PARTICIPATORY SYSTEMS – A PARADIGM SHIFT TO ANSWER THE CHALLENGES OF AN INTEGRATED WORLD

M.Sc. Jánoky L. V.[1] PhD. Ekler P.[1]
Department of Automation and Applied Informatics – Budapest University of Technology and Economics, Hungary [1]

laszlo.janoky@aut.bme.hu, peter.ekler@aut.bme.hu

Abstract: *The increasing integration of technology and everyday life creates a new set of challenges in multiple fields. The fully connected world, in which virtual and physical layers interact seamlessly is a goal envisioned and pursued by many, yet there are some major obstacles in the way of achieving it. One of the fundamental differences between previous systems and the envisioned solution, from information technology point of view, is the diffusion of the roles of service providers and consumers. Traditional solutions, with fixed roles and abstracted physical layers can be too rigid to address the arising challenges. In order for these systems to work, they have to be vastly more scalable and account the physical layer too in the system architecture. In this paper, we investigate and show some examples and notable works on a possible solution to these problems, the field of participatory systems. In a participatory system, the roles of client and service providers are diffused, clients are working together as an integral part of the system, pooling their resources and providing services among themselves.*

Keywords: PARTICIPATORY SYSTENS, VOLUNTEER COMPUTING, SOFTWARE SYSTEM ARCHITECTURE, IoT

1. Introduction

The concept of participatory systems has a long history in computing. The considerable amount of unused computational resources, residing in the hardware of the client has inspired many to search for ways to utilize it. One of the natural consequences of this idea led to the emergence of volunteer computing systems, like BOINC [1]. These systems used the computational resources of their clients, like ordinary desktop PCs to pursue specific goals, primarily scientific ones. Uses of these systems included tasks like protein folding, radio signal analysis and the search for Mersenne primes. These systems are usually unique, dedicated to a single problem and have a limited lifetime, bound to the solving of that specific task. Although they serve different purposes, they often have a common technological basis, for example the aforementioned BOINC platform provides the technological background for several different projects.

Another approach to utilize the resources of clients is to use participatory methods to augment the capabilities of a 'live' system. This approach could be taken even further, clients could become the sole providers of a service, which would offer significant advantages. Even with the limited approach, the inclusion of clients actively participating in the system would achieve a certain amount of self-scaling. The more clients joining and participating, the greater the capabilities of the system becomes. An architecture designed according to these principles would solve many of the greatest technological challenges in a fully connected world, for example the sheer scale of these systems.

The goal of this paper is twofold, in the first part we provide an overview about the state of the field of participatory systems by showing a possible classification method based on our previous work [2], providing examples for each class. Then, in the second part, we focus on open problems in the world of IoT and integrated solutions, where participatory methods could prove to be advantageous. Finally, we outline some research directions for participatory systems, worth pursuing to achieve these goals.

The structure of this paper is dictated by these two goals, after this introductory section, we delve into the overview and classifications of participatory systems in the second section. Here we show the classification methods and examples for each class. In the third section, we discuss the open problems and use-cases where participatory methods could be of use. Finally, we wrap the discussion in the last part, where we reiterate the topics covered and outline some future directions.

2. Classification and examples of participatory systems

Definition: A Participatory System is a system which provides services to its clients with active participation on their part. In a system like this, clients are working as an integral part of the whole providing services themselves, among themselves.

Strictly to the definition a great deal of typical information systems could be classified as participatory, in essence one could say that the quality of the *activity* requirement from clients is the determining factor. In order to narrow the number of possible candidates we establish some classification parameters. For each of these parameters, we show some examples to demonstrate how different solutions behave.

The first consideration for classification is from the perspective of *clients*, the second we investigate is from the direction of the *goals* followed by the system. Finally, we will look at the systems from a *technological* a technical standpoint.

2.1 Clients

When looking at the classification problem from the perspective of *clients*, we intend to determine the effects of the characteristics of participating client on the system. We could ask questions like; Are participating clients a mandatory part for the operation of the system or they just enhance the system capabilities? Are there multiple roles of clients, or every client can be considered the same? When referring to client participation, do we mean the client machine or the actual person sitting behind the desk? In the following, these questions are investigated.

