DEMOS: A Participatory Design Approach for Democratic Empowerment of IS Users

Raphaëlle Bour(✉), Chantal Soule-Dupuy, and Nathalie Vallès-Parlangeau

IRIT, Université Toulouse 1 Capitole, Toulouse, France
{raphaelle.bour, chantal.soule-dupuy, nathalie.valles-parlangeau}@irit.fr

Abstract. The issue of democracy in society is at the heart of our current concerns. The organizations and their information systems are also concerned by this issue. Democracy in organization requires a debate about norms, values and language encapsulated in the information systems. The participatory design approaches address this issue by proposing a democratic empowerment for users during design phase of projects. To go further, we propose a structured method to integrate democracy into information systems. This method named DEMOS for DEsign Method for democratic information System is described and then illustrated by a real experiment provided by a "lifelong training" service at the University. All aspects of the method are addressed: from elicitation phase to implementation. We particularly focus on techniques and tools used during the design phase.

Keywords: Democracy · Method engineering · Information system design · Requirements engineering · Viewpoint · End-users · Participatory design · Agility · User centered design

1 Introduction

The question of democracy in society is a huge topic. One thing is certain, we can establish a parallelism between democracy in political society and democracy in organizations. The democracy in organizations is built through a participation of co-workers and a "high rate of empowerment" [1]. This process of democratization in organizations requires a particular focus on Information Systems (IS). In fact, an IS is not neutral. In his book Brey speaks about "embedded values" in the IS [2]. Mingers says that IT (Information Technology) systems embed particular values which have a "moral impact" [3]. These embedded values take the form of standards, quantification conventions, indicators. In 2010, Floridi proposed to elicit those embedded values and to take them into account during the IS development [4]. Salles and Colletis describe a "three-level grid" highlighting the link between representations, models and norms [5]. In our view, these observations confirm the need of democracy. On the one hand, those norms and values need to be debated, "deliberated and recognized" [3], in a democratic way. On the other hand, if we agree with Salles to say that « democracy is considered above all else to guarantee access to a plurality of worldviews" [6], a democratic IS has

https://doi.org/10.1007/978-3-030-33223-5_32

to respect viewpoints. For that, the end-users representing different viewpoints have to be considered in the system design. To go further, viewpoints must be implemented in the IS. The IS should not conform only to a dominant viewpoint. In the continuity of Van Den Hoven [7], we propose a "proactive integration" of democracy with a Design Method for demOcratic information System, named DEMOS. Our method proposes to integrate democracy in two ways. Firstly, we propose a democratic design method, which lets users debate about IS values and norms, and to bring out viewpoints. Secondly, we propose a democratic IS, which respects viewpoints identified in the design phase and implements them.

In this article, we first present a state of art divided into two parts. First paragraph is about user involvement in design approaches, second paragraph is about viewpoint concept. Then, we identify specific issues for a democratic IS and present how DEMOS can address them. We illustrate this part with a feedback from a real experiment conducted with the "lifelong training" service at the university.

2 The State of Art

The lack of users input during design has been identified as being a major factor in the failure of IS to be adopted by users [8]. The users' participation is a way to increase functional qualities of the system and to be as close as possible to their needs. It can also be a way of democratic empowerment for users, by a direct participation in decision making [9]. In the IT literature, we find several levels of users' involvement in projects: from considering the users as a "subject of study" in User Centered Design (UCD), to users playing a more collaborative role in co-operative design. With Participatory Design (PD), the user drives the design process himself [10]. With UCD, design team analyses users' interaction, and the direct users' input in design decision making is limited. The software developers still "lead the process", whereas users participate by refining their ideas [11]. The co-operative design seeks to find ways to co-operate with users in the design process [10]. The users interact with a prototype of a system being developed and can provide input in the design process. The user has a consultative role, but it is not sufficient to speak about empowerment. PD is the most involving approach in which participation of people in the co-design [12] of the IS is a "central tenet" [13]. The user is considered as a partner and no more as a subject as in UCD [14]. The aim is to increase system quality, and to empower people by a "higher level of participation in decision making" [15]. As Simonsen says, participation is absolutely necessary. According to him, it is a "basic human right" for users to have the opportunity to influence the design and implementation processes if they are affected by the changes resulting from designing and implementing [16]. Some authors like Sanders defines PD as a "democratic approach" [11]. In fact, PD approaches can provide a democratic empowerment if users participate in "defining project objectives and initial plans" [17]. In this sense, DEMOS can be considered as a PD approach.

