THE VIRTUAL TWIN: A SOCIALIZATION AGENT FOR PEER-TO-PEER NETWORKS

Alexandre Gachet
University of Fribourg
Faucigny 2
1700 Fribourg
Switzerland

Pius Haettenschwiler
University of Fribourg
Faucigny 2
1700 Fribourg
Switzerland

ABSTRACT

As peer-to-peer computing finally reaches a critical mass, it triggers changes in the IT landscape that traditional network infrastructures based on centralized, client/server topologies cannot manage. Consequently, the ad hoc, self-organized and loosely controled nature of peer-to-peer networks needs to be supported by a new coordination layer representing the interests of the user.

In order to define this new abstraction layer, this paper introduces the concept of the virtual twin—a kind of anthropomorphic representation of the networked person, with whom the user can identify and feel comfortable. We discuss the inner structure of the virtual twin, first in an intuitive and informal way, with an emphasis on its social aspect, then in a more detailed way, with the analysis of its main components.

INTRODUCTION

After many years of theoretical discussions and technical experimentations, peer-to-peer computing finally reached a decent level of acceptance and a critical mass (Wagner 2003). The answer to many "why now?" questions is technology and money, and that is true here. On the one hand, technological advances allowed Internet access providers to bring low-cost, high bandwidth, and constant Internet connections within everyone's reach, through DSL and/or cable subscriptions. Coupled with cheap WiFi appliances and a growing amount of wireless hot spots in public areas such as airports, hotels, parks, squares, coffee shops, fast food (Brewin 2003b; Fleishman 2003), airborne (Disabatino 2003) or on the train (Brewin 2003a), these Internet connections give computer users a new sense of mobility, virtual presence, and location awareness. On the other hand, peer-to-peer collaborative software (such as Groove) or controversial file exchange tools (such as Napster or Kazaa) suddenly brought the possibilities of decentralized computing to the attention of many eager users of the network.

Interestingly enough, the advent of peer-to-peer technologies on a larger scale triggers changes in the IT landscape that were not necessarily foreseen. For example, the well-known characteristics of traditional client/server architectures (such as simplicity, security, centralized authority, clear connection status[1], replication, backup, and load balancing) are gradually being replaced by a set of features that turn the networks into groupings with fuzzy and unpredictable boundaries:

- Groups of users are formed today in an ad hoc fashion (that is to say, informally and on the fly)
- In the new real-time economy, more and more relationships are established between individuals of different organizations, rather than of the same organization. This type of collaboration in which large numbers of geographically dispersed people quickly self-organize in a peer-to-peer network to deal with a problem or opportunity is called *swarming* (Melymuka 2003)
- The centralized control of the almighty system administator is replaced by new authorization mechanisms based on spontaneous invitations or "friend of a friend" standards
- Newer distributed technologies supporting these kinds of groupings are increasingly dynamic, self-forming, self-managed, and self-healing

In many ways, these characteristics free the users from many constraints related to system configuration and management. However, peer-to-peer technologies also weaken the sense of *control* that the users previously had on their networked transactions, as it becomes more difficult to know precisely who is connected with whom, when, how long, and for what exact purpose. As a result, users come with newer questions, such as who manages the knowledge that I put into a network available environment? Who takes care of my personal objectives in the overall community? Who checks that my preferences are respected during interactions? Who supervises my communications with other users? As a matter of fact, the concepts of identity, reputation, reciprocity, cooperation, boundaries, and social networking are growing more and more important to avoid being the target of free riders (Rheingold 2002). In other words, the actors of traditional client/server networks trust the system administrator, who represents the central authority screening the network activity and punishing those who do not stick to the rules. It is crucial to know who or what replaces this role in a distributed, decentralized system.

The emergence of peer-to-peer technologies also impacts the representation of the individuals in the network. The virtual counterpart of a person in a traditional network is usually called

[1] The connection to a server in a client/server architecture is often a deliberate act. In other words, the user is aware that she is passing from a disconnected state (network absence) to a connected state (network presence). However, ad hoc peer-to-peer networks and new pricing schemes based on volume rather than connection time foster the "always on" paradigm, in which the user is not systematically aware of her connection state.