2.1.1 Client Driven and Client Enhanced systems

A participatory system can be either *client driven* or *client enhanced*. A *client driven* system requires the active participation of its clients to provide service, a *client enhanced* system can serve non-participating clients in itself, while adding new clients improves the system's capabilities.

A good example for a *client driven* system would be a completely distributed peer-to-peer file sharing system. As a concrete example one could look at the tracker-less version of the BitTorrent protocol. In this example, every client is equal in the system, their goals are the same and without their participation the system would be purposeless.

For a *client enhanced* system, we could take the example of a peer-to-peer streaming solution, where clients share the data among themselves [3]. In these systems, usually there is a primary source of content (e.g. a centralized server) and clients who want to access this data. By having the clients share the data they receive, the load on the centralized server can be reduced significantly. This approach has already been adopted by the industry in the form of peer assisted content delivery networks, for example Peermesh [4], Peer5 [5] or Swarmify [6].

2.1.2 Homogenous and Heterogeneous client roles

Another classification point would be the characteristics of client roles, there are *homogenous* and *heterogeneous* systems when looking at these roles.

In the first case, clients are a homogeneous group from the perspective of their roles filled in the system. Each client interacts with the system the same way, consumes its services and actively participates in providing them to others.

In case of a *heterogeneous* system, there are more than one group of clients. E.g. one group may participate in service provision, while others could only consume the said service.

Volunteer computing systems, for example the previously described BOINC bases solutions, like SETI@Home [7] or Folding@Home [8] are heterogeneous systems. In these cases, there are two groups of clients. One of them is the group of participating clients, who willingly dedicate their computational resources to achieve a greater computational capability for the system. The other group is the group of end users, who provide the tasks for clients to solve and use the results for their own purposes.

For a homogeneous system example, we could look at the previously described file sharing system. In a system like that, every client has similar purpose and each of them participates to achieve it.

2.1.3 Interactive and Non-interactive systems

An important consideration for classification is whether the actual human client must participate or not. This interactivity is usually determined by the main purpose of the system, e.g. if the service is request oriented, it often includes interaction.

When the participation requires active human interaction, we talk about an *interactive system*. This interaction could mean a simple acknowledgement or could be a more challenging task. Often the purpose of these systems is to solve human intelligence problems, which usually have no known perfect algorithmic solution. Examples usually involves exploiting the pattern recognition strength of the human brain and could include tasks like media categorization and tagging, visual processing of satellite imagery [9] or text digitization.

On the other hand, a *non-interactive* system does not require the active participation of the human client to work, neither it requires their knowledge or their consent. Usually these systems work in the background to improve the service level of a system, like the previously mentioned peer-to-peer content delivery platforms, but there are also some less reputable real-world examples.

Strictly to the definition, one could say that bot-nets and cryptojackers [10] (browser based crypto currency miners hidden from the user) are some real-world examples for non-interactive systems.

2.2 Goals

Another possible point of examination are the goals of the system; they play a key role in determining the characteristics.

The key questions one could ask regarding the goals could be stated as the following; Is the end goal of the system a one-time event, or is it an ongoing purpose, like providing a service? Does the system rely on purposefully participating clients, or just opportunistically uses their resources to further its goals?

2.2.1 One-off and Continuous services

One could also examine the end goal of the system, whether it's a one-time task that can be completed, or it's a continuous task, integral to the system.

In the first case, if the goal can be fulfilled, we talk about a *one-off* system. In this case, participants are solving parts of the task, or different tasks possibly leading to the desired solution. If this solution is reached, the system's life-cycle ends.

A great example for a *one-off* solution would be the BOINC based volunteer computing system set to solve the minimum sudoku problem [11]. In this case, the question was the following: what is the smallest number of clues (filled numbers) that a Sudoku puzzle can have? The answer that was proved by the system was 17.

