Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I,
IMECS 2010, March 17 - 19, 2010, Hong Kong

Human-centric Mechanism Design for Ubiquitous Systems and Applications

Simon G. M. Koo, Member IAENG *

Abstract—**With the proliferation of communication technologies and mobile electronic devices, there is a fast growing need of ubiquitous applications. However, designing ubiquitous applications faces a lot of challenges, and many of them actually comes from *human* rather than from computers. In order to handle the challenges, system designers must provide a *human-centric* protocol that specifies the rules of the system. The process of designing such protocols is not an easy task, especially in a distributed environment where there are a large variety of devices with different capabilities, or when users have disparate interests on their own. The goal of the paper is to present the need to develop a theory for providing fundamental methodologies in designing *human-centric* ubiquitous systems and applications.**

Keywords: human-centric, mechanism design, ubiquitous systems, peer-to-peer, distributed systems

1 Introduction

In traditional computing environments, users actively choose to interact with computers. During the last decades, however, rapid developments of wireless networking technologies were witnessed, and many forms of *ubiquitous computing* are becoming core parts of our daily lives. Ubiquitous computing and applications are different from their conventional counterparts – they will be embedded in the users' physical environments, taking users' personal preferences into the computing process, and integrated seamlessly with their everyday tasks. With the proliferation of communication technologies and mobile electronic devices, there is a fast growing need of ubiquitous applications.

In his seminal article [1], Mark Weiser presented his vision on ubiquitous computing, which led to a set of defining characteristics and requirements, and most importantly, research challenges for ubiquitous applications. Surprisingly, many aspects of Weiser's vision are still very futuristic today as they did in 1991, and the reason for such lack of progress is primarily a lack of integration in existing systems [2]. On the technical front, high-performance infrastructures and devices such as 3G mobile phones and networks, satellite navigation systems, light-weight handheld and wearable devices, wireless sensor networks, etc., are all readily available or could be deployed relatively trivially. What is lacking are the applications that bring the technologies together – applications that can harness the power of these pervasive technologies.

Designing ubiquitous applications faces a lot of challenges, and many of them actually comes from *human* rather than from computers. For example, the system designer may have an objective that is different from that of some users, and it is difficult to find a compromise even though both objectives may be technically feasible. Moreover, in a large-scale, distributed environment, even the users may not have the same objectives among themselves, as they have different preferences and operate under different settings, and it is hard for the design to take everything into account. To further complicate the matter, users preferences may also change from time to time according to the current environment and the state of the system. In order to handle the above situations, system designer must specify a *protocol* that specifies the rules of the system. For instance, a protocol may specify how the network resources will be allocated, or what information a user should reveal to the system, or how do users communicate among themselves. The process of designing such protocol, however, is not an easy task, especially in a distributed environment where there are a large variety of devices with different capabilities, or when users have disparate interests on their own.

In the field of artificial intelligence and machine learning, agent-based approaches, which emphasize autonomous actions and flexible interactions, suggested computational models for the above scenario [3]. In designing such multi-agent systems, two fundamental issues must be addressed. First, a protocol must still be specified to govern the interactions. These protocols, like network protocols, cover issues such as how the agents' actions translate into an outcome, the range of actions available to the participants, and whether the interactions occur over steps or are one-shot. Second, given the prevailing protocol, each agent's strategy must be defined. Sometimes, a designer might be able to impose each agent's protocol and strategy. In such settings, the agents can

*Department of Mathematics and Computer Science, University of San Diego, San Diego, CA 92110. Email: koo@sandiego.edu.
This work is supported in part by the Faculty Research Grant and the International Opportunity Grant, College of Arts and Sciences, University of San Diego.

cooperate to find a good system-wide solution. It should be noted that, however, this is usually not feasible because the agents represent distinct user requirements with potentially conflicting goals that seek to maximize their own gains.

Consequently, the best a designer can achieve is a non-cooperative strategic analysis, in which the designer can impose only the protocol and can't control which strategies the agents adopt. *Mechanism design* is a subfield in economics that looks at economic systems from the viewpoints of a system designer, who attempts to implement an aggregation of preferences from different participants in a society towards a single joint decision, e.g. allocation of public resources, that maximizes *social welfare*. A *mechanism* is a specification of how economic decisions are determined as a function of the information that is known by the individuals in the economy, which reminiscences protocols in networking and communication systems. Mechanism design provides an elegant mathematical framework in which to study protocols that give the agents incentive to act and interact in particular ways and that also have useful computational properties. Hence, we argue that mechanism design is a better model to study for potential adoption in the design of ubiquitous applications than multi-agent systems.

