EAI Endorsed Transactions
on Collaborative Computing

Research Article **ICST.ORG**

Optimistic Scheduling: facilitating the collaboration by prioritizing the individual needs

Salvatore F. Pileggi[1]

[1]INRIA & UPMC-LIP6, Paris, France.

Abstract

The collaboration among people is one of the key factors for the optimization of many processes and activities. The efficiency and the effectiveness of the collaboration has an intrinsic value which significantly affects performances and outcomes, at a quantitative and a qualitative level both. Open communities, as well as spontaneous or predefined virtual organizations, are demanding for a more solid and consistent support for activity scheduling and managing in a context of flexibility and respect of individual needs. This paper proposes a privacy-friendly model that can be materialized in concrete tools and applications to support virtual organizations in the scheduling and management of the most valuable resource: the time. The model is formally defined and, than, analysed and evaluated by simulations as the function of complex user behaviours. Finally, an implementation of the basic prototype aimed at a large scale deployment is described.

Received on 18 November 2015; accepted on 15 December 2015; published on 21 December 2015

doi:10.4108/eai.21-12-2015.150821

1. Introduction

The collaboration among people is one of the key factors for most processes and activities. The efficiency and the effectiveness of the collaboration has an intrinsic value which significantly affects performances and outcomes, at a quantitative and a qualitative level both [1]. Common but significant examples are easy to find from trivial observations of the real world: researchers, for instance, can have excellent individual skills, expertise and motivation; but it is the convergence of their experience and ability that allows to reach the best results; it is easy to detect like addressing a competitive business implies a pragmatic approach that takes into account multiple perspectives and contributions from different members of specialized heterogeneous teams; it is evident in sport competition at any level: explicitly in team sports, as well as implicitly in individual ones, where top-level (and not only) athletes are supported in the background by specialized teams. The obvious conclusion is that individuals are (or can be) good but, together, they are better. Therefore, a team is much more than people working together.

For a long time the optimization of human resource has been investigated, especially inside specific context aimed at the maximization of the productivity. Classic models commonly applied by companies in the real world have evolved at a theoretical level to integrate a more flexible philosophy (e.g.[2]). More recently some of those models are being considered in practice, as many companies are progressively leaving from classic schemas to evolve towards novel approaches where individual needs and effectiveness converge, under the realistic assumption that personal and collective development come together. It is the case of the "flextime", in which employees can choose when they work, subject to achieving total daily, weekly or monthly hours and to the necessary work being done. More recently, successful and powerful companies have experimented cutting-edge solutions aimed at the empowerment of workers that should fortify the relationship with the company. As an example, Netflix is proposing a model that assumes an unlimited number of holiday days that employees can take at their best convenience as long as the planned work is performed and main goals achieved.

In this paper we uniquely address the problem of the activity scheduling as a response to the needs of new emerging organizational models. Indeed, open

*Email: flavio.pileggi@lip6.fr
Web-page: http://www.flaviopileggi.net

communities, as well as spontaneous or predefined virtual organizations (VOs) [30], are demanding for a more solid and consistent support for activity scheduling and managing in a context of flexibility and respect of individual needs.

This paper proposes a privacy-friendly model called *optimistic scheduling* due to the implicit positivity that drives interactions among people in the paradigm. It can be materialized in concrete tools and applications to support virtual organizations in the scheduling and management of their most valuable resource: the time.

However, simple observations in everyday life clearly show unpredictable human behaviour, even in simple and well-known situations. Therefore, the model is first formally defined and, than, analysed and evaluated by simulations as the function of complex user behaviours. As demonstrated, the human factor has a critical impact on the model performance.

Finally, an implementation of the basic prototype aimed at a large scale deployment is proposed. An empirical overview of collaborative tools demonstrate as the simplest approaches (e.g.[3]) usually reach the best results, meaning products are well accepted by users and indeed they are usable in practice.

The introductory part of the document is aimed at the contextualization of this work and follows with an overview of the latest tools on the market, with a deep description of the reference use cases and, finally, with a brief explanation of the methodological aspects. The second section describes *Optimistic Scheduling* that is analysed and discussed in the third section. Then, in the section 4, an implementation of a basic tool inspired by the model for a large scale deployment is described. As usual, the paper ends with a section about conclusions and future works.

This paper is an extended version of [16], recently presented at *CollaborateCom 2015*. The original paper focuses exclusively on the model and on its behavioural analysis. This extended version definitely addresses a wider scope, integrating the original contribution with an extended discussion, a short overview of the related work, as well as details about the current implementations and applications on a large scale.

1.1. Related Work

Cloud and mobile technology [17] has enabled a massive development of applications and tools that users can get and run through commercial ecosystems (e.g. Apple, Google, Microsoft). The impressive number of apps currently available on the most common marketplaces makes the normal "state of the art" of popular applications (including shared calendars and similar tools) hard to be proposed. Also a simple survey looks definitely far away from an exhaustive analysis.

