

# Enterprise 2.0: Research Challenges and Opportunities

Zakaria Maamar<sup>1</sup>(✉), Noura Faci<sup>2</sup>, Ejub Kajan<sup>3</sup>, Khoulood Boukadi<sup>4</sup>, Sherif Sakr<sup>5</sup>, Mohamed Boukhebouze<sup>6</sup>, Soraya Kouadri Mostéfaoui<sup>7</sup>, Vanilson Burégio<sup>8</sup>, Fadwa Yahya<sup>4</sup>, Valérie Monfort<sup>9</sup>, and Romain Hennion<sup>10</sup>

<sup>1</sup> Zayed University, Dubai, United Arab Emirates

[zakaria.maamar@zu.ac.ae](mailto:zakaria.maamar@zu.ac.ae)

<sup>2</sup> Université Lyon 1, Lyon, France

<sup>3</sup> State University of Novi Pazar, Novi Pazar, Serbia

<sup>4</sup> University of Sfax, Sfax, Tunisia

<sup>5</sup> University of New South Wales, Sydney, Australia

<sup>6</sup> CETIC, Charleroi, Belgium

<sup>7</sup> The Open University, Milton Keynes, UK

<sup>8</sup> Federal Rural University of Pernambuco, Recife, Brazil

<sup>9</sup> University of Paris1 Panthéon Sorbonne, Paris, France

<sup>10</sup> École Centrale de Paris and Global Knowledge, Paris, France

**Abstract.** Blending Web 2.0 technologies with enterprise information systems is setting up the stage for a new generation of information systems that will help enterprises open up new communication channels with their stakeholders. Contrary to traditional enterprises with top-down command flow and bottom-up feedback flow, the same flows in Enterprise 2.0 cross all levels and in all directions bringing people together in the development of creative and innovative ideas. The power of Web 2.0 technologies stems from their ability to capture real-world phenomena such as collaboration, competition, and partnership that can be converted into useful and structured information sources from which enterprises can draw information about markets' trends, consumers' habits, suppliers' strategies, etc. This paper discusses the research efforts that our international research group has put into the topic of Enterprise 2.0 (*aka* Social Enterprise). In particular, our research group advocates that existing practices for managing enterprise information systems need to be re-visited in a way that permits to capture social relations that arise inside and outside the enterprise, to establish guidelines and techniques to assist IT practitioners integrate social relations into their design, development, and maintenance efforts of these information systems, and last but not least to identify and tackle challenges that prevent capturing social relations.

**Keywords:** Business process · Social design · Social coordination · Social monitoring

---

This paper is based on the 1<sup>st</sup> author's invited talk at the 10<sup>th</sup> International Conference on Web Information Systems and Technologies (WEBIST'2014), April 3-5. Barcelona, Spain.

# 1 Introduction

With the recent boom in Web 2.0 technologies (e.g., JSON and AJAX) and applications (e.g., Facebook<sup>1</sup> and Twitter), there is a major trend in blending social computing with other forms of computing for instance, service computing [1,2] and mobile computing [3,4]. In this paper, we look into the blending of Web 2.0 concepts and technologies with enterprise computing that results into the development of **Social Business Processes (SBP)s** [5,6]. In particular, SBPs capitalize on Web 2.0 technologies and applications to ensure that the social dimension of the enterprise (*aka* Enterprise 2.0 [7,8]) is not overlooked during BP design and enactment. This dimension sheds the light on the informal networks that co-exist perfectly with formal networks where relations like supervision and substitution occur [9]. Different studies have shown the value-added of weaving social computing into enterprise operations in terms of demystifying who does what, profiling customers, and even re-engineering processes [10,11]. Moreover, many enterprises recognize the need of rethinking their strategies and reevaluating their operating models, as the world is getting more “social” [12].

Despite the growing interest in enterprise 2.0 in general and SBPs in particular, several limitations continue to hinder this interest. For instance, how to equip BP engineers with the necessary methods and techniques that will assist them capture the requirements of SBPs. Enterprises are still unsure about the return-on-investment of Web 2.0 technologies [13]. A recent study by Gartner reveals that “...many large companies are embracing internal social networks, but for the most part they’re not getting much from them” [14]. Over the last 2–3 years, our international research group has put efforts into enterprise 2.0 topic from different perspectives with focus on social design, social coordination, and social monitoring in this paper. Some of these efforts’ results are reported in [6,15]. Contrary to traditional enterprises with top-down command flow and bottom-up feedback flow, the same flows in Enterprise 2.0 cross all levels and in all directions bringing people together in the development of creative and innovative ideas. In principle, the power of Web 2.0 technologies stems from their ability to capture real-world phenomena such as collaboration, competition, and partnership that can be converted into useful and structured information sources from which enterprises can draw information about markets’ trends, consumers’ habits, suppliers’ strategies, etc.

The remainder of this paper is organized as follows. Section 2 briefly describes a case study to be used for illustration purposes. Section 3 provides an overview of three of our initiatives that look into enterprise 2.0 from the following perspectives: social design, social coordination, and social monitoring of business processes. Finally, we conclude the paper in Sect. 4.