The other option, continuous services are more common, both the previously discussed peer-to-peer content delivery systems and the radio signal analysis done by SETI@Home are ongoing, continuous services, aimed at different clients.

2.2.2 Opportunity driven and Purpose driven systems

From the perspective of initiative, one could distinguish *opportunity driven* and *purpose driven* systems, the main difference being the context in which they use their client's resources.

Opportunity driven systems profit from the participation of its clients whose original goal is the consumption of the provided service. On the other hand, *purpose driven systems* assume its clients are willingly participating to reach the end goal itself.

Volunteer systems, by their nature are always *purpose driven*, this behaviour is the more common of the two classes. Examples in this case would include grid computing and CPU cycle scavenging systems, like project Bayanihan [12].

As an *opportunity driven* system a good example would be the case of reCAPTCHA project [13]. In this system, the clients are consuming a primary service, the security gateway intended to protect a resource from autonomous requests, while in the meantime the participants help with the digitization of written text or images.

It's also feasible to employ primarily purpose driven use-cases, like the running of massively parallel computations in an opportunity driven fashion, which would improve the proliferation of such systems, as described in our previous work [14].

2.3 Technology

From a technological standpoint, we consider the resources contributed by the clients to the system as the prime point of discussion. Client technological characteristics have a determining effect on both system architecture and capabilities. In essence the question on could ask here would sound like this; what does the client contribute?

2.3.1 Computational systems

In a *computational system* clients are pooling their resources to create a larger, more potent system in terms of computational power. These systems are best employed when working on embarrassingly parallel problems, e.g. running simulations, solving cryptographic problems, or running certain machine learning algorithms.

It is a proven fact by previous works in the field - some discussed earlier, that the capabilities of these solutions are significant. In 2007, the computational capacity of the Folding@Home project superseded the world's largest supercomputer's performance [15] at the time, and continued for four years.

In most of these computational systems, the original problem is partitioned to smaller tasks, which are assigned to the clients and executed in parallel. When finished, the client uploads the result to a central server which aggregates them.

For reliable results, this mandates the presence of additional security measures, such as redundant calculations to avoid malicious clients providing false solutions. As a consequence, the central component's complexity and required capacity could be impacted negatively.

2.3.2 Distribution systems

Distribution systems focus on providing data as a service to all of its clients. In a participatory way, this is usually achieved with direct client to client (peer-to-peer) connections. Usually in this case, each client can request data either from the central server or other clients. In theory, the more clients the system has, the greater the ratio of clients distributing the data among themselves, hence less load on the central component.

An example for a participatory *distribution system* would be the previously mentioned peer-to-peer streaming solution. In this case the goal of the system is to provide each client with an acceptable level of service (i.e. enough data in a timely manner), while minimizing the impact of growing client traffic on the central component.

A more interesting class of distribution systems are information sharing systems, where the information is bound to the clients themselves. This may either be caused by the information originating from the clients themselves, for example if it's based on a geographical location. An information distribution system's aim is to provide the relevant information to the interested parties (i.e. clients interested in a geographical area receive information from other clients located there). Good real-world examples here would include community driven navigation software, such as Waze [16].

2.3.3 Sensing systems

A sensing system is another special kind of participatory system, where clients are building up larger sensor networks, usually aimed at collecting data from a geographical area.

The field of participatory sensing [17] or crowdsensing [18] deals with these kinds of systems and they are already being adopted by the industry.

While by strict definition we consider all sensing systems participatory, we should make a distinction between sensory networks made up of relatively intelligent, participating clients, such as smartphones and simple distributed sensor networks.

3. Potential uses of participatory methods

The basic concept of a system and its interaction with the users in integrated world is close to what we described as a participatory system. In this case integration means the closing of the gap between physical and virtual layers, making the system boundaries less clear, involving the clients in the actual process is a goal.

Participatory methods are becoming more important in this environment, as this level of integration provides new challenges. Methodologies used in purely virtual or lightly integrated systems can become insufficient at this scale.

