Using Agents for a Participatory Collaborative Media Sharing Experience

Luigi Benedicenti

and

Songsiri Srisawangrat
Faculty of Engineering, University of Regina
Regina, SK, S4S 0A2, Canada

ABSTRACT

This paper describes the architecture of an agent-based music service that generates dynamic playlists based on suggestions from multiple users. The system employs a number of agents capable of autonomy, mobility, scalability, collaboration and platform independence. In the system, agents are classified into two groups: requesters and music service team. Requesters work on behalf of their users, and bring music suggestions from the users to a DJ agent (DJ) which is part of the service team. The DJ has a reasoning component responsible for music selection and negotiation.

Music selection is the result of negotiation among agents in sealed-bid auctions. The bidding strategy is specified by users. The amount of a bid implies the value of a song for a specific user; the DJ will evaluate this amount to determine the winner of the auction, whose song will be placed on the playlist. All agents interact with each other using an interaction/negotiation protocol, through agent messaging.

A proof of concept implementation of this architecture shows that agents in the A-DJ system automatically perform tasks and collaborate with each other under the proposed protocol. The implementation appears to be scalable up to approximately 800 customers on a general-purpose computer system.

Keywords: Software Agents, Electronic Commerce, Multimedia, Mobile and Distributed Computing, Negotiation Protocols.

1. INTRODUCTION

Participatory behaviour in music listening experiences is often limited: the selection of music to be played often lacks of user participation (e.g., selection by director or D.J.). Participatory selection of music has to be mediated in some way; this can either be completely automated or it can rely on a human mediator. This paper aims at constructing a participatory music selection experience.

A music service that allows users participating in selecting music for a public space becomes possible through the use of mobile agents and personal mobile devices such as cellular telephones and personal digital assistants. This service helps increasing the level of participatory social activities collaborative entertainment. However, supporting these social activities requires dealing with a potentially unpredictable number of users and possible compatibility issue in multi-platforms. Choosing an appropriate mechanism to make the system automatic is also one issue.

The ideal requirements for a music selection system were derived by literature review and consultation with researchers. They are as follows:

1. Music suggestions from multiple users
2. No required human intervention to conduct the music selection
3. Less user effort and convenient for users
4. Support for multiple platforms
5. Ability to operate in sporadic network connectivity conditions
6. Scalability
7. Collaborative music selection

The literature reports on several experimental music selection systems developed with different structures and architectures [1, 2, 3]. Table 1 presents a comparison of each existing system with respect to the features just presented.

The systems in the literature meet many of the features, but not all of them. Each of the systems has different design premises and therefore meets different design goals.

Table 1: Comparison of music selection services

Requirements	Music Service Systems		
	Jukola	r-MUSIC	MusicFX
1.	Yes	Yes	less
2.	Yes	Yes	Yes
3.	No	No	Yes
4.	Yes	Yes	Yes
5.	No	No	No
6.	No	No	Yes
7.	No	No	No

To fulfill the requirements for the ideal music selection system, it is necessary to look for a supporting set of technologies that facilitate the implementation of these requirements. This led to the choice of mobile agents as the underlying architectural model for the music selection application. The reason for this selection is that mobile agents inherently possess the features of a distributed computational environment and the facilities and utilities to allow simple communications and mobility [5, 6, 7].

The agent research group at the University of Regina, in collaboration with TRLabs Regina – Canada's largest R&D Information and Communication Technology consortium of academia, industry, and government, has developed an agent platform called TEEMA [8]. TEEMA has already demonstrated its validity as an architecture for distributed systems with mobile code (see for example [9]). TEEMA is available both in its compiled form and as source code, which allows complete freedom of experimentation with an agent architecture.

Other technologies complement this architectural choice as shown in Table 2. They are as follows: the Java language and runtime, a java-enabled set of wireless handhelds (for example, cellular telephones), and sealed bid auctions.