---

<sup>1</sup> “By the end of 2013, Facebook was being used by 1.23 billion users worldwide, adding 170 million in just one year”, [www.theguardian.com/technology/2014/feb/04/facebook-10-years-mark-zuckerberg](http://www.theguardian.com/technology/2014/feb/04/facebook-10-years-mark-zuckerberg).

## 2 Illustrative Case Study

Our case study refers to the electronic-patient-folder system at Anderson Hospital that handles approximately 6000 annual inpatient admissions<sup>2</sup>. We leverage this system to identify first, some business processes' components (tasks, persons and machines) and second, the execution nature of some tasks. When a patient shows up at the hospital, the necessary documentation is scanned into a system known as ImageNow. Upon completion the scanned documentation is linked to the patient's MEDITECH record. An advantage of this linkage is that different stakeholders like billing staff, coders, and other authorized people have immediate, electronic access to the necessary information instead of waiting for the paper documentation to arrive. Prior to implementing the new system Anderson Hospital faced different challenges such as paper records limit access to one user at a time and paper and manual processes hamper compliance with some healthcare standards.

## 3 Challenges and Opportunities

Our international research group has launched several initiatives to tackle the challenges that enterprise 2.0 faces and tap into the opportunities that enterprise 2.0 offers. In the following, we discuss three of these initiatives that look into enterprise 2.0 from the following perspectives: social design, social coordination, and social monitoring of business processes.

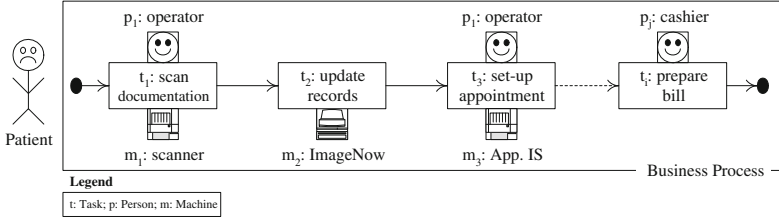
### 3.1 Social-Design Research Initiative

According to Faci et al. [16], the lack of design approaches for modeling SBPs is not helping enterprises capitalize on Web 2.0 applications' capabilities such as reaching out to more customers, collecting customers' online posts, and profiling customers. A recent study of 1,160 businesses and IT professionals reveals that *"only 22 percent of organizations believed that managers are prepared to incorporate social tools and approaches into their processes. Moreover, two-thirds of respondents said they were not sure they sufficiently understood the impact these technologies would have on their organizations over the next three years"* [17].

Our initiative on BP social design sheds the light on the three components of a process as well as the relations that connect these components, i.e., task (t), person (p), and machine (m) [6]. A task is a work unit that constitutes with other tasks a business process and that a person and/or machine execute. Task execution is either manual, automatic, or mixed. Figure 1 illustrates a simple healthcare-driven BP along with some components for instance,  $t_1$ : scan documentation,  $m_2$ : ImageNow system, and  $p_j$ : cashier. To execute tasks and hence complete processes, resources are required (Sect. 3.2).

In addition to the traditional execution relation that connects persons/machines to tasks, we establish social relations between tasks, between persons, and between machines. The following are examples of social relations [6]:

<sup>2</sup> [www.perceptivesoftware.com/pdfs/casestudies/psi\\_cs\\_anderson.pdf](http://www.perceptivesoftware.com/pdfs/casestudies/psi_cs_anderson.pdf).



**Fig. 1.** Example of a business process's components.

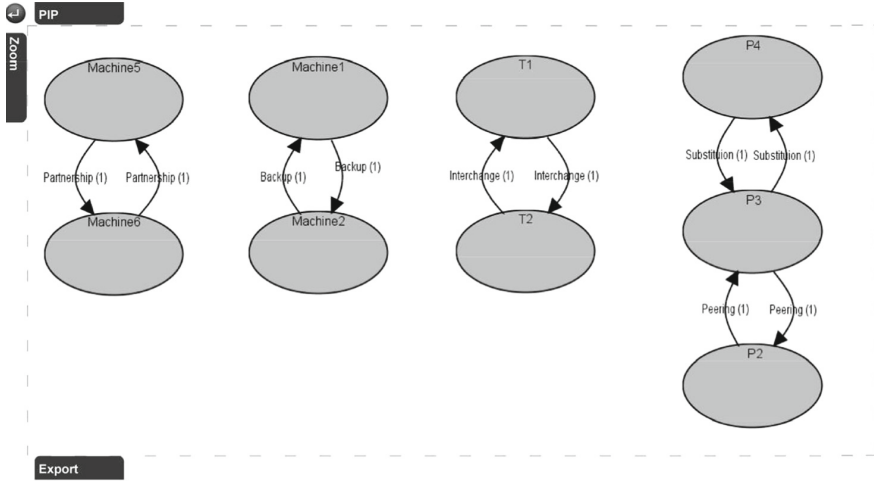
- **Interchange:**  $t_i$  and  $t_j$  engage in an interchange relation when both produce similar output with respect to similar input submitted for processing and their requirements do not overlap (e.g.,  $t_1$ : scan documentation and  $t'_1$ : enter patient details manually). The non-overlap condition avoids blockage when  $t_i$ 's requirements (e.g., online data entry) cannot be met due to absence of executors and thus,  $t_i$  needs to be interchanged with  $t_j$  that has different requirements (e.g., manual data entry).
- **Backup:**  $m_i$  (e.g., scanner in main reception) and  $m_j$  (e.g., three-function printer in nurse station) engage in a backup relation when both have similar capacities.
- **Delegation:**  $p_i$  and  $p_j$  engage in a delegation relation when both are already engaged in a **substitution** relation based on their respective capacities and  $p_i$  decides to assign a task that she will execute or is now executing to  $p_j$  due to unexpected changes in her status, e.g., call-in-sick or risk of overload (e.g., general practitioner transferring patient to emergency physician due to case severity).

