E-banking Services for Automated Agent-based Trading

Monique Calisti
Whitestein Technologies AG
Gotthardstrasse 50
8002 Zurich, Switzerland
mca@whitestein.com

Emma Rollon
Laboratoire d'Intelligence Artificielle
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
emma.rollon@epfl.ch

Steven Willmott
Laboratoire d'Intelligence Artificielle
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
steven.willmott@epfl.ch

ABSTRACT

This paper describes a prototype agent-based framework providing e-banking services to support various business transactions in software environments. We outline the system developed and discuss some of the issues that need to be considered and solved for an effective deployment of agent-based payment services in real applications.

1. INTRODUCTION

Internet technologies are rapidly evolving and radically modifying the way people interact with each other. In particular, the increasing number of virtual market places facilitates trading transactions by bringing together a vast number of potential buyers and sellers. In this context, business interactions are moving toward more dynamic and automated solutions and electronic *payment methods* play a key role for all forms of on-line business [3]. Nowadays, despite the growing deployment of electronic banking systems that allow some degree of automation [1], Web interfaces or ad hoc tools still require a large degree of human interactions. Transactions and payment orders, for instance, can be performed electronically, but only if a human is entering the right code or pressing the right button in a specific graphical interface. This kind of approach is evolving from customer self-service channels to more automated and fully integrated business applications that drive products and services to the consumer. Analysts predict that in the near future people will not physically go to their financial institution branches nor log on to the Internet to deal with their banking tasks. Instead, humans will delegate the management of their bank accounts, paycheck, investments, insurances, mortgages, loans and credits to their personal electronic financial assistant[1]. The key concept is that self-interested software entities acting on behalf of humans and/or service providers can automate several electronic business and commercial activities such as service advertisement, market trend monitoring, services pricing and negotiation [4]. This scenario facilitates *banking services* to be increasingly incorporated as part of larger packages and organised around a primary product [2] (e.g., credit or insurance in conjunction with a house or a car purchase).

In this paper, we present and discuss the *Banking Agent-based Transaction* (BAT) system, i.e., a virtual institution offering banking services to the agents accessing the Agentcities market place. The Agentcities initiative[2] aims at defining an open and distributed framework to create an on-line network in which autonomous and heterogeneous agents can supply and/or provide a variety of services. In particular, the integration of business services into the network has the objective of enabling agents to trade with one another and form dynamic, compound 'value added' services. In this context, the BAT prototype provides a generic way for agents to make payments to one another. The expression 'banking services' refers to the set of processes and mechanisms required for (1) enabling agents to make / receive side payments; (2) for creating / maintaining / closing bank accounts.

This paper has the twofold objective of:

- Describing the key components that have been defined to design and thereby implement e-banking services in the Agentcities network (see Section 2).
- Discussing the practical experience of integrating and deploying such services in combination with other services in an open, distributed and heterogeneous environment (see Section 3).

This aims, on one side, to identify the main benefits and challenges of the solution we developed. On the other side, this hopes to stimulate discussion about which aspects should be considered for a better integration of e-banking services with other existing business services.

2. THE BAT FRAMEWORK

In order to enable automated transactions between electronic entities representing different businesses and/or end customers accessing the e-banking institution, a formal and common representation of the available services is needed. Although distinct entities in the market can have different

[1] http://www.rfeurope2001.org/white.html

[2] http://www.agentcities.org

Table 1: Registered Banking Services.

Service Name:	creating-account
Type:	acities.creating-account
Protocol:	FIPA-Request
Ontology:	BankOntology
Language:	FIPA-SL0
Ownership:	EPFL
Service Name:	closing-account
Type:	acities.closing-account
Protocol:	FIPA-Request
Ontology:	BankOntology
Language:	FIPA-SL0
Ownership:	EPFL
Service Name:	getting-account-info
Type:	acities.getting-account-info
Protocol:	FIPA-Query-ref
Ontology:	BankOntology
Language:	FIPA-SL
Ownership:	EPFL
Service Name:	transferring-money
Type:	acities.transferring-money
Protocol:	FIPA-Request
Ontology:	BankOntology
Language:	FIPA-SL0
Ownership:	EPFL

internal representations of services, for a common global understanding, a minimal agreement on common terms, definitions and expressions is required. This agreement corresponds to adopt a common ontology to refer to when speaking about the same concepts. More formally, we assume that:

- *Assumption 1*: All agents communicate with each other by adopting the same standard communication stack. This includes the deployment of FIPA-ACL [8] as communication language, FIPA-SL [5] as content language, and standard FIPA interaction protocols such as fipa-request [6] and fipa-query [7].