From a quick overview of commercial solutions, the most novel tools on the market offers features that current reference tools (e.g. Google Calendar) are missing. That is the case of *Kalendi* [4] which includes features to add attachments to event invites, to schedule individual SMS reminders, to publish calendars online and to set up unlimited calendars to share across groups. *UpTo* [5] proposes a calendar model which provides a kind of "new dimension", providing the ability to discover important events ahead of time and plan accordingly. The key idea is to get a more complete view of everything coming up that matters to users but without the clutter. This calendar has two layers: the front layer is the existing calendar and the back layer includes calendars the user follows based on likes and interests. *Teamweek* [6] is designed according to a model that should efficiently support those who are collaborating on time-sensitive projects, which require a step-by-step approach. Many other examples could be reported. A full overview is out of the scope of this paper.

Summarizing, the most dominating trends appear in coherence with the current technological climate [18] that pushes towards a progressive socialization of tools and applications. The main limitation of those products look the base model itself: shared activities are always scheduled with a kind of implicit high priority established a priori and somehow passively "pushed" to individuals. The model proposed in this paper works with a completely opposite logic and pretends to push the cooperation or collaboration among people by priorizing individual needs. Even though it can appear like a contradiction, we are pretty confident that this approach can suit the requirements of emerging organizational models, as well as the philosophy of many professionals and cutting-edge organizational models.

1.2. Beyond the state of the art: overcoming a simple use case

The primary scope of this work is to provide a privacy-friendly model for effective and efficient time sharing.

With the support of current digital tools, the dynamic scheduling of a shared event is commonly done according to two different main models:

- *Scheduling by invitation*. The organizer sends an explicit invitation to all the expected participants by using some shared channel (e.g. email, message, sms, social network). Due to the extensive use of emails and messaging in both private and professional life, this is evidently the simplest and, indeed, the most used method. But this model is very vulnerable in fact. First of all, considering a group of peers, the event cannot be considered as committed if confirmations are not received from all the group members. If at least one of the expected participants rejects the invitation, in theory the organizer needs to restart the process sending a new invitation. Furthermore, a missed notification from at least one member of the group can generate misunderstandings (not received? Not seen? Not interested? Unable to respond?), requires further actions from the organizer (e.g. reminders) and can lead to a situation of potential deadlock. A different organization of the group (e.g. assuming a leader or people that are requested but not strictly needed) can mitigate the impact of those situations but, in general, the synchronization schema among users is pretty poor even considering the support of specific tools.

- *Scheduling by poll*. It is the most common alternative to the method previously described. The organizer proposes a set of possible slots and participants are asked to give their preferences in order to reach an agreement. Having a poll of possible chooses reduces significantly the vulnerability of the model as the organizer, always assuming a group of peers in which everyone is requested to join the scheduled event, has to set up a new poll only if there is no agreement on any of the proposed slots. On the other hand, a lack of response from some member proposes the same problems described above. This method is commonly supported by specific tools (such as the very famous *Doodle* [7]) to minimize human efforts. The poll allows to simultaneously reason on a set of possibilities instead of on just one but, at the same time, can be quite uncomfortable and inefficient: waiting for the result of the poll, people are blocking a part of their time according to the preferences in the poll.

Last but not the least, both models present a further common weakness: what if a group member changes his plans after a commitment? In general, under the assumptions mentioned above, the process should start over. In practice there is no support provided by existing tools.

1.3. Methodology and Approach

The model proposed in the paper has been designed combining practical needs and requirements (e.g. usability and privacy preserving) with a research-oriented approach aimed at complex studies in the field of social science [8] (e.g. human behaviour [9]) and other domains (e.g. organizational models [31]). The key methodological idea assumes scientists and application designers as the main actors of a feedback process that produces concrete tools on one side and valuable data for analysis and improvements/refinements on the other side.

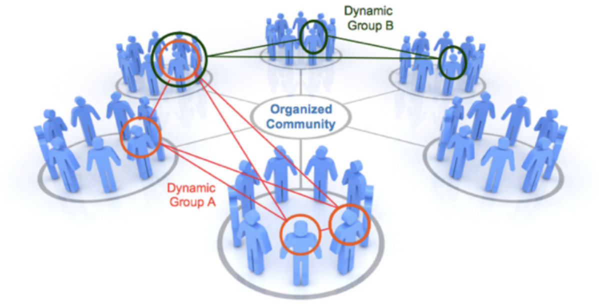


Figure 1. Virtual Organizations as a reference model.

One of the key factor that affects the whole understanding of this work and that determines clear design decisions at the time to materialize the model in a concrete tool is the *Reference Community Model* (fig. 1). Indeed, proposing a completely generic approach is hard and could result non-effective or ambiguous. On the other hand, abstractions could provide a simplified view of complex realities with a consequent and fundamental lack of realism. In the context of this work, the community is understood as a whole, meaning that users can interact among them at a global level. However, the ideal application domain assumes a *virtual organization* model where existing and relatively static real groups (e.g. companies, institutions, teams, group friends) can be integrated with dynamic groups that can often change in the time (such as in cooperative projects) or that are defined on the fly as a function of concrete tasks or activities.