In software engineering, viewpoint may have different meanings, as we present in this second paragraph. Today, the importance of involving end-users as stakeholders in the Requirements Engineering (RE) phase is well established [18]. The common point of each participative RE process is that authors make no distinction inside the users

group. They only make a distinction inside the stakeholder's group, like in the IEEE standard where stakeholders are characterized as client, owner, operator, architect, developer and users. Moreover, goal oriented RE is a way to elicit a project motivation, but rarely address norms and values issues with a democratic debate. When authors use the term "viewpoint", they also have different visions. For Kotonya, viewpoints are clients of the system, as in a client-server system [19]. With Sommerville, viewpoints are considered in a multi-perspective way. The aim is to separate stakeholder's concerns. The end-users are again considered as one viewpoint [20]. In the field of computer design, the viewpoints are often attached to different project actors: designers, architects, end-users, etc. For its part, the European Standard CEN recognizes four points of view: functional, informational, resources, organizational. These classifications do not correspond to our problematic, which focuses on the different user's viewpoints only. About user's viewpoint, there are different ways to tackle the problem. In terms of RE, we are talking about user viewpoint modeling [21]. Unfortunately, this form of modeling does not continue beyond the requirements during the design work. The requirements model will not be translated into a conceptual model. With component oriented information systems, viewpoint modeling is managed by base schema and view schema concepts [22]. This form of modeling is primarily a way to architect the system and remains a designer vision. The way to deal with the issue of user views that is closest to ours is handled by Nassar. He proposes an adaptation of the UML diagram with the notion of view extensions [23]. But here again, it is an answer to a technical problem related to access rights, and the solution provided is not at all user-oriented. Then, viewpoint issue is often seen in a technical perspective to separate concerns, or to separate roles, but never as a respect guarantee of democracy.

3 DEMOS: A DEsign Method for demOcratic Information System

DEMOS is presented in the form of a MAP: a "navigational structure" [24] developed by Rolland. It allows presenting the method as a selection of intentions (circles) and appropriate strategies (arrows) to achieve it (Fig. 1).

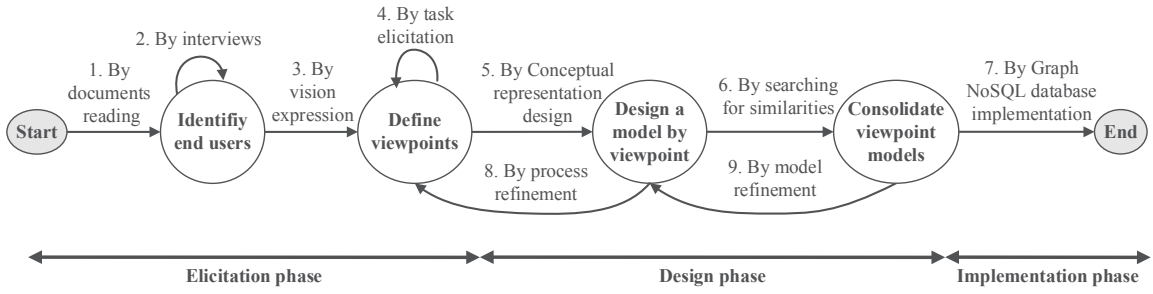

Fig. 1. General view of DEMOS

There are four intentions in the proposed method. Each intention in the MAP is a way to solve an issue that the method addresses:

- *Identify end-users* to **involve end-users in a participatory and democratic process**
- *Define viewpoint* by **allowing a debate to let viewpoints emerge**.
- *Design a model by viewpoint* to **design a democratic IS which considers these viewpoints**.
- *Consolidate viewpoint models* to **provide traceability of viewpoints.**