Table 2 summarizes the social relations between tasks, between machines, and between persons along with their respective pre-conditions, conditions, and post-conditions. Pre-condition defines the rationale of establishing a social relation between a process's components. Condition indicates when a network built upon a social relation is used so that solutions to conflicts that prevent a business process completion are addressed (Sect. 3.2). Finally post-condition indicates when to stop using a network.

Figure 2 depicts examples of networks related to the case study that are generated at run-time. For example, a network of machines is built to specify the backup relation between  $m_1$  (**scanner**) and  $m_2$  (**three-function-printer**) since both machines have similar capacities. A configuration network of tasks is also constructed to express the interchange relation between  $t_1$  (**scan-documentation**) and  $t_2$  (**enter-patient-details-manually**). Both tasks produce similar output and their requirements do not overlap. Last and not least, a social network of persons is built to describe the peering relation between  $p_2$  (**cashier**) and  $p_3$  (**financial-manager**) since both persons have complementary capacities that are necessary to achieve  $t_5$  (**prepare-bill**). This social network expresses also a substitution relation between  $p_3$  (**general-practitioner**) and  $p_4$  (**emergency-physician**) since both have the same capacity.

**Table 1.** Summary of social relations.

Between	Types	Pre-conditions	Conditions	Post-conditions
$t_i, t_j$	Coupling	$t_i$ and $t_i$ participated in joint business processes	Review of business process design or concern over coupling level	Business process design completion or coupling level satisfaction
	Interchange	$t_i$ and $t_j$ producing similar output in receipt of similar input	$t_i$ lacking of executor who satisfies its requirements	Executor found for $t_j$
$m_i, m_j$	Backup	$m_i$ and $m_j$ having similar capacities	$m_i$ unexpected failure or concern over $m_i$ reliability	Backup/replacement machine found for $m_i$
	Cooperation	$m_i$ and $m_j$ having similar capacities	Concern over machine collective performance	collective performance level satisfaction
	Partnership	$m_i$ and $m_j$ having complementary capacities	Concern over machine collective performance	Collective performance level satisfaction
$p_i, p_j$	Substitution	$p_i$ and $p_j$ having similar capacities	$p_i$ expected unavailability (e.g., annual leave and sick leave) or concern over $p_i$ availability	Substitute found for $p_i$
	Delegation	$p_i$ and $p_j$ having similar capacities	$p_i$ unexpected unavailability (e.g., call-in-sick, urgent tasks to complete, and risk of overload)	Delegate found for $p_i$
	Peering	$p_i$ and $p_j$ having similar or complementary capacities	Concern over peering appropriateness	Peer found for either $p_i$ or $p_j$



**Fig. 2.** Screenshot of the BP execution social analysis component. A demo video is available at <https://www.youtube.com/watch?v=Py5oGPQot64>.

### 3.2 Social-Coordination Research Initiative

According to Faci et al. [16], agility of today's enterprises should not be confined to the organizational borders of the enterprise. Other vital aspects of the enterprise need to be taken into account such as re-engineering business processes, revisiting the practices of those executing these processes, and redefining the nature of resources that are made available for these processes at run-time. Since resources (e.g., data, power, and CPU time) do not sometimes last forever and are not unlimited and/or shareable, tasks and persons/machines need to coordinate the consumption and use of these resources so that conflicts are avoided and addressed, if they occur. Besides regular conflicts in terms of data and policy incompatibilities between enterprise systems, additional conflicts exist due to time constraints and/or simultaneous access to limited and/or non-shareable resources. Coordination is the best way to address these conflicts.

Our initiative on BP social coordination includes four steps [15]: categorize resources that tasks require for their execution, define how tasks/machines/persons of a BP bind to resources in order to achieve this execution, categorize conflicts on resources that arise between tasks, between machines, and between persons, and finally analyze the appropriateness of certain networks of tasks/persons/machines for addressing these conflicts. These networks are established during BP social design as per Sect. 3.1.

We categorize resources into (i) *logical*, i.e., their use/consumption does not lead into a decrease in their reliability/availability level and (ii) *physical*, i.e., their use/consumption does lead into a decrease in their reliability/availability level. This decrease requires resource replacement<sup>3</sup>/replenishment at a certain stage. We also define a set of properties that permit to describe a resource. These

<sup>3</sup> Replacement can be the result of degradation.