 Copyright © #### by ASME

a *client*, defined by one or several well-defined roles and by precise capabilities managed on the server. The advent of peer-to-peer computing and wireless networking now inspires the vision of "mobile devices [that] will broadcast *clouds of personal data* to invisible monitors all around us as we move from place to place" (Rheingold 2002). The "clouds of data" image gives a good idea of the blurred boundaries of an individual's virtual representation in a peer-to-peer network. The "invisible monitors all around us" vividly express the fact that we are losing awareness of our connections. Even stronger, the person is referred to as a *personal area network*, an interconnected network of devices worn or carried by the user (Zimmermann 1996). Identifying an individual as a personal area network certainly opens new opportunities on the technical level, but is also much more intimidating and less intuitive for the user.

This is where the main problem lies. Newer distributed technologies provide a distributed *infrastructure* allowing system designers to build dynamic and distributed computer systems. However, an infrastructure is not enough to deal with the specifics of a social network and to answer the above mentioned questions. These specifics need to be managed by a different coordination layer. What we need is a new, anthropomorphic representation of the networked person, that the user can identify with and feel comfortable with. To define this new abstraction layer, we introduce a new concept called the *virtual twin*. We believe that this concept can enable and promote the design of human-friendly, secure, dynamic, and social peer-to-peer systems, enhancing the inherent qualities of modern technologies without limiting the freedom of the developers. It can also lead to a new language allowing system designers, architects, programmers and end users to communicate about decentralized models. More formally, it can be the basis of new methodologies able to solve problems and develop systems that satisfy the users' requirements.

The remainder of this paper introduces the inner structure of the virtual twin, first in an intuitive and informal way, with an emphasis on its social aspect, then in a more detailed way, with the analysis of its main components. Many ideas, concepts, and components presented in this paper have been implemented (or are in the process of being implemented) in an open-source project named Dicodess (*http://www.dicodess.org*). In a nutshell, Dicodess is a software framework for developing distributed cooperative decision support systems (DSS). It helps various actors active in a same LAN or WLAN to cooperate during decision making activities. The main purpose of this software framework is to help build DSS for mission-critical decision-making situations happening in highly decentralized environments, where traditional network appliances may be missing or strongly restricted. To reach that goal, the software framework takes advantage of the capabilities of modern computer devices to build ad hoc, peer-to-peer networks without relying on external network infrastructures.

THE VIRTUAL TWIN

Informal Presentation

From a conceptual point of view, a virtual twin can be seen as the *alter ego* of a user, living on the network instead of in the real world. Modern IT systems become natural extensions of the users' capabilities. The virtual twin precisely personifies this extension. The three main components of a virtual twin are *(a)* its working memory, *(b)* its network capabilities, and *(c)* its computing capabilities. Together, these three components build a *working environment*. A careful development taking into account the specific requirements of these components can lead to distributed systems able to build coherent social networks. The components will be described in detail in the coming sections of this paper and are only informally introduced here.

It is easy to draw a simple parallel between a user and its corresponding virtual twin. While the human user manages her knowledge in the part of her brain called memory, her virtual twin manages a formalized version of this knowledge in a part of the network called working memory. Then, while the user employs her cognitive abilities to think, her virtual twin uses the available computing power to process data, to infer new information, and to provide specialized services to other virtual twins. Finally, while the user takes advantage of various verbal and non-verbal communication mechanisms to socially interact with her peers, her virtual twin uses the available network capabilities to interact with other virtual twins. From the user's point of view, the virtual twin hides the specifics of the underlying distributed architecture. In other words, the virtual twin of a user manages the interests of the user on her behalf.

Deciding if the virtual twin should be considered as a simple agent or not is difficult, given the very broad scope of the field and the numerous definitions of what an agent can be. As explained in the coming sections of this paper, we prefer to view virtual twins as a fluctuating population of software components, services, and agents. Traditional agents provide local functionality. Mobile agents are able to move from device to device to provide this local functionality (*insourcing*). Services provide remote functionality (*outsourcing*). As explained later, a typical population of virtual twins includes a few generic agents and services, and many specialized agents and services.