In the following sections, each strategy is described, with a brief list of means used and expected results. The sections are grouped by intentions, for a better understanding. From January to June 2018, DEMOS has been used for a real project. The project focused on the implementation of an attendance management tool for the Toulouse 1 Capitole university's "lifelong training" service. This experiment was conducted with 8 end-users: 3 teachers, 4 schooling managers and the "lifelong training" service manager. The aim of the project was to develop a prototype to be tested by users. The designed software is currently being implemented. In this paper, third et fourth intention will be illustrated with feedback from this recent experiment.

3.1 First Intention: Identify End-Users

Our first issue is to involve end-users in a participatory and democratic process. For that, the first intention: *Identify end-users*, is the starting point of our participative approach. We developed two strategies to achieve this intention: ***by interviews*** and ***by document reading***. Both strategies are achieved with the client and the main managers concerned by the project. They give information during interviews: motivation of the project, issues to solve and constraints (as brake of users). Business documents complete this information: organigram, specifications. The client must validate the list of end-users obtained. These two strategies have been chosen to be complementary: they give an analysis of prescribed work and real work [25] as said in work ergonomics. Documentation is a representation of the prescribed work and gives a first list of end-users, whereas interviews give information about real work.

3.2 Second Intention: Define Viewpoints

Our second issue is to allow a democratic debate to let viewpoints emerge. The second intention: *Define viewpoints* is crucial. In fact, these viewpoints are the starting point of the rest of the method. We developed two strategies to achieve this intention: ***by vision expression*** and ***by task elicitation***. Both strategies are achieved with end-users and a moderator of the method during the scoping workshop. First, the end-users are encouraged to debate about their visions, and to let viewpoints emerge. Thanks to tools as photolanguage and mind mapping, they make a breakdown of different professions and they express their vision of business domain to propose first viewpoint list. Then, they identify tasks and describe processes with a simplified BPM notation. After that, they consolidate the viewpoint list. During the experiment, two viewpoints were identified: Management viewpoint (with the schooling manager and the lifelong training service managers), and Education viewpoint (with the teachers).

3.3 Third Intention: Design a Model by Viewpoint

Our third issue is to design a democratic IS which takes into account viewpoints. The third intention: *Design a model by viewpoint* corresponds to the design phase starting point. This intention requires a strategy: ***conceptual representation design***. This strategy is achieved during several viewpoint workshop. There are as many workshops as identified viewpoints. The aim is to obtain one conceptual model by viewpoint. During the workshop, each end-user uses his own vocabulary to express concepts he manipulates to achieve his tasks. They do not have to adapt to other norms, values or to use another vocabulary. Because the description of concepts is sometimes confusing for end-users, we proposed several activities to achieve this goal: Photolanguage, brainstorming, markers, etc. The main outputs of these section are simplified class diagrams for each viewpoint. During the ***process refinement*** strategy, work on processes during the viewpoint model designing intention can affect viewpoint identification. For example, end-users can realize at this step that a viewpoint is badly or insufficiently described. In this case, a new definition of the viewpoint is necessary. Figure 2 presents an instantiation of a part of DEMOS meta-model corresponding to the experiment with lifelong training service.