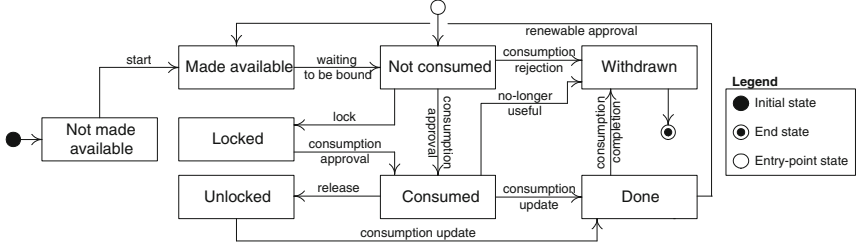
**Table 2.** Resource categories and examples.

Resource		Tasks	Examples of resources
Category	Property		
Logical	Unlimited (ul)	Put together medical team for surgery	Patient medical record (read mode)
	Limited (l)	Prepare on-call shifts	Doctors' weekly schedules (valid for one week only)
	Limited but renewable (lr)	Prepare interns' access rights to labs	File of interns (internship possible extension)
	Shareable (s)	Prepare patient for surgery	Patient lab results
	Non-shareable (ns)	Report on surgery outcome	Patient medical record (update mode)
Physical	Unlimited (ul)	Not applicable	Not applicable
	Limited (l)	Prepare patient for surgery	Anesthetic injection
	Limited but renewable (lr)	Carry out surgery on patient	Surgical staple cartridge
	Shareable (s)	Carry out surgery on patient	Oxygen tank
	Non-shareable (ns)	Check patient's vitals	Blood pressure tensiometer

properties are *limited* (l - when a resource use/consumption is measured or a resource ceases to exist either in compliance with its use/consumption cycle - to be introduced later - or because of constraints like temporal), *limited but renewable* (lr - when a resource use/consumption either hits a certain threshold or is subject to constraints like temporal; in either case renewal is possible), and *non-shareable* (ns - when a resource simultaneous use/consumption has to be scheduled). Unless stated a resource is by default unlimited (ul) and/or shareable (s). Table 2 suggests some examples of resources per category and property. To identify these examples we rely on the business process of Fig. 1.

Figure 3 shows a state transition diagram of a resource. The diagram is developed independently of whether the resource is logical or physical, takes into account the properties of Table 1, and permits to establish a resource's consumption cycle (cc) per property. On the one hand, the states ( $s_i$ ) of a resource include **not made available** (neither created nor produced yet), **made available** (either created or produced), **not consumed** (waiting to be bound by a task), **locked** (reserved for a task in preparation for consumption), **unlocked** (released by a task after consumption), **consumed** (bound by a task for ongoing performance), **withdrawn** (ceased to exist after unbounding all tasks if necessary), and

done (updated as per relevant property if necessary, and unbound by task). It should be noted that other states in a consumption cycle's diagram can be adopted if different properties are used without deviating from the aforementioned purpose of using this diagram. On the other hand, the transitions ( $trans_j$ ) connecting a resource's states together include start, waiting to be bound, consumption approval, consumption update, lock, release, consumption rejection, consumption completion, renewable approval, and no-longer useful. Upon a transition satisfaction, a resource takes on a new state.



**Fig. 3.** State transition diagram of a resource.

We list below some sequences of states ( $s$ ) and transitions ( $trans$ ) that represent a resource's consumption cycle per property denoted as  $r(cc_{property}) = s_i \xrightarrow{trans_i} s_{i+1} \xrightarrow{trans_{i+1}} s_{i+2} \dots s_{j-1} \xrightarrow{trans_{j-1}} s_j$  (subscripts in state and transition names are given for notational purposes, only).

- $r(cc_{ul})$ : not made available  $\xrightarrow{start}$  made available  $\xrightarrow{waiting\ to\ be\ bound}$  not consumed  $\xrightarrow{consumption\ approval}$  consumed  $\xrightarrow{no-longer\ useful}$  withdrawn. The resource (e.g., patient medical record) remains available for additional consumption until the transition from consumed to withdrawn is satisfied (e.g., patient discharge).
- $r(cc_{l1})$ : not made available  $\xrightarrow{start}$  made available  $\xrightarrow{waiting\ to\ be\ bound}$  not consumed  $\xrightarrow{consumption\ approval}$  consumed  $\xrightarrow{consumption\ update}$  done  $\xrightarrow{consumption\ completion}$  withdrawn. The transition from done to withdrawn then end-of-state shields a resource (e.g., patient list with contagious diseases) from any new or additional tentative of consumption by tasks (e.g., submit patient list to healthcare authorities upon disease detection) after completing a consumption cycle. In done state a resource's parameters (e.g., accuracy level such as patient list obsolete) are updated and the resource is detached (or unbound) from tasks.
- $r(cc_r)$ : not made available  $\xrightarrow{start}$  made available  $\xrightarrow{waiting\ to\ be\ bound}$  not consumed  $\xrightarrow{consumption\ approval}$  consumed  $\xrightarrow{consumption\ update}$  done  $\xrightarrow{renewable\ approval}$  made available. The transition from done to made available permits to regenerate a resource (e.g., file of interns) for another round of consumption (e.g., internship extension).