This model fosters cooperation and collaboration between the agents and services provided by a virtual twin. In that sense, the virtual twin becomes a kind of *knowledge factory*. Another parallel can be drawn here: while real-world factories have become natural extensions of the physical capacities of humans, the virtual twin—as a knowledge factory—becomes a natural and networked extension of its owner's brains. Workers in this virtual factory are mostly represented by agents and services.

The Social Effect

To understand why we call the virtual twin a socialization agent, it is important to understand the ins and outs of social software. Traditional project-oriented collaboration tools place people into groups in a top-down way. The new trend of social

Copyright © #### by ASME

software support the desire of individuals to be pulled into groups to achieve goals, in a bottom-up manner. Social software is likely to come to mean the opposite of what groupware and other project- or organization-oriented collaboration tools were intended to be (Boyd 2003). This is in strong contrast with the groupware approach where people are placed into groups defined organizationally or functionally.

Interestingly enough, Boyd (2003) put forth similar arguments about control as per the peer-to-peer networks vs client/server systems described above:

> Traditional groupware puts the group, the organization or the project first, and individuals second. As a member of a Lotus Notes group, for example, you are provided specific access to specific sorts of information based on the administrator's settings. It's all about control. (...) Social software reflects the "juice" that arises from people's personal interactions. It's not about control, it's about co-evolution.

Among the premises of social software, Boyd mentions (1) support for conversational interaction between individuals or groups, (2) support for social feedback (for example, through digital reputation), and (3) support for social networks (to explicitly create and manage a digital expression of people's personal relationships). As we will see in the coming sections, the three main components of the virtual twin help implement this support in systems going beyond simple group forming networks.

The Federalist Model of Cooperation

The concept of the virtual twin is built on top of the *federalist model of cooperation* (Gachet 2003). In this peer-to-peer, human-centered model, each user receives a working environment tailored to her role(s) and skills, and able to adapt continuously to her changing requirements. This working environment contains both the specific knowledge of the user and services provided by the user to other members of the community. Each working environment contains a minimal set of *infrastructure services* needed to run the distributed infrastructure. These services mostly provide basic functionalities such as services lookup, transactions management, inter-processes communication, and distributed storage. If the user has the appropriate rights, she can invite a new user to join the community. A community can only grow by invitation. This simple scheme based on trust is both natural and intuitive.

Gachet and Haettenschwiler (2003b) showed that this federalist model was suitable to create dynamic, self-formed, self-managed, and self-healing[2] communities. However, they did not indicate how the model could be successfully and efficiently implemented. The concept of the virtual twin goes one step further in that direction and provides techniques that end users can use to define their requirements and that developers can use to implement the corresponding IT systems.

[2] the entire user community is self-formed and self-managed at run-time

The next sections describe the three components of the virtual twin and the functionalities they should provide.

THE VIRTUAL TWIN'S COMPONENTS

The Working Memory

The *working memory* is the repository of the virtual twin's knowledge (that is, the knowledge of the user, stored in her twin's working memory). Each virtual twin possesses her own working memory and several virtual twins interacting in a distributed environment can share knowledge by accessing other virtual twins' working memories. In that sense, the architecture is perfectly scalable.

The end users should use this component to describe their data requirements and the privacy and security policy that should be applied to the management of this data. The developers should use this component to implement a distributed data management subsystem[3] that satisfies the users' requirements, as well as technical requirements. As examples of technical requirements, we can mention that the data of a specific virtual twin should be broken down into well defined, independent *knowledge units*. It should be possible to share, reuse, extend, and combine these units. They should be easily represented and managed in the GUI of the system. They should also have a privacy level, such as public, protected, or private.

Value enhancement of the knowledge is a fundamental function of a social network and depends on three criteria: information traceability, information assessment, and peer pricing. First, *information traceability* is very important in a peer-to-peer network. Even if the knowledge units can easily be exchanged and modified, the identity of the various actors involved in the lifecycle of the unit must be retained in its history. Otherwise, there is no incentive for an individual to improve the quality of knowledge units if her contributions are not recognized in the community or, even worse, if they are misused by others. Moreover, the clearly documented history of a unit lifecycle creates a kind of value-added chain at the knowledge level, as it becomes possible to know who changed what in a knowledge unit, when, and for what purpose.