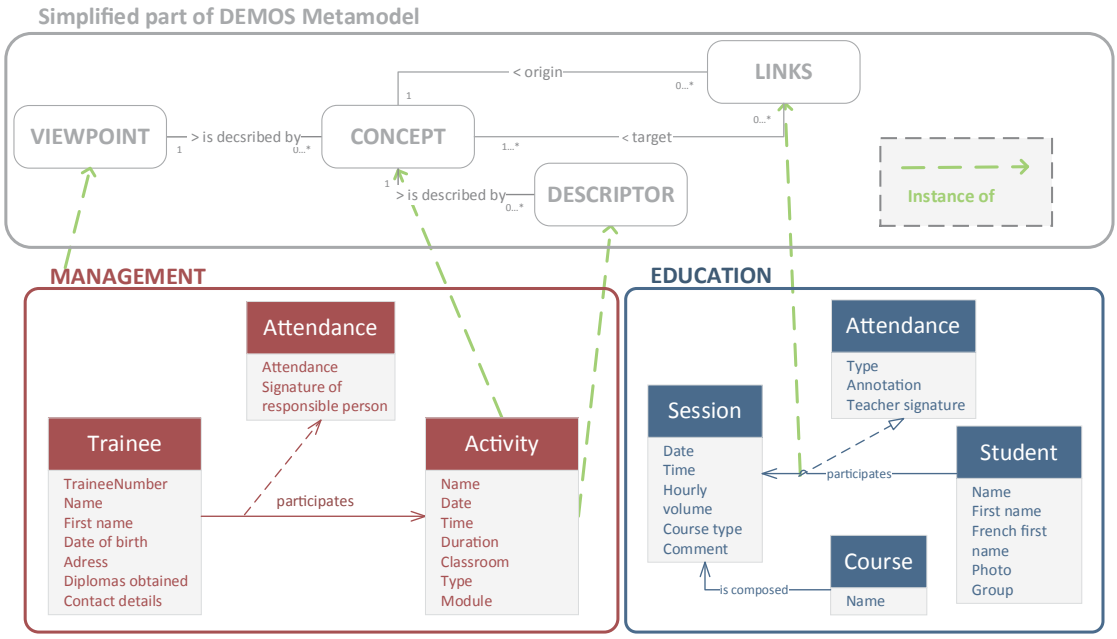

Fig. 2. Instantiation of the strategy ***By conceptual representation design***

We obtained two different conceptual models, and we present here just a little part of them. The vocabulary employed on each model corresponds to the corresponding viewpoint, described with concepts, descriptors and links, using a simplified UML class diagram formalism. Here the structure of the attendance representation is not the same for each viewpoint: the management viewpoint speaks about trainees whereas the education viewpoint is interested in students. Furthermore, while the education

viewpoint is only interested in the class attendance, the management viewpoint needs to attest about attendance for other activities like internships.

3.4 Fourth Intention: Consolidate Viewpoint Models

The fourth intention: *Consolidate viewpoint models*, corresponds to the last step of the design phase. This intention requires a strategy: ***searching for similarities***. This strategy is achieved during a sharing workshop where all participants are grouped. This intention is crucial because even if each viewpoint corresponds to a conceptual representation, some elements between these representations are common. Sometimes these elements are identical, sometimes they are named differently but have the same meaning, and sometimes they are simply organized differently. The similarities between models must be identified to share the same IS. For these reasons, after the moderator organizes the models pooling, the end-users can search for similarities. With the ***representation refinement*** strategy, work on conceptual representation during the viewpoint model consolidation section can affect the previous designed model. In fact, searching for similarities can allow for some new concepts for a viewpoint to emerge. In this case, viewpoint models must be completed.

Figure 3 presents an instantiation of a part of DEMOS metamodel corresponding to the experiment with the lifelong training service.