Based on resource categories and their properties and how tasks/persons/machines bind to resources, we examine conflicts over resources between tasks, only for illustration purposes. The following notation is adopted:  $t_i/m_i/p_i \rightarrow r_i$  means that a resource is made available for a task/machine/person;  $r_{i,j}$  means that a resource produced by a certain task (resp. machine/person) is transferred to another similar (resp. similar/different) peer for the needs of consumption (resp. use); and,  $r_{i,\{j,k,\dots\}}$  generalizes  $r_{i,j}$  but this time the resource is shared between peer $_{\{j,k,\dots\}}$ .  $\mathcal{T}$ -Conflict<sub>1</sub> is an example of conflicts between tasks with emphasis on resource consumption and not data (inputs and outputs) and policy incompatibilities between these tasks.  $\mathcal{T}$ -Conflict<sub>1</sub> arises when (i) a prerequisite relation between  $t_i$  and  $t_j$  exists, (ii)  $\text{consume}(t_i, r_i) \rightarrow \text{produce}(t_i, r_{i,j})$ , and (iii)  $t_j$  needs  $r_{i,j}$  (i.e.,  $t_j \nrightarrow r_j$ , no  $r_j$  is made available for  $t_j$ ). Potential conflicts on  $r_{i,j}$  (and eventually  $r_{\{i,k,\dots\},j}$  and  $r_{i,\{j,k,\dots\}}$ ) because of the *limited* property of  $r_{i,j}$ , include:

- l: two cases result out of the prerequisite relation between  $t_{\{k,\dots\}}$  (e.g., complete necessary paperwork) and  $t_j$  (e.g., direct patient to appropriate department) on top of the same relation between  $t_i$  (e.g., check patient vitals) and  $t_j$ :
  - (a)  $r_{i,j}$  (e.g., report on vital levels) ceases to exist (e.g., blood sample no longer valid) before the execution of  $t_j$  begins;  $t_j$  waits for  $t_{\{k,\dots\}}$  to produce  $r_{\{k,\dots\},j}$  (e.g., insurance provider approval); (at least one)  $t_{\{k,\dots\}}$  either is still under execution (e.g., due to delay in receiving approval from insurance provider) or failed.
  - (b) Only one consumption cycle of  $r_{i,j}$  is permitted (per type of property) but it turns out that several consumption cycles of  $r_{i,j}$  are required to complete the execution of  $t_j$  and finish the consumption of  $r_{\{k,\dots\},j}$  that  $t_{\{k,\dots\}}$  produce.

After identifying the different task conflicts on resources, we suggest solutions for these conflicts based on the networks that are established in Sect. 3.1. These solutions consider the fact that tasks are associated with transactional properties (e.g., pivot, retrievable, and compensatable) that limit their re-execution in the case of failure [18]. The following examines briefly how the interchange and coupling networks of tasks are used to address  $\mathcal{T}$ -Conflict<sub>1</sub>-Case a.

- (a)  $r_{i,j}$  ceases to exist before the execution of  $t_j$  begins;  $t_j$  waits for  $t_{\{k,\dots\}}$  to produce  $r_{\{k,\dots\},j}$ ; at least one  $t_k$  either is still under execution or failed. Current statuses of tasks and resources are:  $\text{state}(t_i)$ : done;  $\text{state}(r_{i,j})$ : withdrawn;  $\text{state}(t_j)$ : not-activated; and  $\text{state}(t_k)$ : either activated (still under execution) or failed with focus on the latter state below. Because  $t_i$  now takes on done state, pivot (canceling  $t_i$ ) and retrievable (re-executing  $t_i$ ) transactional properties are excluded from the analysis of developing solutions to address resource conflicts. This analysis is given in Table 3. The objective is to re-produce  $r_{i,j}$  (or produce  $r_{i',j}$  with  $t_{i'}$  being obtained through the interchange network of  $t_i$ ). Because of  $t_k$  failure,  $r_{k',j}$  is produced using  $t_{k'}$  that is obtained through the interchange network of  $t_k$ .