It is also assumed that users inside groups are peers. Even though implicitly maintaining some of the typical roles (such as the "organizer" or "moderator" of a shared activity), that is not always realistic considering that real groups are often organized according to some structure or hierarchy. Moreover, it is assumed that a shared activity inside a dynamic group requires the participation of all the members of the group. This ideal case could not suit virtual organizations that propose some kind of internal priority among members, as well as other common concepts such as "optional" or "not mandatory".

Finally, the philosophy of the model assumes that individual preferences have a priority over groups. The scope is to facilitate the cooperation and to optimize the use of the time. Limiting or conditioning individual behaviours would mean re-proposing common approaches.

2. Efficient and effective time sharing inside Virtual Organizations

This section proposes the description of *Optimistic Scheduling*. This model can be materialized in concrete tools and applications designed according to its philosophy.

At the time t, each community member has his own calendar TS_P. It is considered like a private asset as no-one else in the community has access or visibility to it. Each user i performs on his private calendar a kind of preliminary filtering as he sets a priori the slots that can be used for some personal activity ts_P or shared activity ts_S, defining his personal space K_i. A simple example of preliminary filtering is represented by a work calendar that only includes normal work hours and that doesn't include leaves or other kind of absences known a priori.

The whole space K of slots is given merging users' spaces according to eq.1.

$$K = \bigcup_i K_i \quad \forall i \tag{1}$$

On that filtered calendar, a user i can schedule personal activities (eq.2) for any available slot $k \in K_i$.

$$TS_P^i(t) = \sum_k ts_P^i(k,t) \tag{2}$$

According to the same logic, a user can try to schedule a shared event (eq.3) involving other persons (group).

$$TS_S^i(t) = \sum_g \sum_k ts_S^g(k,t), \quad i \in g \tag{3}$$

The full activity set TS for a community member is given by merging his personal activities (TS_P) and shared activities (TS_S) as in eq.4.

$$TS^i(t) = TS_P^i(t) \cup TS_S^i(t) \tag{4}$$

The whole potentially shared time can be at least equal to K assuming people have no personal activities scheduled (eq.5).

$$\bigcup_i TS^i(t) = O(K) \quad \Rightarrow \quad TS_P^i(t) = \emptyset, \quad \forall i \tag{5}$$

As previously mentioned, users can only see their own calendars. In order to get an effective guide to schedule shared events, any user can access, for any defined group he joins, a shared structure FTS obtained by merging the personal calendars and returning the anonymized complementary set according to eq.6.

$$FTS^g(t) = K - \bigcup_i TS^i(t), \quad i \in g \tag{6}$$

That structure shows (fig.2a) the slots that can be potentially used, inside a considered group, to schedule a shared event. This simple operation allows users to automatically understand the availability of a certain group in the respect of the privacy of its members. Assuming a significant size for a group, inferring information is not easy in practice, so the privacy is completely preserved.

By using those structures, whichever member in a group can schedule a shared activity for a dynamic or static group with a very good chance to be successful (fig.2b).

The semantic described implicitly defines the main global invariant (a logical assertion that is held to always be true during a certain phase of execution of a program [10]) for applications working according to this model: if a time slot ts is used by a member i of the group g for a private purpose, then that slot cannot be used for a shared activity inside any group i is member of (eq.7).

$$\exists ts_P^i(k_a,t) \quad \Rightarrow \quad \nexists ts_S^{i \in g}(k_a,t), \quad k_a \in K \quad \forall t \tag{7}$$

The model scales to a multi-group environment (fig.2c) providing an individual-specific view of each group in a privacy-friendly context.