Then, *information assessment* is necessary to appreciate the perceived value of a knowledge unit in a community. The more the actors use and develop existing knowledge units (through their virtual twins), the more the units become rich in contents. The degree of development of a unit lifecycle should be clearly expressed in the system to help users identifying the level of maturity of any given unit.

Finally, *peer pricing* acts as an incentive for individuals to contribute to the development of knowledge units. Active and first-class contributors should be rewarded according to a model similar to supply and demand. The more the contributions of a user are retrieved, the more she is rewarded[4].

[3] Generic distributed data management requirements have been described in Gachet, A. and P. Haettenschwiler (2003a). "A decentralized approach to distributed decision support systems." Journal of Decision Systems **12**(1)..

[4] The exact form of this reward system is an implementation detail.

Copyright © #### by ASME

Rewards can only be granted if the knowlege units have a price. And this price is dictated by the other peers of the community. This kind of return on investment should motivate the contributors to input new knowledge units in the working memory of their virtual twins. Peer pricing is closely linked with peer reputation, a concept that will be described in the next section. Conversely, knowledge units that are outdated, devaluated, neither used nor developed should be gradually removed from the memory to avoid cluttering the knowledge space. After all, any lifecycle ends with the death, or destruction, of its subject. This aspect should also be accounted for by the working memory.

The Network Capabilities

The *network capabilities* group the services needed to bring the distributed, peer-to-peer network up and alive. On the technical level, they group the infrastructure services that need to be implemented by developers to run the distributed system. Examples of infrastructure services include services lookup, transactions management, inter-processes communication, device transparency, and security. On the user level, they group the socialization services needed to turn a basic computer network into a coherent social community. Socialization services should be defined as modules.

A critical module is the *reputation management subsystem*. Reputation systems have been made popular by successful websites such as eBay. In a nutshell, eBay is an Internet auction site allowing sellers and potential buyers to exchange goods through an auction system. As a global marketplace, eBay could face a lot of distrust between buyers and sellers if both categories were completely anonymous. By introducing the possibility for each buyer to evaluate the seller (positively, neutrally, or negatively) and each seller to evaluate the buyer, eBay gives each buyer/seller a reputation, as everyone can see how many buyers/sellers appreciated the behavior of a given individual in the past. In the working environment presented in this paper, reputation is not built by direct peer evaluation, but by the number of contributions in high-priced knowledge units (according to the peer pricing criterion described in the previous section). In other words, the reputation of a virtual twin is based on the quality of its knowledge.

The reputation subsystem should be completed by an *identity management subsystem*. Given that peer-to-peer networks are built on the fly, without central repositories able to store and retrieve identity information about all the connected users, each virtual twin should be able to broadcast personal data about the corresponding individual. This data, combined with reputation information, shapes the personality of the virtual twin.

Yet another important socialization module should take the form of a collaboration framework fostering reciprocity, awareness and the preservation of the user's own interests. Unlike the identity management subsystem, which returns generic information about a virtual twin, *awareness functions* propagate real-time information and answer questions like: is this user present in the network right now? Is she available, is she active, is she cooperating with other users, is she in a good mood, etc.? Once again, such contextual information enriches the personality of the virtual twin.

The preservation of the user's own interests is also critical in a social network. Even if a social network relies on trust and confidence between its members, a realistic framework needs to account for sources of distrust and conflicts (Gans, Jarke et al. 2001). The fragile equilibrium between trust and distrust can only be achieved if the users feel that their own objectives and interests are respected by the supporting network infrastructure—only then will the users feel comfortable in the network. This last functionality gives each virtual twin its uniqueness in the social community.

The Computing Capabilities

Computing capabilities represent extended services provided by individual virtual twins. The computing capabilities describe the ability of a virtual twin to accomplish high level tasks for others, by exploiting in a transparent way local functionality, web services, agent services and/or the capabilities of other virtual twins. For example, a role responsible for data management could offer a specific data manipulation service or the role responsible for reporting could offer an extended reporting service. Other examples of specialized services provided by the computing capabilities of a virtual twin include cartography services, directory lookups, news feeds, etc. In that sense, each virtual twin can be the client *and* the server of other virtual twins. This is an important departure from traditional peer-to-peer systems, such as file exchange tools or desktop collaboration software (e.g. Groove.net), that mostly exchange passive data, but no active services (called *behavior* in the virtual twin terminology). In such systems, data exchanges are distributed, but processing is mostly executed locally. The ability given to the virtual twins to share behavior is a much closer model of human interactions and can lead to richer social networks. By analogy, a virtual twin does not ask other virtual twins if it can *give* them something (data exchange), but if it can *do* something for them. This is a kind of outsourcing model for peer-to-peer networks. The computing capabilities of the virtual twin form a distributed application layer on top of the network capabilities.