**Table 3.** Possible coordination actions for  $\mathcal{T}$ -Conflict<sub>1</sub>/limited property/case>a.

Transactional property		Coordination actions	Network involved
$t_i$	$t_k$		
Null	Null	– re-execute $t_i$ to re-produce $r_{i,j}$	N/A
		– re-execute $t_k$ to produce $r_{k,j}$	
	Pivot	Deadlock	N/A
	Compensatable	Deadlock	N/A
	Retriable	– re-execute $t_i$ to re-produce $r_{i,j}$	Interchange( $t_k, t_{k'}$ )
		– replace $t_k$ with $t_{k'}$ then execute $t_{k'}$ to produce $r_{k',j}$	
Compensatable	Null	– compensate $t_i$ ; either re-execute $t_i$ to re-produce $r_{i,j}$ or replace $t_i$ with $t_{i'}$ then execute $t_{i'}$ to produce $r_{i',j}$	Interchange( $t_i, t_{i'}$ )
		– either re-execute $t_k$ to produce $r_{k,j}$ or replace $t_k$ with $t_{k'}$ then execute $t_{k'}$ to produce $r_{k',j}$	Interchange( $t_k, t_{k'}$ )
	Pivot	Deadlock	N/A
	Compensatable	– compensate $t_i$ ; either re-execute $t_i$ to re-produce $r_{i,j}$ or replace $t_i$ with $t_{i'}$ then execute $t_{i'}$ to produce $r_{i',j}$	Interchange( $t_i, t_{i'}$ )
		– replace $t_k$ with $t_{k'}$ then execute $t_{k'}$ to produce $r_{k',j}$	Interchange( $t_k, t_{k'}$ )
	Retriable	– compensate $t_i$ ; either re-execute $t_i$ to re-produce $r_{i,j}$ or replace $t_i$ with $t_{i'}$ then execute $t_{i'}$ to produce $r_{i',j}$	Interchange( $t_i, t_{i'}$ )
		– re-execute $t_k$ to produce $r_{k,j}$	

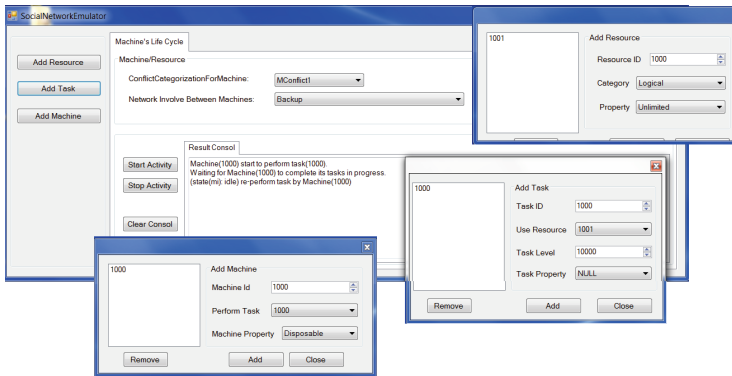
**Fig. 4.** System's screenshots.

Figure 4 illustrates a screenshot for an under-development proof-of-concept for a rule-based engine that focuss on implementing the strategies reported in Table 3.

### 3.3 Social-Monitoring Research Initiative

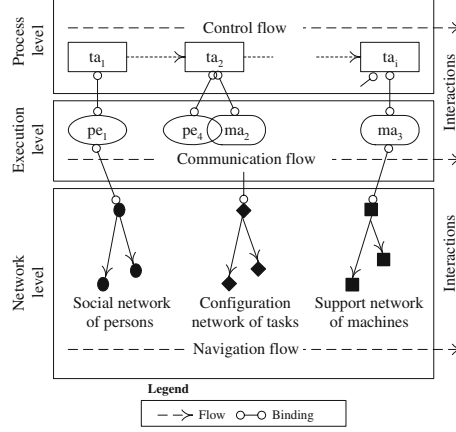
According to Faci et al. [16], monitoring seems to be the commonly-used technique for tracking the execution progress of BPs. Besides providing a real-time and end-to-end view of this progress, monitoring should also offer an organizational and social view over this progress in terms of who executes what, who delegates to whom, and who sends what, to whom, and when. Obstacles that hinder BP successful completion are multiple (e.g., lack of necessary machines that can execute tasks) and hence, will impact the enterprise effectiveness (e.g., delay in delivery) and efficiency (e.g., costly machine re-allocation). The difficulty of measuring intangible and ad-hoc exchanges between people when executing tasks represents a major barrier to social interaction pattern recognition like collaboration and delegation, as well as the role of these patterns in BP improvement. The way these exchanges should happen can be part of a social monitoring framework in which specialized flows connecting these messages are developed to detect anomalies.

Our initiative on BP social monitoring aims of ensuring the successful completion of BPs despite obstacles that could arise and hence, delay this completion. Absence of necessary machines to execute tasks is an example of obstacle. We develop specialized flows known as *control*, *communication*, and *navigation* and enact the development of these flows in conjunction with the completion of a BP. The control flow connects tasks together with respect to the BP's business logic. The communication flow connects the BP's executors (machines and/or persons) together when they execute tasks. Finally, the navigation flow connects networks of tasks, networks of machines, and networks of persons together when the BP completion runs into difficulties. These networks are established in the BP social design (Sect. 3.1). We recall that BP completion and hence, task execution requires resources that are sometimes *limited*, *not shareable*, and/or *not renewable* (Sect. 3.2).

BP social monitoring looks into the exchanges of messages that occur between tasks, between executors (persons/machines), and between networks during task execution and conflict resolution. These exchanges lead into developing flows that help identify who supports whom, who sent what, to whom, and when. Figure 5 illustrates the architecture of our flow-based approach for BP social monitoring. Three levels along with their respective flows constitute this architecture. The process level is linked to the control flow, the execution level is linked to the communication flow, and the network level is linked to the navigation flow.