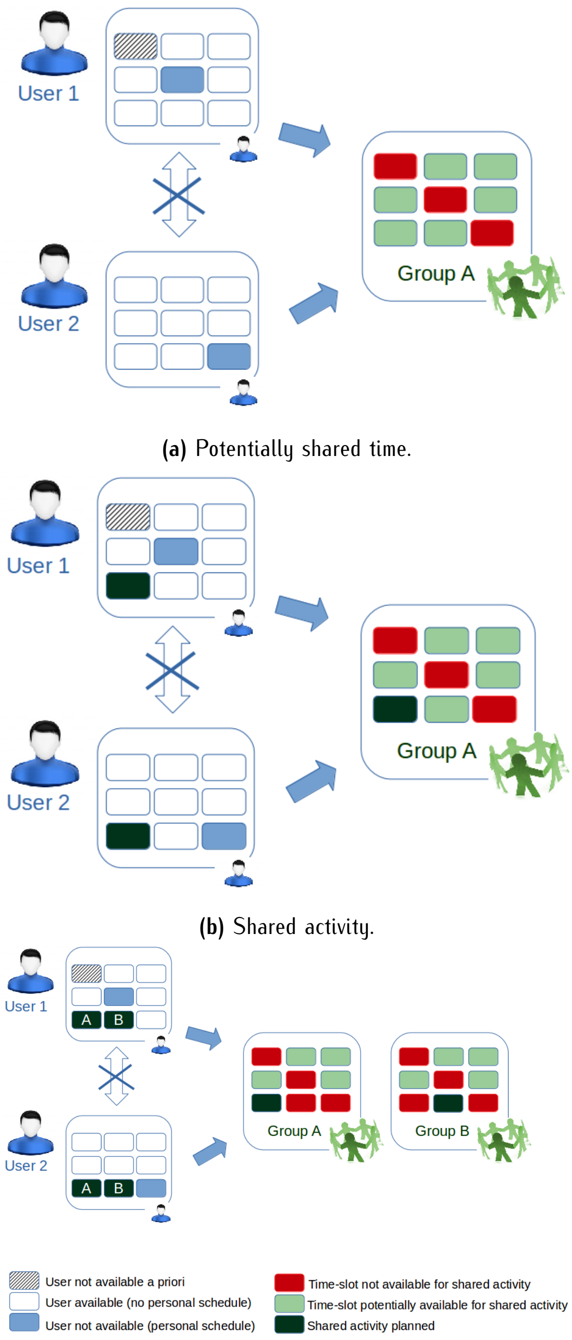
As already mentioned, users preferences have a priority on group activities according to a logic that wants to push co-operation/collaboration without adding barriers or constraints. Consequently, users can still schedule their own activities also for slots that are already used as a shared resource. In this case the invariant defined by the eq.7 is not respected any more, determining a non-valid state for the system that, coherently with the assumptions, reacts (fig.3) cancelling the shared event and notifying the interested group about. An extensive discussion about this strong condition, as well as about its implications on performance and possible variants are discussed later in the paper.

3. Model Analysis

As activities involve people, the validation and evaluation of the model have to take into account the human factor. Simple empirical observations show how unpredictable people can be, even in well know and straightforward situations.

The most significant issue for the analysis and the full understanding of the model is the definition of simple but realistic behaviours under the assumption that people are or can be unpredictable. Indeed, the simple fact that different personalities act in a different way in a given context often reduces the standardization or categorization of the human behaviour just to an academic exercise.

In this study we assume a simple and uniform behaviour at the time to schedule shared activities,


User 1
User 2
Group A
Group B
A
B
▨ User not available a priori
□ User available (no personal schedule)
🟦 User not available (personal schedule)
🟥 Time-slot not available for shared activity
🟩 Time-slot potentially available for shared activity
■ Shared activity planned

(a) Potentially shared time.

(b) Shared activity.

(c) Multiple groups.

Figure 2. Overview of the model behaviour.

which are periodically scheduled according to a regular pattern. But we assume more than one possible behaviour to manage personal activities, affecting indirectly the shared ones.

The synthetic actor that emulates the users uses a linear logic to schedule a shared event: the activity is planned and associated with a well determined time slot; if the slot is available for shared activities, then it is submitted; else there is the possibility to don't submit the request or to look for another slot. Concerning the personal activities management, there are three different possible behaviours, defined as in the follow:

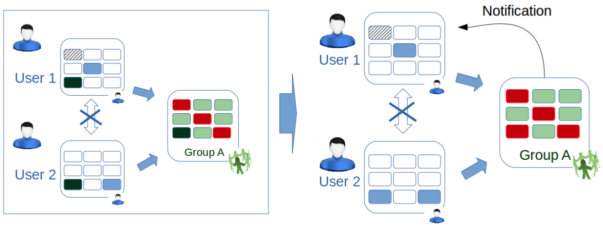


Figure 3. Cancelling a shared activity as a reaction to a global invariant violation.

- *Constructive*. The user is cooperative and, therefore, acts according to a logic that facilitates the successful scheduling of group initiatives. A constructive user schedules his activities only in slots not currently used for shared events and uses shared slots only if there is no other choice.

- *Disruptive*. This is the opposite of the previous as he schedules his personal activities prioritizing the slots currently occupied by shared events. This behaviour causes the continuous re-organization of the already scheduled shared events. It is not necessarily reproducing a malicious user, as it could also simulate an involuntary noise caused by random circumstances or periodic conflicts on schedules.

- *Random/Independent*. Between the two extremes (constructive and disruptive) there is a wide range of behaviours, including an independent user that acts according to a pseudo-random logic that doesn't take into account groups: an independent user schedules independently his activities without taking care if the target slot is currently used for some shared activity or not.