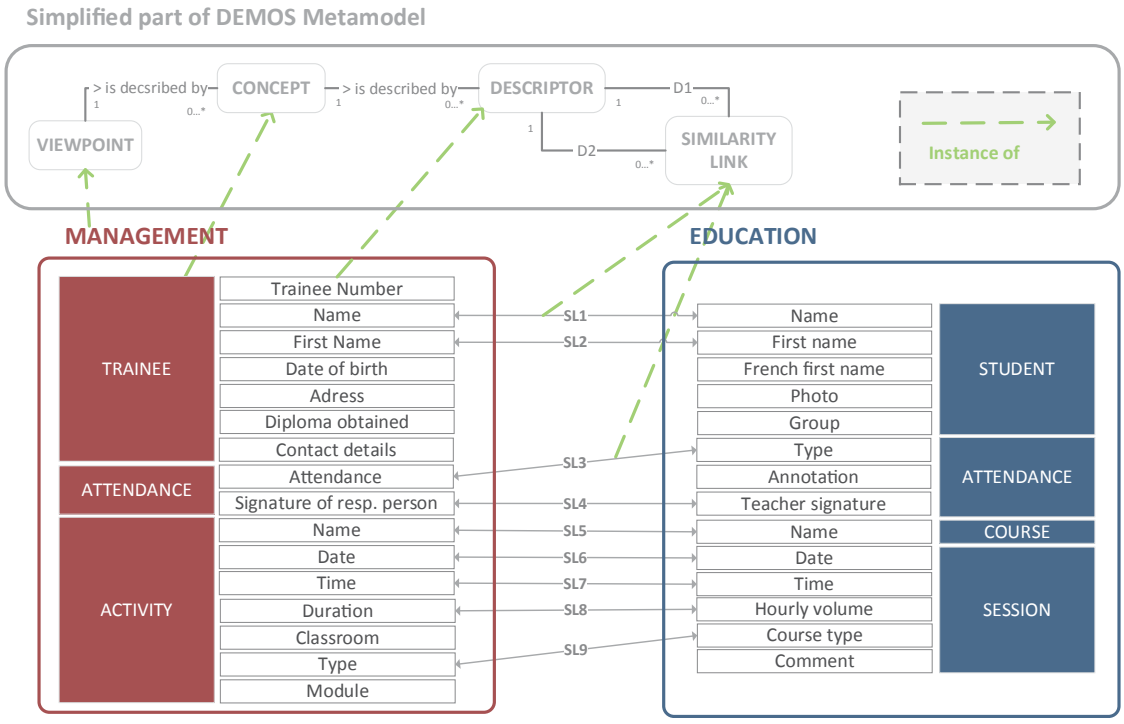

Fig. 3. Instantiation of the strategy ***By searching for similarities***

In this illustration, we have selected just a part of the similarity links, corresponding to previous viewpoint models. Here for example, the name of a trainee and the name of a student are similar, so they must be linked. Each similarity link is a way to create a

data repository. It guarantees that even if there is an implementation of several conceptual models, objects are shared through this repository.

The last strategy: ***By Graph NoSQL Database implementation*** is the purpose of the method because it is an implementation of the viewpoints. To keep viewpoint traceability in the IS, it was necessary to implement a database structure in accordance with viewpoints. Thus, the viewpoints can continue to exist independently during the life of the IS. If a viewpoint evolves, the other viewpoints are not impacted, expected by re-creating similarity links. One efficient solution to implement several linked models is Graph No SQL Database, we have adopted this strategy. In the future, other strategies could be added with other technical solutions.

4 Discussion

We have shown with DEMOS that a structured design method can contribute to integrate democracy in IS, and we have illustrated our proposition with a concrete case. Following the experiment, we have evaluated DEMOS with semi-structured interviews with end-users. After a review of the method and of the results obtained during the experiment, we have conducted an interview with each participant to evaluate the method intentions, the method strategies and the method results. The evaluations revealed that end-users understand intentions of the method. They understand both aspects: a democratic process for a democratic IS which respects their viewpoints. The viewpoint notion that was not obvious to them at first became clearer during the workshops. Moreover, for users, sequencing of steps was coherent according to the intentions. Overall, they were assisted by techniques and tools used during the process, especially by the photolanguage activities. At the end, they are satisfied with the method results, which are consistent with what they have expressed. The final software is under development and was not considered for evaluation.

The issue addressed in this article is democracy in IS. We consider this issue with two different perspectives: how to integrate democracy in IS conception approaches and how to bring democracy into IS. It requires a participative approach involving end-users, respect of viewpoints inside end-users' group, and traceability of these viewpoints. During the experiment, and according to the evaluation, each point has been respected. However, the implementation of viewpoints is guaranteed by the implementation of a database structure. As future work, we want to add other intentions to implement activity model and interface model.