2 Incentive Mechanism Design in Distributed Systems

The foundation of most distributed systems, particularly peer-to-peer (P2P) systems, is based on the contribution of resources from peers and such contribution plays a significant role in the efficiency and performance of the system. A lot of efforts have been made to avoid freeriding and *tragedy of the commons* [4], two of the major problems in P2P networks. In a distributed environment, where users, represented by *automated agents*, may have disparate interests on their own, a mechanism, or *protocol*, for negotiation and compromise on how to share resources and maintaining fairness will be a critical component for the overall success of the system. Here we use the term protocol *exclusively* for the "negotiation and compromise" mechanisms adopted by a P2P system, which is distinctive from a network protocol, such as TCP/IP, or an application-layer protocol for communications between processes. In the rest of the paper, we shall use the term protocol and mechanism interchangeably.

We shall first define a few terminologies before the discussion continues. *Agents* are generally considered to be electronic entities representing there (human) clients in the process of negotiation, and they will act on behalf of their clients and bind to the protocol given.

A *protocol* specifies the rules on what deals agents can make and what information one should reveal in order for others to make decisions. Once the protocol is laid out, it will be up to the agents to choose an appropriate *strategy* and interact with other agents. It should be noted that agents merely execute the negotiation – it is the clients who design and propose the strategies for the agents. If the client doesn't propose one, its agents will follow a *default* strategy proposed by the protocol.

As the goal of each P2P system differs, they will need a different protocol to achieve them. *Mechanism design*, a term first used in economics, attempts to implement mechanisms to maximize social welfare, or as defined in [5], an attempt to implement "desired social choice in a strategic setting." As in game theory, mechanism design generally assumes that agents (and clients) will act rationally and are trying to maximize their own utilities [6].

Mechanism design has to deal with every configuration of agents' utility functions. For tractability, we can assume there is a known set of all possible utility functions, and each agent will be assigned a type based on which functions it is currently using. Knowing the type of other agents can improve one's decision process which in turns provide more possibilities for one to reach a better choice. However, it may not be beneficial for an agent to reveal its own type, or worse, an agent may be better off if it lies about its type. A mechanism is considered to be *incentive compatible* if agents have incentive to reveal their types. In other words, the best strategy for any agents in an incentive compatible system is to reveal their true identity and follow the protocol.

There has been various attempts to design a P2P system that is incentive compatible [7, 8, 9, 10, 11]. Among them, BitTorrent [12] is one of the most popular P2P-based file distribution systems that is incentive-compatible. It uses a "tit-for-tat" mechanism to reward peers that contribute more of their capacity in assisting the distribution process. However, these protocols either suffer from a high cost of reputation management, or a course granularity in classifying client types. In [13] and [14], we have proposed an incentive compatible mechanism for neighbor-selection that does not require maintenance of reputation, and a "seeing-is-believing" resource sharing mechanism where the best strategy for clients is to comply with the protocol and contribute as much as they can.

3 Human-centric Mechanism Design

It is until recently that stocks and other derivatives trading have been deploying *automated trading agents* to represent the clients. These agents are programmed with client-defined strategies, and react to the market based on their prediction or other modeling outcome. The upside of these agents include fast responses (to a short arbitrage opportunity, for example) and do not make emotional decisions. An agent will follow its strategy regardless of how many times it has failed in the past, but the same thing

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

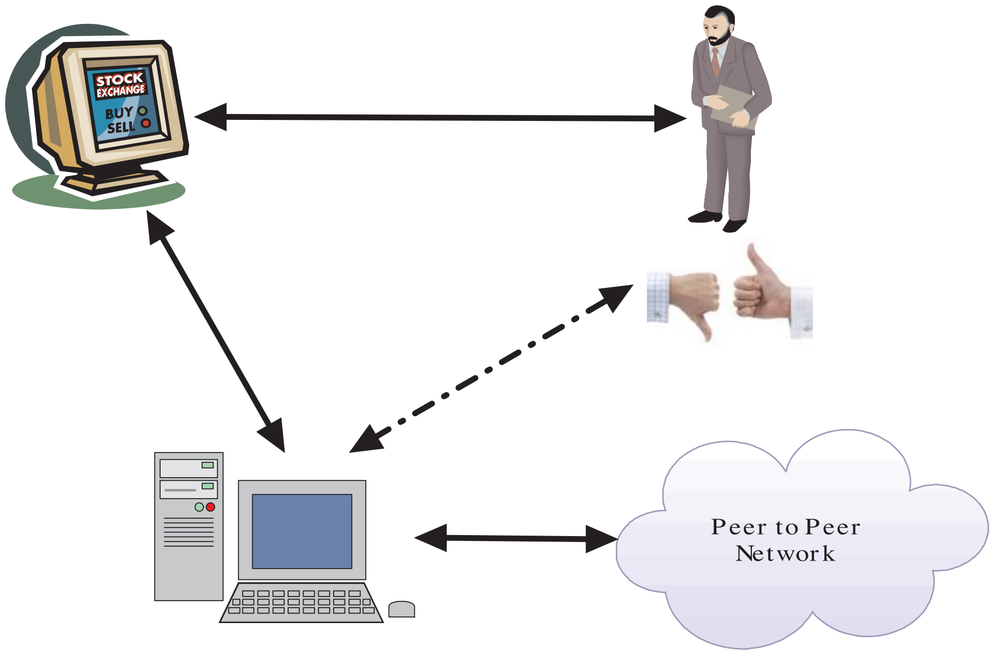

Figure 1: "Grey-box" decision with human intervention.

may not be said about human traders.