In the following, we examine the interactions between the social, configuration, and/or support networks during a control-flow execution. These interactions establish navigation flows as per the nature of scenario that shapes the progress of this execution. We identify two scenarios: task replacement

(explained hereafter) and task delay and decompose the messages supporting network interactions into vertical ( $v$ , between the control flow and networks) and horizontal ( $h$ , between networks). We associate each network with an authority component ( $N_{auth}$ , [19]).



**Fig. 5.** Cross-flow interactions for BP social monitoring.

Assigning tasks (e.g., scan documentation and enter patient details manually) to potential executors (persons and/or machines) might turn out unsuccessful when the capacities of these executors do not satisfy the requirements of these tasks (Sect. 3.1). Using specific networks, an option consists of replacing tasks that cannot be executed with similar ones and then, looking for executors who have the capacities of satisfying the requirements of the replacement tasks. For illustration purposes, let us assume a control flow that consists of  $t_1$ ,  $t_2$ , and  $t_{i,i \neq 3}$  and that  $t_1$ 's requirements are not satisfied.

First  $t_1$  needs to be replaced with another similar task using the interchange network. The control flow sends this network's authority ( $N_{auth(interchange)}$ ) a  $v$ -message as shown below asking to find a replacement for  $t_1$ .  $t_1$  now acts as an entry node in the interchange network:

```
v-query(
  from      :  $Flow^c$ 
  to        :  $N_{auth(interchange)}/nt_1$ 
  subject   :  $findReplacement(t_1)$ )
```

After screening the edges that connect  $t_1$  to other tasks in the interchange network,  $t_3$  is selected using the highest weight among all these edges as a selection criterion, for example. Prior to confirming  $t_3$  as a replacement for  $t_1$  the  $N_{auth(interchange)}$  sends the  $N_{auth(coupling)}$  a  $h$ -message asking to check for the coupling level between  $t_3$  and those tasks, i.e.,  $t_2$ , that are dependent on  $t_1$  in the control flow. This message sending is shown below:

```

h-query(
  from      :  $N_{auth(interchange)}/nt_1$ 
  to        :  $N_{auth(coupling)}/nt_3$ 
  subject   : requestCouplingLevel( $t_3, t_2$ )

```

Upon receipt of the coupling level from the  $N_{auth(interchange)}$  through the  $N_{auth(coupling)}$  using the  $h$ - and  $v$ -messages shown below, the control flow compares this level to a threshold that the BP engineer has set up earlier. Assuming the comparison is valid,  $t_3$  is now confirmed as a replacement for  $t_1$  in the control flow with  $t_3$  being connected to  $t_2$ . Otherwise the search for another task continues.

```

h-reply(
  from      :  $N_{auth(coupling)}/nt_3$ 
  to        :  $N_{auth(interchange)}/nt_1$ 
  subject   : couplingLevel( $t_3, t_2$ )

```

```

v-reply(
  from      :  $N_{auth(interchange)}/nt_1$ 
  to        :  $Flow^c$ 
  subject   : couplingLevel( $t_3, t_2$ )

```

After updating the control flow, the next step is to identify executors for  $t_3$ . Assuming a successful match between  $t_3$ 's requirements and some executors' capacities two exclusive cases arise, which means more messages need to be exchanged. For illustration purposes we assume that executors correspond to persons ( $p_3$  for  $t_3$ ).

1. **Substitution:** if there is a concern over  $pe_3$ 's availability level, the control flow sends the substitution network's authority ( $N_{auth(substitution)}$ ) a  $v$ -message asking to find a substitute for  $p_3$  with respect to a certain availability threshold ( $T_{rel}$ ). This message along with its response are shown below with  $p'_3$  being the new executor for  $t_3$ .

```

v-query(
  from      :  $Flow^c$ 
  to        :  $N_{auth(substitution)}/np_3$ 
  subject   : findSubstitute( $p_3, T_{rel}$ )
v-reply(
  from      :  $N_{auth(substitution)}/np_3$ 
  to        :  $Flow^c$ 
  subject   : substitute( $p_3:p'_3$ )

```

2. **Delegation:** if  $p_3$  turns out unavailable unexpectedly or will become overloaded by receiving  $t_3$  for execution, the  $N_{auth(substitution)}$  sends the delegation network's authority ( $N_{auth(delegation)}$ ) a  $h$ -message asking for identifying a delegate, e.g.,  $p'_3$ , who could execute the task.

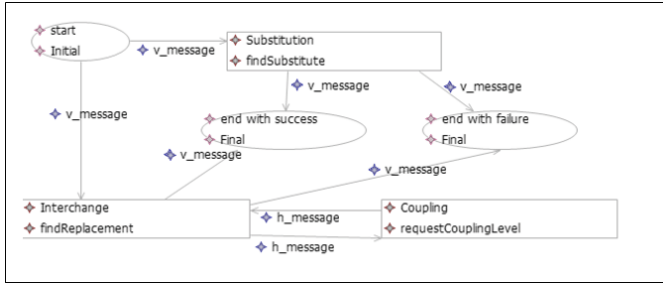


Fig. 6. Partial navigation flow for an example of business process.

The aforementioned horizontal messages constitute a navigation flow in which states and transitions correspond to nodes in networks and messages between nodes, respectively. Vertical messages trigger the establishment of navigation flows (Fig. 6).