The metric to evaluate the model performance (eq.8) is directly proportional to the number of shared activities successfully scheduled and inversely proportional to the number of shared events cancelled upon request of users.

$$\alpha(t) = \frac{1 + \sum_i TS_S^i(t)}{\sum_i TS_S^i(t) + \sum_i TS_S^i(t)_{|cancelled}} \tag{8}$$

The simulations are assuming a *sliding window* to reproduce the logic time (fig.4): users have a view of a finite number of slots m and, at the generic time t, they can only schedule between the slot t+*1* and the slot t+m. As a simulation of the time, the logic transition from t to t+*1* implies the slot t no more available (past) and a new free slot t+m+*1* available.

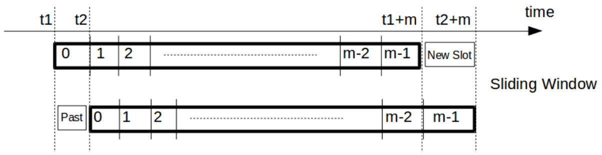


Figure 4. Sliding Window.

Furthermore, for the experiments proposed in this work, the qualitative behaviour previously described is integrated with a quantitative profile as in eq.9: the number of scheduled events tends to increase in the time. In other words, users averagely schedule a higher number of events than the number of events they cancel.

$$\frac{d}{dt}TS^i(t) > 0, \quad \forall i \tag{9}$$

For simplicity we are assuming atomic slots that don't overlap each others (eq.10).

$$ts^i_{P/S}(k_1, t) \cap ts^i_{P/S}(k_2, t) = \emptyset, \quad \forall k_1, k_2 \in K \quad \forall i \tag{10}$$

The simulations performed assume a sliding window of 12 weeks to schedule activities and members averagely available 35 hours (slots) per week. This is the reproduction of a common calendar. The calendar is assumed to be empty when the simulation starts, so there is a transitory period. The members schedule averagely an activity per day and the 25% of the planned activities are shared. The simulation ends when the system is saturated (no more possibility to schedule events due to the quantitative behaviour) or when the system has reached stationary/stable conditions.

The simulation results are showed in fig.5. The chart at the top represents the decreasing of performance when the group size increases and assumes independent behaviours (as previously defined). That is a very good approximation of the performance inside a virtual organization where users are not explicitly acting according to the model but in a kind of "neutral" mode. The chart in the middle proposes the same statistics assuming a cooperative behaviour. This emulates a community that acts according to the model. Performances are evidently higher than the previous and decrease only as a function of the natural saturation of the system determined by the quantitative behaviour (fig.9). The chart at the bottom provides an overview of the potential impact of disruptive behaviours on the whole performance. As showed, if one or more members is acting according to a disruptive behaviour, then performances quickly decrease and the system tends after a very short time to the instability.

Disruptive behaviours are part of the real life and have to be taken into account at the time of designing real tools. They are easy to detect in common mechanisms such as invitation and polls due to the explicit character of the interactions. On the contrary, disruptive behaviours are hidden in a privacy-friendly context.

At a model level, the global invariants (eq.7) can be relaxed to mitigate the effect of disruptive behaviours. This approach introduces at least one significant and critical trade-off between functionality and privacy/simplicity. Indeed, as showed in fig.6, assuming that a slot inside a group can be used simultaneously for a personal and a shared activity protects the system from disruptive behaviours, as well as it support role-based and hierarchical virtual organizations. But it also introduces a complexity and ambiguities in the understanding and the management of the system state. Considering anonymous non-availabilities, the organizer cannot know who is missed, so the further steps of the activity planning could be negatively affected. On the other hand, concessions about privacy could invalidate most premises and, consequently, modify significantly the whole model focus. In any case, integrating a complex state that assumes the simultaneous use of shared slots limits tools autonomy and, in general, applications could miss their aimed simplicity.

4. Applications and Implementations

Optimistic Scheduling has a generic focus and, consequently, can be used to approach specific problems related to the activity scheduling in the context of different application domains.

This section proposes the description of a generic purpose tool for activity scheduling and a brief discussion on the potential applications of the model in e-Education [19][27], as en example of target domain.

4.1. A generic purpose tool supporting deployment on a large scale

Tools and applications designed according to this model can be understood like systems working at a low scale (such as a corporative tool), as well as like services working at a global scale (e.g.[32]). This implementation focuses on large scale deployments.

To assure a scalable and fault-tolerant environment, the application is implemented upon replicated databases. More concretely, a noSQL [11] philosophy, assuming key-value interface, is adopted and data structures are CRDTs [10][12] to support a partition-tolerant [13] deployment on a large scale in a context of eventual consistency [14][15].