- *Assumption 2*: All agents accessing and making use of the banking services have a common understanding of the possible ontologies that need to be used for coordinated interactions. At the moment, this includes a DAML+OIL[3] ontology[4] for banking services and an XML based ontology for an example insurance services.

- *Assumption 3*: In this first version of the BAT institution, all agents trust each other so security services will be included in the next version.

2.1 Banking Services Design

The expression "banking services" refers to the set of processes and mechanisms that a virtual agent-based banking institution offers to the agents accessing Agentcities market place services. As anticipated in Section 1, this includes:

1. Electronic payment service for enabling agents to make / receive payments.

2. Account Management service for creating / maintaining / closing bank accounts.

The banking service design consists therefore of two main sub-sets that are described in the following as two distinct frameworks.

2.2 Agents Roles

Three main types of agents are implemented in e-banking application. In this framework, distinct software entities mainly differ because of the different roles they can cover and/or services they can offer during trading transactions.

1. Bank Agent (BA) is acting on behalf of a banking institute. Every BA offers mainly two kind of services: 1) Account management service that includes the open account, close account and list account information operations, 2) Electronic payment service in which the transfer of funds between two accounts is performed. This entity can therefore be considered as the agent interface of the BAT institution toward the external world.

2. Personal Agent (PA) can be considered as a personal assistant that can act on behalf of final end users and use the services offered in the AGC-market.

3. Insurance Agent (IA) represents an insurance business offering, mainly, to sell an insurance policy. For the moment, only the credit card based payment is provided.

These last two entities represent agents making use of banking services, i.e., specific possible customers of the BAT institution. Of course, a given agent can play different roles considering different goods and/or services in the market. Therefore, an agent can be both *buyer* and *seller*, *customer* or *provider* depending on the specific transaction. However, in order to facilitate the tasks decomposition, in the following PAs act mainly as buyers and IAs act essentially as sellers.

2.3 Agents Interactions

In the following, we list the scenarios that are developed for defining and verifying the specific mechanisms needed and offered by the different types of agents depicted above.

Scenario 1: opening an account. The scenario starts when a PA-X wants to open an account within a given bank. PA-X requests the BA-Y to open an account including in the request message the information needed. The BA-Y will send back to PA-X the result of his demand. The message in Figure 1 is an example of the performative that an agent acting on behalf of a bank customer would use in order to ask for the creation of a bank account. The recipient is the agent representing the BAT institution.

Scenario 2: closing an account. The scenario starts when a PA-X wants to close an account within a given bank. PA-X requests the BA-Y to close an account. BA-Y will then verify if the given account belongs to the agent PA-X and, and after having performed the required action, BA-Y will send back to PA-X the corresponding action's result.

[3] http://www.daml.org/2001/03/daml+oil-index.html
[4] http://www.lausanne.agentciites.net/

```
(request
      :sender (agent-identifier :name user@sample.lausanne.agentcities.org)
      :receiver (set (agent-identifier :name bank@sample.lausanne.agentcities.org))
      :content ((action
(agent-identifier :name bank@sample.lausanne.agentcities.org)
(create-account (
user@sample.lausanne.agentcities.org
(personal-info
      :pid 4096788
      :firstname Lluis
      :lastname Perez
)
(account
      :type checking
      :total_credit 5000
))
            )))
)
```

Figure 1: A FIPA-ACL message to request an account to be opened.

```
(query-ref
      :sender (agent-identifier :name user@sample.lausanne.agentcities.org)
      :receiver (set (agent-identifier :name bank@sample.lausanne.agentcities.org))
      :content ((all ?x
 (is-account-description
            ?x
user@sample.lausanne.agentcities.org
4096788
1
            )))
)
```

Figure 2: A FIPA-ACL message to query for account information.