Table 2: Supporting technologies

System Characteristics	- Supporting technologies
1. User participation and ubiquity	- Wireless handheld devices
2. Interoperability	- Java technologies - Agent technologies
3. Distributed system and collaboration environment	- Client-server paradigm - Agent technologies
4. Mobility	- Agent technologies
5. Ease of deployment	- Java technologies - Agent technologies
6. Ease of use and flexibility (level of automation)	- Agent technologies
7. Selection model	- Sealed-bid auction

Java is the computer language used to implement TEEMA. Its choice greatly reduces the complexity of mobile code handling, as it comes with a classloader and object streaming capabilities natively. TEEMA merely adopts these capabilities and makes use of them.

Java-enabled wireless hand-held devices are used to participate in the music selection process. These components are not essential to the system, but they are a very useful user interface component: without these devices, the users would have to use some other form of computer access, such as terminals and other wired devices, which would reduce the level of immersiveness for the users. Due to the mobility of TEEMA's agents and the portability of the TEEMA platform, however, the mobile device can easily be simulated on any personal computer.

Sealed-bid auctions are used as an arbitration mechanism for selecting the songs to be put in the playlist. Sealed-bid auctions are easily implemented and therefore represent a mechanism onto which further negotiation strategies can be implemented.

2. SYSTEM DESIGN AND IMPLEMENTATION

The component breakdown diagram of the system design is shown in Figure 1. Agents are enveloped in their supporting platform (TEEMA) which requires Java to work. Other components of the system, such as the web server, may communicate to the agents but are not required to run on the same machines as the agents. All agents may further be distributed in a network and deployed to different computational nodes. There is only one set of mobile agents: the client agents, which interact with the users through wireless devices.

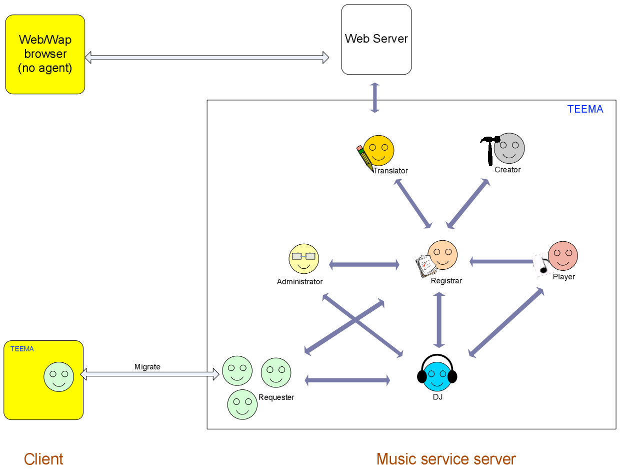

Figure 1: System design

Functional design was started by use cases. Figure 2 depicts the use case diagram for the most frequent type of interaction, the user interaction. The sub-diagram in Figure 2 illustrates what happens during the sealed-bid auction from the point of view of the user. It is important to point out that in this case, the user has the choice of letting the agent bid for the songs the user desires, or a more direct approach can be taken, by making the user interact directly with the auction process, using the agent only as a communication interface.

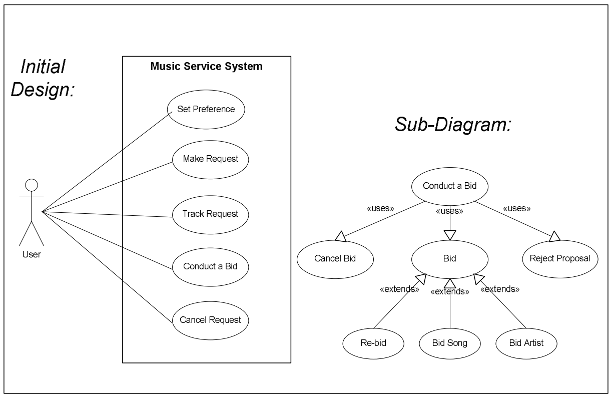

Figure 2: Use case for user interactions

Auctions are created such that a dynamic playlist contain the maximum amount of contribution and to maximize the interaction among agents, and thus ultimately users. The process followed to create the agents is shown in Figure 3.