References

1. Jardat, R.: How democratic internal law leads to low cost efficient processes. Soc. Bus. Rev. **3**(1), 23–40 (2008)
2. Brey, P.A.E.: Values in technology and disclosive computer ethics. In: Floridi, L. (ed.) The Cambridge Handbook of Information and Computer Ethics, pp. 41–58. Cambridge University Press, Cambridge (2010)
3. Mingers, J., Walsham, G.: Toward ethical information systems: the contribution of discourse ethics. MIS Q. **34**(4), 833–854 (2010)

4. Floridi, L.: The Cambridge Handbook of Information and Computer Ethics. Cambridge University Press, Cambridge (2010)
5. Salles, M., Colletis, G.: How to deal with the conflicting views of the world expressed in regional economic development policies. In: International Conference of Territorial Intelligence, Besançon 2008, Besançon, France, p. 10 (2008)
6. Salles, M.: Decision-Making and the Information System, vol. 3. Wiley, Hoboken (2015)
7. van den Hoven, J.: Moral methodology and information technology. In: Himma, K.E., Tavani, H.T. (eds.) The Handbook of Information and Computer Ethics, pp. 49–67. Wiley, Hoboken (2008)
8. McConnell, S.: Rapid Development: Taming Wild Software Schedules. Microsoft Press, Redmond (2006)
9. Kautz, K.: Investigating the design process: participatory design in agile software development. Inf. Technol. People **24**(3), 217–235 (2011)
10. Andre, K., Christian, N.: Participatory design, user involvement and health IT evaluation. Stud. Health Technol. Inform. **222**, 139–151 (2016)
11. Ferrario, M.A., Simm, W., Newman, P., Forshaw, S., Whittle, J.: Software engineering for 'social good': integrating action research, participatory design, and agile development. In: Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014, Hyderabad, India, pp. 520–523 (2014)
12. Sanders, E.B.-N., Stappers, P.J.: Co-creation and the new landscapes of design. CoDesign **4** (1), 5–18 (2008)
13. Kensing, F., Blomberg, J.: Participatory design: issues and concerns. Comput. Support. Coop. Work CSCW **7**(3–4), 167–185 (1998)
14. Dell'Era, C., Landoni, P.: Living lab: a methodology between user-centred design and participatory design: living lab. Creat. Innov. Manag. **23**(2), 137–154 (2014)
15. Kujala, S.: User involvement: a review of the benefits and challenges. Behav. Inf. Technol. **22**(1), 1–16 (2003)
16. Simonsen, J.: Routledge International Handbook of Participatory Design. Routledge, London (2013)
17. Dearden, A., Rizvi, H.: Adapting participatory and agile software methods to participatory rural development. In: Proceedings of the Tenth Anniversary Conference on Participatory Design 2008, Indianapolis, IN, USA, pp. 221–225 (2008)
18. Milne, A., Maiden, N.: Power and politics in requirements engineering: embracing the dark side? Requir. Eng. **17**(2), 83–98 (2012)
19. Kotonya, G., Sommerville, I.: Requirements engineering with viewpoints. Softw. Eng. J. **11** (1), 5 (1996)
20. Sommerville, I., Sawyer, P.: Viewpoints: principles, problems and a practical approach to requirements engineering. Ann. Softw. Eng. **3**, 101–130 (1997)
21. Darke, P., Shanks, G.: User viewpoint modelling: understanding and representing user viewpoints during requirements definition. Inf. Syst. J. **7**(3), 213–219 (1997)
22. Caron, O., Carré, B., Muller, A., Vanwormhoudt, G.: A framework for supporting views in component oriented information systems. In: Konstantas, D., Léonard, M., Pigneur, Y., Patel, S. (eds.) OOIS 2003. LNCS, vol. 2817, pp. 164–178. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45242-3_16
23. Nassar, M.: VUML: a viewpoint oriented UML extension, pp. 373–376 (2003)
24. Rolland, C., Prakash, N., Benjamen, A.: A multi-model view of process modelling. Requir. Eng. **4**(4), 169–187 (1999)
25. Vilela, R.A.D.G.: et al.: Work ergonomic analysis and change laboratory: similarities and complementarities between interventionist methods (2015, Unpublished)