The design of the application can follow two main common approaches for distributed systems:

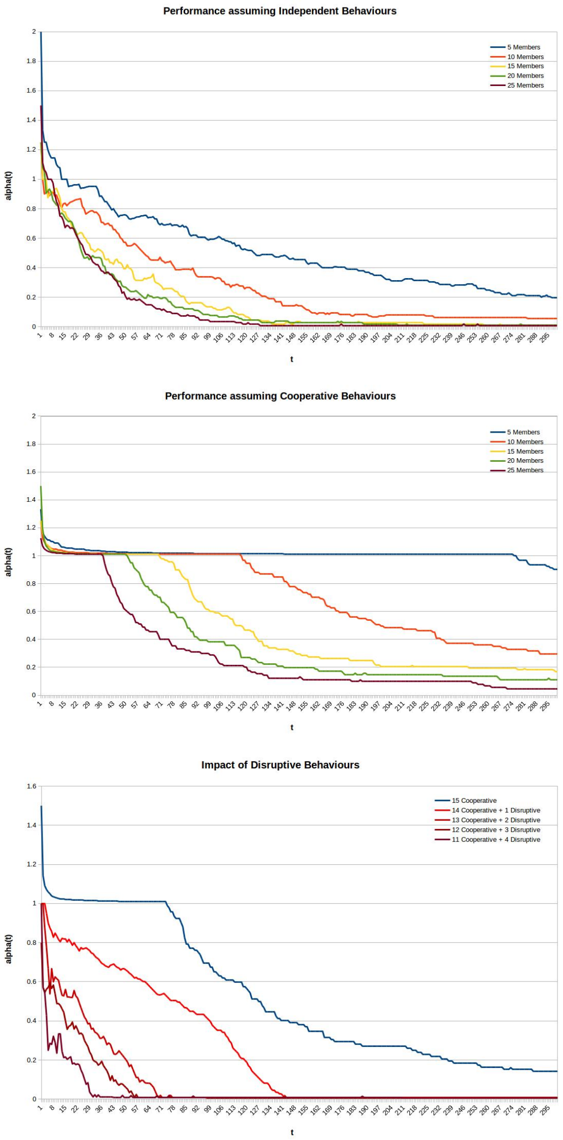


Figure 5. Simulation outcomes.

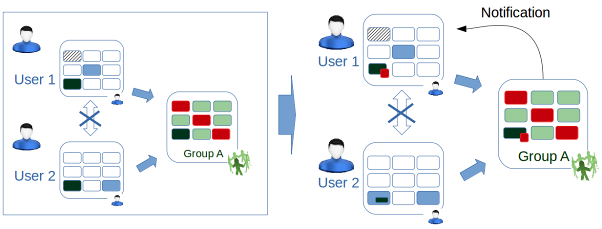


Figure 6. Relaxing global invariants.

- *Computation in the cloud* [20] which is commonly defined as a type of Internet-based computing, where different services, such as servers, storage and applications, are delivered through the Internet.(fig.7, top).

- *Computation at the edge* (e.g. [21]) that pushes the frontier of computing applications, data, and services away from centralized nodes to the logical extremes of a network (fig.7, bottom). It enables analytics and knowledge generation to occur at the source of the data. This approach requires leveraging resources that may not be continuously connected to a network such as laptops, smart-phones, tablets and sensors.

This implementation is in fact a fully distributed application composed of local agents without any central coordination that assumes data in the cloud and computation at the edge. The *consensus* to assure the system converging to a correct state is defined as part of the application logic. The weak consistency model allows off-line activity and eventual synchronization once online.

The application distinguishes between:

- *State* of the system, which is a correct state according to the application logic.

- *Meta-state* of the system, that is a kind of pseudo-state that can evolve to different states as the function of the application logic (fig. 6). A meta-state is a situation of conflict that is not preserving the application invariants.

Concerning the data model, there are two main possibilities in terms of data structure to use:

- *Sets of data*. Data is modelled on a pure key-value interface. Each group is associated with an unique identifier that is also the identifier of the corresponding data set. Each element of the set includes a prefix of the time slot, the anonymized user identifier and an informative part. This representation is simple and explicitly allows both the representation of states and meta-states of the system.

- *Maps of data*. Using maps, keys do not contain all the information but points to an object. A different map is defined for each group and keys are the time slots. Therefore, the information is the combination of the key and the object. The main advantage of maps is the explicit representation of the last state for each slot, due to the effective concurrency on the keys. On the other hand it requires to maintain the history of the updates to assure the convergence and, therefore, the correct implementation of the application logic.

For this implementation, sets are preferred as they are a simpler solution.