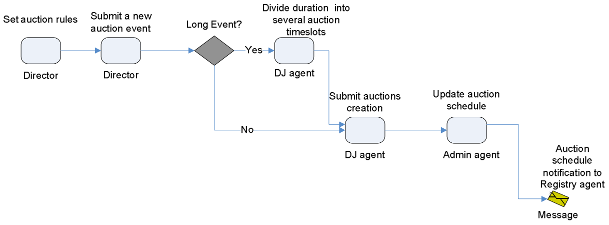

Figure 3: Auction creation process

The behaviour of agents is controlled by state machines. Figure 4 depicts the state machine for the DJ agent.

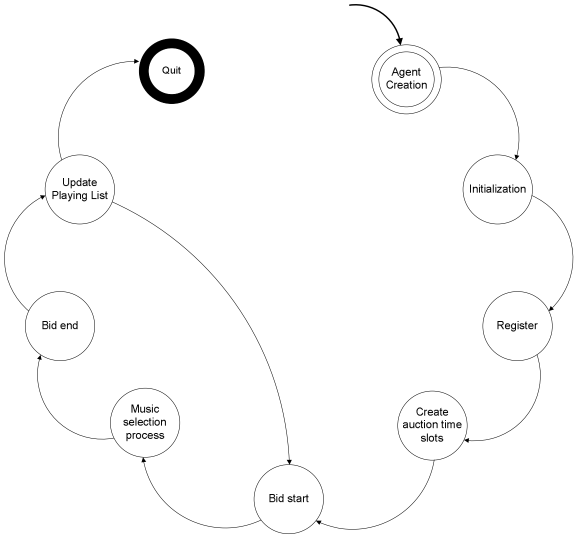

Figure 4: State machine for the DJ agent

As shown in Figure 2, the user has a choice of strategy for competing in the music selection process. The strategies that the user can select are either automatic, in which the agent represents the user at the DJ agent computer and can repeatedly bid on a selection to maximize the chances of that selection, or manual, in which the agent travels back and forth from the user's hand-held to the DJ agent computer. The set of interactions between agents is summarized in Figure 5.

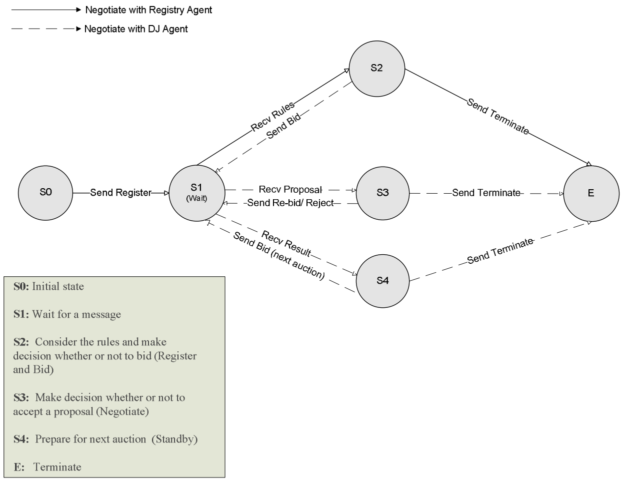

Figure 5: Agent interactions

The implementation of the music selection system is called the Automatic DJ music selection system, or A-DJ. The A-DJ is a test implementation of the architecture and design presented above, and as such it has some minor limitations, mainly in the area of user interface. The system was unit-tested and then validated through the use of scripts. In particular, the maximum number of simultaneous users on a "average" computer (Pentium IV machine at 2GHz with 1 GB of RAM and ATA hard disks, 100MB/s wired network) is approximately 800. This is based on the time required to create agents in TEEMA, and it assumes a 10-minute slot for bidding in each auction, and that the computer contains only TEEMA and the A-DJ system, whereas the web browser used to set the system parameters resides in another computer.