Scenario 3: getting information from an account. The scenario starts when a PA-X asks for account/s information (Figure 2). After BA-Y receives the query and verifies the ownership of the account, the result is sent back to PA-X. The message is either an error notification or the requested information (see Figure 3). If the account identifier is not specified, the query is interpreted as a query for information about the full list of accounts owned by PA-X.

Scenario 4: credit-card based payment. PA-X stipulates an insurance policy with IA-T for 500 CHF. PA-X sends its credit card details to IA-T that requests the BAT institution to transfer 500 CHF from PA-X's account to IA-T's account. Both PA-X and IA-T have accounts with BA-Y (i.e., the same bank agent managing both accounts) and are informed about the transaction. The agent's tasks are divided as follows.

- *Negotiation phase.* The negotiation terminates successfully if the two agents find a common agreement about the insurance policy conditions including its price. Depending on the specific policy that agents have been negotiation about, the final agreement can include different information. Whenever an agreement is found, PA-X (buyer) has to pay IA-T (seller) and the seller has to supply the service to the buyer. The payment is done by making use of BA-Y services.
- *Starting a payment procedure.* In this case, IA-T gets the PA-X's account needed information and requests BA-Y to transfer the specified amount of money to its account. BA-Y determines if the transfer will be an intra- or inter-bank transfer. We assume here to consider an intra-bank scenario.
- *Making a payment.* BA-Y verifies if PA-X's account contains enough money to make the payment. If this is the case, the money are transferred from the buyer's account to the seller's one and both PA-X and IA-T are notified about the transaction. In case there is not enough money in the buyer's checking account, BA-Y notifies both agents.

```
(inform
      :sender (agent-identifier :name bank@sample.lausanne.agentcities.org)
      :receiver (set (agent-identifier :name user@sample.lausanne.agentcities.org))
      :content ((= (all ?x
      (is-account-description
            ?x
user@sample.lausanne.agentcities.org
4096788
1
            ))
            (set (account
                  :aid 1
                  :type checking
                  :total-credit 5000
                  :credit-interest-rate 2
                  :debit-interest-rate 1
                  :creation-date 24-01-2002
            )
      ))
)
```

Figure 3: A FIPA-ACL message to send information about a specific account.

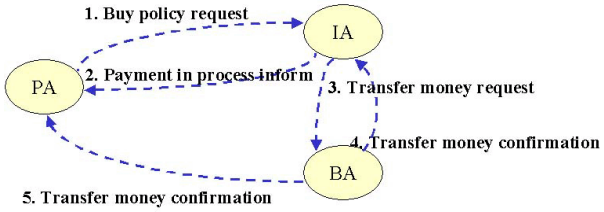

Figure 4: Nested interactions involving IA, BA and PA.

2.4 Deployment

The first BAT prototype has now been deployed in the Agentcities network and can be accessed by third parties using the agent interfaces provided. More details can be found at http://www.lausanne.agentcities.net/.

3. OPEN ISSUES

While the current bank services are very simple and still require extensive testing they do demonstrate that it is possible to represent a first step towards developing the robust services that would be required to support an effective agent economy. Possible extensions of this work include:

- Refinement of agent interfaces and payment functionality.
- Addition of Web accessible interfaces (to enable owners of agents using the banking services to access account information).
- Ensuring interoperability with market place and content services being developed in the European Union 5th Framework funded Agentcities.RTD project.[5]
- Addition of bank-bank interfaces to enable third parties to develop different types of payment services.

[5]http://www.agentcities.org/EURTD/

- Integration with or modelling of payment services supported by real financial institutions.
- Integration with prototype security services. It is for instance crucial to be able to authenticate agents accessing the bank (i.e., verify that and agent is who is claiming to be) and verify whether an agent has the appropriate rights to perform an action (i.e., transfer money from a given account).

More specifically, concerning security, it is possible to envisage two main levels at which mechanisms are needed. (1) Authentication of an agent that is willing to access the bank. The banks as a virtual institution may define specific policies to be followed when interacting with the bank or with other agents making use of banking services. For instance, in order to access the bank, agents may need to be identified as regular customers or they may be required to introduce themselves and enter specific information. (2) Specific security mechanism/policies at the level of every different service offered within the virtual bank. Hence, for instance, if an agent admitted to the bank (i.e., that has been identified as a customer at the previous level) wants to know the amount of money on a specific account, the bank will verify if the agent has the rights to access this information.