Even though trading agents have these advantages, they are also incapable to make adjustments[1] or take into account other features that are not described in their model. For large investment banks and hedge funds, trading without the supervision of human being is very risky, so many automated trading systems deploys a *grey-box mechanism* (Fig. 1) [15]. As opposed to a completely automated *black-box mechanism*, where the agent makes all the decision and the process is virtually transparent, the client (the trader in this scenario) is given mechanisms to veto a trade or propose a new one if deemed necessary. Such actions, if not taken into account, may harm the social welfare of the system, as strategies are now not simply derived by agent, but also with human intervention.

With the above innovation, we present our theoretical framework for *human-centric mechanism design* (HMD). One major difference between HMD and conventional mechanism design is that human are generally less rational than machines and contains emotions, which violates the rationality assumption in the first place. In the case of automated trading, for example, human also have a tendency to either settle too soon or hold on too long. Nonetheless, machine makes decisions (and take corresponding actions) in milliseconds, which is something infeasible for human to do.

It is very difficult, if not impossible, to model a hybrid of human behavior and machine behavior using a single, unified model. The bright side is, in a grey-box scenario, most of the decisions are still made by the agents, and human interventions appears only on critical situations. Our design has taken into account serval of those situations, and incorporated them into the overall mechanism design process. Design considerations such as computational cost, stability, and symmetry are particularly important in our framework, and each challenge has been properly addressed.

4 Conclusions

We have presented the need and our position on human-centric mechanism design for ubiquitous systems. In the design, we identified situations where human intervention impact social welfare, and incorporated the design framework for peer-to-peer grey-box systems.

References

[1] M. Weiser, "The computer of the 21st century," *Scientific American*, vol. 265, no. 3, Sep. 1991.

[2] N. Davies and H.-W. Gellersen, "Beyond prototypes: Challenges in deploying ubiquitous systems," *Pervasive Computing*, pp. 26–35, Jan–Mar 2002.

[3] N. Jennings, "An agent-based approach for building complex software systems," *Communications of ACM*, vol. 44, no. 4, pp. 35–41, Apr. 2001.

[4] G. Hardin, "The tradegy of the commons," *Science*, vol. 162, pp. 1243–1248, 1968.

[1] We consider adaptive strategy a fixed strategy in the sense that the updating and learning algorithm is fixed.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[5] N. Nisan and A. Ronen, "Algorithmic mechanism design," *Games and Economic Behavior*, vol. 35, no. 1–2, pp. 166–196, Apr. 2001.

[6] C. A. Holt, *Markets, Games, and Strategic Behavior*. Addison Wesley, 2007.

[7] R. Jurca and B. Faltings, "An incentive compatible reputation mechanism," in *Proc. of the IEEE Conference on e-Commerce*, Newport Beach, CA, Jun. 2003, pp. 285–292.

[8] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for cooperation in peer-to-peer networks," in *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, Jun. 2003.

[9] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, "An incentive mechanism for P2P networks," in *Proc. of the 24^{th} Intl. Conf. on Distributed Computing Systems (ICDCS '04)*, Tokyo, Japan, Mar. 2004, pp. 516–523.

[10] T. Moreton and A. Twigg, "Trading in trust, tokens, and stamps," in *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, Jun. 2003.

[11] Y.-W. Sung, M. Bishop, and S. Rao, "Enabling contribution awareness in an overlay broadcasting system," in *Proceedings of ACM Sigcomm*, Pisa, Italy, Sep. 2006.

[12] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, Jun. 2003. [Online]. Available: http://bitconjurer.org/BitTorrent/

[13] S. G. M. Koo, K. Kannan, and C. S. G. Lee, "On neighbor-selection strategy in peer-to-peer networks," *Future Generation Computer Systems*, vol. 22, no. 7, pp. 732–741, Aug. 2006.

[14] S. G. M. Koo and C. S. G. Lee, "An incentive-compatible mechanism for efficient distribution of bulk contents on peer-to-peer networks," *Telecommunication Systems*, vol. 34, pp. 81–90, 2007.

[15] S. G. M. Koo, "Grey-box mechanism design in peer-to-peer cooperative networking," in *Proceedings of the first IEEE International Conference on Ubiquitous and Future Networks (IEEE ICUFN)*, Hong Kong, China, Jun. 2009.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)