CONCLUSION

This paper introduced the concept of the virtual twin, a kind of anthropomorphic representation of the networked person, made of three main components: *(a)* a working memory, *(b)* network capabilities, and *(c)* computing capabilities. We discussed how this socialization agent can turn primitive distributed infrastructures into social groups promoting features such as knowledge value enhancement, information traceability and assessment, peer pricing, trust, reputation, identity management, and networked behavior.

From a technical perspective, a virtual twin can be seen as a personal operating system managing the interests of a peer through many services, on many devices, whereas a traditional operating system covers the interests of a single device. The virtual twin can hide several forms of distributed processing, such as client/server, master/slave, parallel processing, or agents subcontracting. It coordinates both the interactions

Copyright © #### by ASME

between a user and her virtual twin, and those between the twin itself and other virtual twins.

Many ideas, concepts, and components presented in this paper have been implemented (or are in the process of being implemented) in a project helping various actors active in the same LAN or WLAN to cooperate during decision making activities. This is an open source project in which the scientific community is welcome to participate (http://www.dico-dess.org).

REFERENCES

Boyd, S. (2003). Are You Ready for Social Software? Darwin.

Brewin, B. (2003a). All aboard with Wi-Fi on VIA Rail Canada, *Computerworld*. Retrieved July 11, 2003, from http://www.computerworld.com/mobiletopics/mobile/story/0,10801,82940,00.html.

Brewin, B. (2003b). McDonald's signs on for Wi-Fi in San Francisco area, Singapore, *Computerworld*. Retrieved July 9, 2003, from http://www.computerworld.com/mobiletopics/mobile/story/0,10801,82839,00.html.

Disabatino, J. (2003). AT&T to provide satellites for Connexion by Boeing, *Computerworld*. Retrieved July 14, 2003, from http://www.computerworld.com/mobiletopics/mobile/technology/story/0,10801,72106,00.html.

Fleishman, G. (2003). Hot Spots Start to Get Real, *O'Reilly Network*. Retrieved July 1st, 2003, from http://www.oreillynet.com/pub/a/wireless/2003/03/06/hot_spots.html.

Gachet, A. (2003). A software framework for developing distributed cooperative decision support systems. *Inaugural dissertation*. UniPrint, University of Fribourg, Fribourg.

Gachet, A. and P. Haettenschwiler (2003a). "A decentralized approach to distributed decision support systems." Journal of Decision Systems **12**(1).

Gachet, A. and P. Haettenschwiler (2003b). Distributed decision support systems: a federalist model of cooperation. in R. Bisdorff (eds) Human Centered Processes - Distributed Decision Making and Man-Machine Cooperation, proceedings of the 14th MINI EURO Conference, Luxembourg: 211-216.

Gans, G., M. Jarke, S. Kethers, G. Lakemeyer, L. Ellrich, C. Funken and M. Meister (2001). Requirements Modeling for Organization Networks: A (Dis-)Trust-Based Approach. in (eds), IEEE Computer Society Press, proceedings of the RE-01 Conference, Toronto, Canada.

Melymuka, K. (2003). Meeting of the Minds: Technology for business "swarming", *Computerworld*. Retrieved August 1, 2003, from http://www.computerworld.com/softwaretopics/software/groupware/story/0,10801,83401,00.html?SKC=groupware-83401.

Rheingold, H. (2002). Smart mobs : the next social revolution. Cambridge, MA, Perseus Pub.

Wagner, M. (2003). From Peer to Maturity, *Network World*. Retrieved January 13, 2003, from http://www.nwfusion.com/research/2003/0106peer.html.

Zimmermann, T. G. (1996). "Personal Area Network: Near-Field Intrabody Communication"." IBM Systems Journal **35**(3&4).

Copyright © #### by ASME