3.1 Discussion

On-line businesses, and in particular e-banking, require complex interactions between diverse systems owned by different organisations or individuals. Despite the development of more robust information and reasoning systems and the evolution of agent infrastructures, major challenges lie ahead various visions of flexible and automated integration between agent-based business systems.

- Integration of agent infrastructures with existing non-agent based environments such as databases, legacy systems, various tools, etc. This task is even more complex when considering diversified access channels - from Internet-only to to embrace mobile (PDAs, wireless, etc.), IVR and ATMs.
- Development of more usable and simple agent communication frameworks. This includes less complex formal semantics and logic models for agent communication languages, content expressions and ontologies so that dynamic and flexible agent coordination and service aggregation can be more easily implemented.
- Security of agent-based interactions. When building business systems relying on electronic platforms and components, such as agents, the notion of trust has to be redefined so that the main characteristics of electronic environment and emerging technologies are taken into account for building an appropriate secure framework. This is particularly crucial for bank-specific services, since authentication, non-repudiation, privacy and confidentiality represent intrinsic requirements that need to be satisfied.

At the moment, we are focusing on the integration of an authentication phase (protocol) that precedes any kind of interaction within the Bank Agent. Once a customer agent has been authenticated, a secure session is opened (i.e., SSL) so that message exchanges (between the external agent and the BA) can satisfy authentication, integrity and confidentiality. However, since this part is still in a preliminary stage and it involves the collaboration of other partners in the Agentcities.RTD Project, more details will be published in the future.

4. CONCLUSIONS

Today, agent technology represents one of the most dynamic fields in which various techniques are used to build distributed systems with intelligent local components designed to both cooperate and coordinate their activities. This approach has been adopted for dealing with many tasks in different domains, such as resource allocation, network management, e-commerce, health care, etc., for its intrinsic capability of representing the decentralised nature of many problems, the existence of multiple control and logic components with distinct roles, the multiple capabilities/roles of distributed and eventually self-interested entities. In the on-line business context, software agents can evaluate and optimise the utility of specific actions (such as bidding, offering, selling, etc.), gather and process huge amounts of information and follow specific strategies more efficiently and more rapidly than humans or traditional mechanisms can. It could be argued that an agent future for Internet-based business transactions will be possible only if agents will be able to properly implement market services (including payment) in software environments. However, it is also true that the current increasing deployment of agent technology in several fields has the opportunity of influencing how services will be supplied in virtual markets. One of the main ideas behind the integration of the BAT prototype in the Agentcities network is the capability of evaluating in a more realistic way:

- What kind of impact these services will have on a larger population of users and vice versa,
- How and at which level service composition will be possible,
- How security mechanisms should work and be implemented,
- How banking services could evolve in such a vast and differentiated electronic environment.

5. ACKNOWLEDGEMENTS

The research described in this paper is partly supported by the EC project Agentcities.RTD, reference IST-2000-28385. The opinions expressed in this paper are those of the authors and are not necessarily those of the Agentcities.RTD partners.

6. REFERENCES

[1] N. Asokan, P. A. Janson, M. Steiner, and M. Waidner. The state of the art in electronic payment systems. *Computer*, 30(9):28–35, 1997.

[2] M. Calisti, D. Deluca, and A. Ladd. An agent-based framework for financial transactions. In *Proceedings of Autonomous Agents 2001 Workshop on Agent-Based Approaches to B2B*, May 2001.

[3] A. Furche and G. Wrightsom. *Computer Money - A Systematic Overview of Electronic Payment Systems.* DPunkt Verlag, 1996.

[4] H. S. Nwana, J. Rosenschein, T. Sandholm, C. Sierra, P. Maes, and R. Guttmann. Agent-mediated electronic commerce: Issues, challenges and some viewpoints. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 189–196, New York, May 9–13, 1998. ACM Press.

[5] XC00008. Fipa sl content language specification. *Foundation for Intelligent Physical Agents*, 2000.

[6] XC00026. Fipa request interaction protocol specification. *Foundation for Intelligent Physical Agents*, 2000.

[7] XC00027. Fipa query interaction protocol specification. *Foundation for Intelligent Physical Agents*, 2000.

[8] XC00061. Fipa acl message structure specification. *Foundation for Intelligent Physical Agents*, 2000.