In this section, we identify some key aspects and scenarios, where participatory methods offer potential advantages over previously employed practices.

One of the inspirations for grouping the problems in this section was the work of John A. Stankovic's on "Research Directions for the Internet of Things" [19].

3.1 Scaling

Maybe the largest problem with the concept of a fully interconnected world is its scale. The traditional client-server architectures used currently in most information systems are simply unable to scale beyond a point. An alarming indicator of how insufficient the current infrastructure is, is the 2016 Dyn DDoS attack [20], which was the largest of its kind at the time of writing. The relevant aspect of this attack is that it was performed using a botnet made from mostly IoT devices [21]. Experts predict the use of these devices to rise exponentially in the near future, it's easy to imagine the strain they would impose on current infrastructure.

Answering to the scalability problem with a general solution is not possible. Each individual use-case and system requirement influences the overall system characteristics, sometimes drastically. What we can do however is come up with some guidelines and key considerations on how to improve scalability, many of them includes the usage of participatory methods.

As opposed to previous large-scale systems, when thinking world scale, it is not enough to provide the large horizontal scaling capability of a system. It's also imperative to be able to provide the actual computational capacity. While its certainly possible physically to simply build more datacentres, it may be economically more feasible to look for other alternatives as the number of clients drastically increase.

Participatory systems could help a great deal in this regard, by utilizing clients to enhance system capabilities, these systems could become self-scaling. By using the clients as service providers themselves, an increase in their number would also mean a larger capacity in the system for serving the said clients. This approach naturally has some limiting factors regarding where it can be used, but the general characteristics of these systems make them a suitable candidate for such solutions.

Novel approaches would be required for some cases where using traditional methods on a smaller scale would be trivial. This task is not insurmountable however, as it was already demonstrated several times throughout history. As recent example, one could look at is the millennium old concept of currency. Currency was re-imagined and recreated in a distributed, yet secure way with the emergence of cryptocurrencies like BitCoin [22]. The task of account keeping and monetary transaction management is one of the source of the original motivation behind the theory of centralized information systems with proper safeguards for consistency and security. Both of these concepts are usually considered the antithesis of peer-to-peer and distributed systems, yet a novel approach with proper theoretical background and innovative implementation could solve it in a satisfactory way.

3.2 Architecture and environment

Heavily related to the scaling, a key difference between the new integrated systems and their predecessors would be the architecture they employ. Previously architecture could be easily separated to virtual and physical components, and each could be evaluated separately. In an integrated system, the components are much harder to separate by this measure.

A key difference however, where participatory methods also come into play is the emergence of a new separating factor; geography. By integrating the virtual and physical world its largely unavoidable to introduce this new factor to the system. It's necessary for these systems to account for the geographical locations of their clients, as it became a primary client attribute with the close integration.

The geographical aspect however has some fortunate consequences in solving the problems. Similarly to the concept of light cone in general relativity, one could determine the possible dimensions in space and time which would affect the given client when serving a request. This could be used to break up the larger system into smaller chunks, based on client geo-location.

Combining this with the usage of participatory methods, one could turn a centralized, world scale system into a collection of collaborating geographically close clients. For example, a public transportation navigation system could use information shared directly between clients travelling at different locations, without the need for maintaining a centralized service for providing it.

One of the consequences of this approach is that the system would have to be scaled according to the client density instead of the total client number.

Ultimately the base concept of these systems could be summed up as the following; Instead of collecting information to a central place and serving it, each information source serves it themselves. Clients in these systems turn to information's source, what is not a centralized role in this case. As information is usually bound to the geographical location this makes a natural grouping of clients.

4. Overview and future directions

In this section, we outline some possible directions worth pursuing in the field of participatory systems and wrap the conversation with an overview of the paper.

4.1 Research directions

One interesting directions to examine would include the modelling of information flow based on geographical locations and its effect on the system. Its worth investigating how clients could discover each other and get the relevant, geo-bound data from the most relevant source.

Another interesting direction would be the investigation of the effects of different client behaviours and characteristics on different participatory systems and possibly set QoS metrics.

