8] Marcello Sarini, Enrico Blanzieri, Paolo Giorgini, and Claudio Moser, ‘From actions to suggestions: supporting the work of biologists through laboratory notebooks’, in *Proceedings of 6th International Conference on the Design of Cooperative Systems (COOP2004)*, pp. 131–146, French Riviera, France, (2004). IOS Press.
- [9] Ian Sommerville, *Software engineering (7th ed.)*, Addison-Wesley, Boston, MA, USA, 2004.