3. RESULTS AND CONCLUSIONS

The A-DJ system was tested in two separate scenarios: one simple selection scenario that assumes no repeated conflicts, defined as conflicts that occur over several auctions; and one complex scenario in which repeated conflicts occur by design.

The results prove that the A-DJ is able to select music respecting the design criteria, and that it meets all the requirements for the music selection system. In particular, the system exhibited the following characteristics:

- Autonomy: the system is capable of working without the need for reconfiguration or administration. The only administrative task required is to set the schedule of a party and select a music theme. The system can then work autonomously.
- Mobility: although A-DJ possesses only one set of mobile agents, the entire system can migrate to another computer or a set of computers without any loss of functionality. This can be used in future work for multi-room parties or for further scalability.
- Scalability: A-DJ is scalable to 800 users on a single machine; the multi-room capability described above can be deployed to additional computers seamlessly.
- Collaboration: agents collaborate and negotiate during the auctions; the resulting playlist belongs to all participants and is accepted as a collaborative effort.
- Platform Independence: due to its Java roots, A-DJ can work on many different platforms seamlessly. So far, it has been tested on Windows ™ machines, Linux machines, and pocket PCs running several different implementations of the Java Virtual Machine.

A summary of advantages and disadvantages of A-DJ is shown below in Table 3.

Table 3: Advantages and disadvantages of A-DJ

Advantages	Disadvantages
Encourages social activities and user participation Entertaining – users have the sense of playing a game competitively All users can suggest their individual songs to the group Consistent music theme Automated – less user effort Support for disconnected mode of operation Scalable Flexible – high level of automation Ease of use Ease of deployment	1. Limited to run on Java-supported devices

A-DJ is a proof of concept implementation, and as such, it suffers from some limitations that will have to be addressed before the system can become a fully implemented product. Specifically, security needs to be addressed, as currently there is only a very simple login process, and reliability agents may need to be deployed to ensure all components are running during the party.

In conclusion, A-DJ showed that a music selection system based on mobile agents has many advantages over conventional systems. Many further refinements to A-DJ are possible, such as multi-room music selection, beat-based music selection, changes of theme and beats per second through party mood sensor agents, and additional security for use in public places.

REFERENCES

[1] O. H. Kenton, L. Matthew, J. Marcel, U. Axel, J. Huw, and M. Peter, "Jukola: democratic music choice in a public space," in *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques*. Cambridge, MA, USA: ACM Press, 2004.

[2] U. Wolz, M. Massimi, and E. Tarn, "r-MUSIC, a collaborative music DJ for ad hoc networks," 2004.

[3] J. F. McCarthy and T. D. Anagnost, "MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts," in *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 98)*, 1998.

[4] M. Rodriguez, P. Arroyo, A. I. Martinez, J. Favela, and C. Navarrol, "Autonomous Agents as Conference Aids in Ubiquitous Collaborative Environments," in *The 2003 International Symposium on Collaborative Technologies and Systems*. Orlando, Florida, USA: The Society for Modeling and Simulation International, 2003.

[5] F. Brazier, O. Marin, and N. Wijngaards, "About Dirstributed Agents," in *IEEE Distributed Systems Online*, 2006.

[6] K. Sycara, "Multi-Agent Systems," in *AI Topics*, vol. 2006: American Association for Artificial Intelligence (AAAI), 2006.

[7] M. Vallee, F. Ramparany, and L. Vercouter, "A Multi-Agent System for Dynamic Service Composition in Ambient Intelligence Environments," in *European R&D Projects*: Philips, 2005.

[8] C. Gibbs, "TEEMA Reference Guide (Version 1.0)," TRLabs, Regina 2000.

[9] K. Smith, R. Paranjape, and L. Benedicenti, "Agent behavior and agent models in unregulated markets," *SIGAPP Appl. Comput. Rev.*, vol. 9, pp. 2-12, 2001.