4.2 Overview

In the paper we introduced the term of participatory systems, described the brief history of the field and presented some of the more significant related works. We have shown a classification method for these systems, providing examples and possible use-cases for each examined class. We have identified three main classification factors: roles of the participating clients, characteristics of the goals of the system, and technological archetypes.

After that we have shown two major areas in the concept of the fully integrated world where participatory methods could be advantageous; scaling and geographically aware architectures. We discussed the possibility in employing participatory methods to build unprecedented scale systems and compared it to traditional solutions.

We have also examined the effect of client geographical considerations in case of the system architecture. We determined that when the information source is geography based (e.g. depends on the location of the client) we could use participatory methods for the clients to share this information with each other. This would result in a much more scalable system, where the information flow is decentralized.

Finally, we have proposed research direction worth pursuing in the future.

5. Acknowledgments

Supported by the ÚNKP-17-4-IV New National Excellence Program of the Ministry of Human Capacities

References

1. D. P. Anderson, "BOINC: a system for public-resource computing and storage," in *Fifth IEEE/ACM International Workshop on Grid Computing*, 2004, pp. 4–10.
2. L. V. Jánoky, "Participatory Systems and Their Potential for IoT Applications" in *Proceedings of the Automation and Applied Computer Science Workshop 2017*, 2017, pp 103-114.
3. J. Braun, P. Ekler, and F. Fitzek, "Network coding enhanced browser based Peer-to-Peer streaming," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, 2016, pp. 2104–2109.
4. "PeerMesh World First Peer Powered GIS." [Online]. Available: http://peermesh.io/. [Accessed: 14-Nov-2017].
5. C. Z. 1 T. Aviv and Israel, "Peer5," *Peer5*. [Online]. Available: https://www.peer5.com/. [Accessed: 14-Nov-2017].
6. "Swarmify | Video Made Effortless." [Online]. Available: https://swarmify.com/. [Accessed: 14-Nov-2017].
7. D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@Home: An Experiment in Public-resource Computing," *Commun. ACM*, vol. 45, no. 11, pp. 56–61, Nov. 2002.
8. S. M. Larson, C. D. Snow, M. Shirts, and V. S. Pande, "Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology," *arXiv:0901.0866 [physics, q-bio]*, Jan. 2009.
9. C. Fishwick, "Tomnod-the online search party looking for Malaysian Airlines flight MH370," *The Guardian*, vol. 14, p. 37, 2014.
10. "Your Browser Could Be Mining Cryptocurrency For a Stranger," *WIRED*. [Online]. Available: https://www.wired.com/story/cryptojacking-cryptocurrency-mining-browser/. [Accessed: 14-Nov-2017].
11. G. McGuire, B. Tugemann, and G. Civario, "There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem," *CoRR*, vol. abs/1201.0749, 2012.
12. L. F. Sarmenta and S. Hirano, "Bayanihan: Building and studying web-based volunteer computing systems using Java," *Future Generation Computer Systems*, vol. 15, no. 5, pp. 675–686, 1999.
13. L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-Based Character Recognition via Web Security Measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, Sep. 2008.
14. L. V. Jánoky and P. Ekler, "The analysis of participatory computing in dynamic web environments," in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2017, pp. 000067–000072.
15. V. Pande, "Crossing the petaflop barrier," *Folding@home*. [Online]. Available: http://folding.typepad.com/news/2007/09/crossing-the-pe.html. [Accessed: 30-Apr-2017].
16. W. Mobile, "Waze," *Retrieved January*, vol. 3, p. 2013, 2012.
17. J. A. Burke *et al.*, "Participatory sensing," *Center for Embedded Network Sensing*, May 2006.
18. R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
19. J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, Feb. 2014.
20. N. Woolf, "DDoS attack that disrupted internet was largest of its kind in history, experts say," *The Guardian*, 26-Oct-2016.
21. J. Biggs, "Hackers release source code for a powerful DDoS app called Mirai," *TechCrunch*.
22. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.