Fairness is not completely assured, as the causality of events is not guaranteed [24]. Indeed, actions happening almost simultaneously can generate conflicts that,

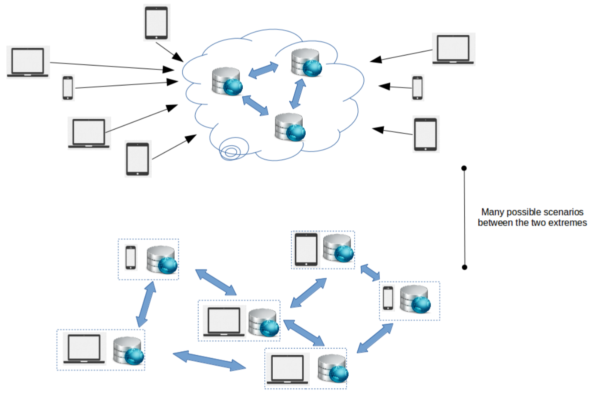


Figure 7. Computation in the Cloud (top) and computation at the edges (bottom) assuming replicated databases.

according to a fair approach, should be solved giving priority to actions generated before. Typical situations of conflict happen when users try to use simultaneously the same free slot (fig.9). However, assuming a low latency, the lack of fairness has a very minor impact on real cases.

The implementation directly follows the model logic, as the mechanism for the detection and the resolution of inconsistent states is developed by comparing the user private information structure (in the cache) with the shared structures in the cloud (fig.8). The conflict resolution is model-driven, giving a priority to the personal scheduling (fig.9, on the top). In case of conflict between two shared activities (by two different users in the same group) that are targeting the same free slot (fig.9, down), the conflict resolution is solved according to a *last-write-wins* philosophy that, as mentioned, is not reflecting a fair approach in this case.

Future evolutions of the system will include advanced features for the management of the meta-states according to different philosophies and VOs classes (e.g. role-based).

4.2. Applications in e-Education

Team-work is a rather important aspect also in e-learning [22], where moderation [23], supervision and any other activities that explicitly require the synchronization of people are affected by problems of dynamic scheduling.

For instance, last trends include advanced features in e-Learning programs that move from the relatively simple personalization of processes [25] to their socialization [26]. Indeed, dynamic groups [33] can be established for people with similar needs, lacks or marks in order to push the co-operation among them in most advanced e-Education systems.

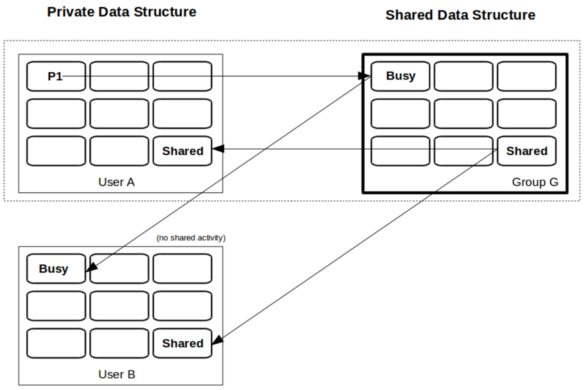


Figure 8. Maintaining consistent states by comparing private and shared data structures.

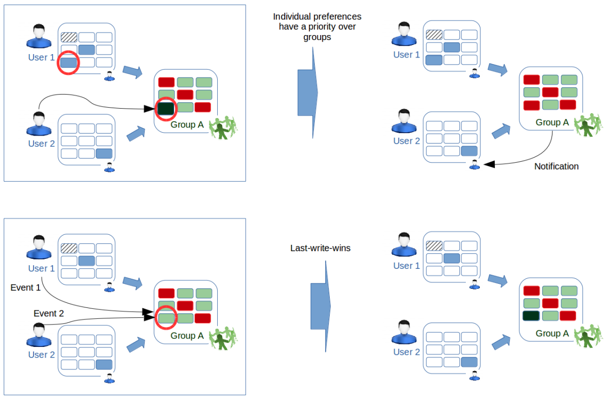


Figure 9. Conflicts resolution.

Prioritizing individual needs could be one of the key issues for the effectiveness and efficiency of the e-learning approaches involving groups of people.

Furthermore, the potential automaton of the activity scheduling in a privacy-friendly context would be useful and easy to integrate in many advanced systems (e.g. [29]) that are already addressing security and privacy issues [28].

5. Conclusions and Future Work

Optimistic Scheduling is designed from empirical observations assuming realistic conditions. It has been analysed considering unpredictable human behaviours inside virtual organizations and require a cooperative attitude to achieve high performances.

The flexibility of the model suggests a family of interesting tools for generic or specific purpose communities in a wide range of application domains.

The basic prototype is implemented according to a large scale philosophy and can be easily extended or integrated with further advanced features, as well as it can be particularized to address specific environments, requirements and purposes.

Acknowledgements

This research has been partially funded by the EU Project *SyncFree* (grant agreement number 609551).

References

[1] J. A. Wagner, “Studies of individualism-collectivism: Effects on cooperation in groups,” *Academy of Management journal*, vol. 38, no. 1, pp. 152–173, 1995.

[2] G. R. Jones and J. M. George, “The experience and evolution of trust: Implications for cooperation and teamwork,” *Academy of management review*, vol. 23, no. 3, pp. 531–546, 1998.

[3] M. Raman, “Wiki technology as a" free" collaborative tool within an organizational setting,” *Information systems management*, vol. 23, no. 4, pp. 59–66, 2006.

[4] “Kalendi,” `https://www.kalendi.com/`, accessed: 2015-06-05.

[5] “Upto,” `http://upto.com/`, accessed: 2015-06-05.

[6] “Teamweek,” `https://teamweek.com/`, accessed: 2015-06-05.

[7] “Doodle,” `http://doodle.com/`, accessed: 2015-06-05.

[8] R. Conte, N. Gilbert, G. Bonelli, C. Cioffi-Revilla, G. Deffuant, J. Kertesz, V. Loreto, S. Moat, J.-P. Nadal, A. Sanchez *et al.*, “Manifesto of computational social science,” *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 325–346, 2012.

[9] B. Schmidt, *The modelling of human behaviour*. Society for Computer Simulation International, 2000.

[10] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, “Conflict-free replicated data types,” in *Stabilization, Safety, and Security of Distributed Systems*. Springer, 2011, pp. 386–400.

[11] J. Pokorny, “Nosql databases: a step to database scalability in web environment,” *International Journal of Web Information Systems*, vol. 9, no. 1, pp. 69–82, 2013.

[12] S. Burckhardt, A. Gotsman, H. Yang, and M. Zawirski, “Replicated data types: specification, verification, optimality,” in *ACM SIGPLAN Notices*, vol. 49, no. 1. ACM, 2014, pp. 271–284.

[13] D. J. Abadi, “Consistency tradeoffs in modern distributed database system design: Cap is only part of the story,” *Computer*, no. 2, pp. 37–42, 2012.

[14] A. Bieniusa, M. Zawirski, N. Preguiça, M. Shapiro, C. Baquero, V. Balegas, and S. Duarte, “Brief announcement: Semantics of eventually consistent replicated sets,” in *Distributed Computing*. Springer, 2012, pp. 441–442.

[15] W. Vogels, “Eventually consistent,” *Communications of the ACM*, vol. 52, no. 1, pp. 40–44, 2009.

[16] S.F. Pileggi, “A privacy-friendly model for an efficient and effective activity scheduling inside dynamic virtual organizations,” in *11th EAI International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2015)*. Wuhan, China, 10-11 November, 2015.

[17] H.T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches,” *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[18] S.F. Pileggi, C. Fernandez-Llatas, and V. Traver, “When the social meets the semantic: Social semantic Web or Web 2.5,” *Future Internet*, vol. 4, no. 3, pp. 852–864, 2012.

[19] A.T. Bates, “*Technology, e-learning and distance education*,” Routledge, 2005.

[20] M. Michael at al., “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, 50–58, 2010.

[21] M. Rabinovich, Z. Xiao, and A. Aggarwal, “Computing on the edge: A platform for replicating internet applications,” *Web content caching and distribution*, pg.57–77, Springer, 2004.

[22] E. Williams, R. Duray, and V. Reddy, “Teamwork orientation, group cohesiveness, and student learning: A study of the use of teams in online distance education,” *Journal of Management Education*, vo. 30, no. 4, 592–616, 2006.

[23] G. Salmon, “*E-moderating: The key to teaching and learning online*,” Psychology Press, 2004.

[24] W. Vogels, “Eventually consistent, ” *Communications of the ACM*, vo. 52, no. 1, 40–44, 2009.

[25] M.K. Khrib, M. Jemn, and O. Nasraoui, “Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval, ” *Eighth IEEE International Conference on Advanced Learning Technologies*, 2008.

[26] J. Dron, and T. Anderson, “Collectives, networks and groups in social software for e-Learning, "*World conference on e-learning in corporate, government, healthcare, and higher education*, vo. 2007, no. 1, 2460–2467, 2007.

[27] S. Guri-Rosenblit, “Distance Education' and 'E-learning: Not the same thing, "*Higher education*, vo. 49, no. 4, 467–493, 2005.

[28] B. Jerman-Blažič, and T. Klobučar, “Privacy provision in e-learning standardized systems: status and improvements, " *Computer standards & interfaces*, vo. 27, no. 6, 561–578, 2005.

[29] B. Dong, Q. Zheng, J. Yang, H. Li, and M. Qiao, “An e-learning ecosystem based on cloud computing infrastructure, "*Ninth IEEE International Conference on Advanced Learning Technologies*, 2009.

[30] L. Camarinha-Matos, and H. Afsarmanesh, “*Brief historical perspective for virtual organizations*, "Springer, 2005.

[31] D. Ancona, and D. Caldwell, “Bridging the boundary: External activity and performance in organizational teams, "*Administrative science quarterly*, 634–665, 1992.

[32] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment, " *IEEE Journal on Selected Areas in Communications*, vo. 22, no. 1, 41–53, 2004.

[33] K. Franceschi, R. Lee, S. Zanakis, and D. Hinds, “Engaging group e-learning in virtual worlds, "*Journal of Management Information Systems*, vo. 26, no. 1, 73–100, 2009.