## 4 Conclusion

In this paper we discussed the blend of Web 2.0 concepts and technologies with enterprise computing. This blend results into the development of **Social Business Processes (SBP)s**. A SBP exposes the social relations that exist between tasks, between machines, and last but not least between persons. These relations led into developing specialized networks that are used to solve conflicts over resources as well as monitoring the enterprise performance. Although we acknowledges that tasks and machines cannot “socialize” (in the strict sense) like persons do in real life, putting tasks together and machines together suggests a lot of similarities with how persons behave in their daily life. Supporting our claims that tasks and machines can to a certain extent socialize, Tan et al. state that “... *Currently, most social networks connect people or groups who expose similar interests or features. In the near future, we expect that such networks will connect other entities, such as software components, Web-based services, data resources, and workflows. More importantly, the interactions among people and nonhuman artifacts have significantly enhanced data scientists’ productivity*” [20].

## References

1. Maamar, Z., Faci, N., Boukadi, K., Sheng, Q.Z., Yao, L.: Commitments to regulate social web services operation. *IEEE Trans. Serv. Comput.* **7**(2), 154–167 (2014)
2. Maamar, Z., Hacid, H., Hunhs, M.N.: Why web services need social networks. *IEEE Internet Comput.* **15**(2), 90–94 (2011)
3. Baumer, E.P.S., Khovanskaya, V.D., Adams, P., Pollak, J.P., Volda, S., Gay, G.: Designing for engaging experiences in mobile social-health support systems. *IEEE Pervasive Comput.* **12**(3), 32–39 (2013)

4. Roussaki, I., Kalatzis, N., Liampotis, N., Kosmides, P., Anagnostou, M.E., Doolin, K., Jennings, E., Bouloudis, Y., Xynogalas, S.: Context-awareness in wireless and mobile computing revisited to embrace social networking. *IEEE Commun. Mag.* **50**, 74–81 (2012)
5. Erol, S., Granitzer, M., Happ, S., Jantunen, S., Jennings, B., Johannesson, P., Koschmider, A., Nurcan, S., Rossi, D., Schmidt, R.: Combining BPM and social software: contradiction or chance? *J. Softw. Maint. Evol. Res. Pract.* **22**, 449–476 (2010)
6. Kajan, E., Faci, N., Maamar, Z., Loo, A., Pljaskovic, A., Sheng, Q.Z.: The network-based business process. *IEEE Internet Comput.* **18**(2), 63–69 (2014)
7. Drupad, S.: Enterprise Web 2.0 Worth \$4.6 Billion in 2013 (Visited September 2008) Searchviews. [www.searchviews.com/index.php/archives/2008/04/enterprise-web-20-worth-46-billion-in-2013.php](http://www.searchviews.com/index.php/archives/2008/04/enterprise-web-20-worth-46-billion-in-2013.php). Accessed 21 April 2008
8. McAfee, A.P.: Enterprise 2.0: the dawn of emergent collaboration. *MIT Sloan Manage. Rev.* **47**(3), 21–28 (2006)
9. Grim-Yefsah, M., Rosenthal-Sabroux, C., Thion, V.: Using information of an informal network to evaluate business process robustness. In: *Proceedings of the International Conference on Knowledge Management and Information Sharing (KMIS 2011)*, Paris, France (2011)
10. Badr, Y., Maamar, Z.: Can enterprises capitalize on their social networks? *Cutter IT J.* **22**(10), 10–14 (2009)
11. Cross, R., Gray, P., Cunningham, S., Showers, M., Thomas, R.J.: The collaborative organization: how to make employee networks really work. *MIT Sloan Manage. Rev.* **51**, 83–90 (2010)
12. Chandler, S.: Social BPM: Gateway to Enhanced Process Efficiency. <http://www.virtusa.com/blog/2011/11/>, visited September 2012 (November 2011)
13. Seshadri, S.C.: Exploring business and collaboration with web 2.0. *SETLabs Briefings J.* **7**(3), 3–10 (2009)
14. Gartner: How to Analyze Your Sales Processes on Efficiency versus Effectiveness. Gartner report (2012)
15. Maamar, Z., Faci, N., Kouadri Mostéfaoui, S., Kajan, E.: Network-based conflict resolution in business processes. In: *Proceedings of the 10th IEEE International Conference on e-Business Engineering (ICEBE 2013)*, Coventry, UK (2013)
16. Faci, N., Maamar, Z., Kajan, E., Benslimane, D.: Research roadmap for the enterprise 2.0 - issues and solutions. *Scientific Publications of the State University of Novi Pazar Ser. A: Appl. Math. Inf. Mech.* **6**(2), 81–89 (2014)
17. Vizard, M.: IBM: business processes need to get social in 2013 (2012). <http://www.itbusinessedge.com/blogs/it-unmasked/ibm-business-processes-need-to-get-social-in-2013.html>
18. Little, M.: Transactions and web services. *Commun. ACM* **46**(10), 49–54 (2003)
19. Maamar, Z., Faci, N., Luck, M., Hachimi, S.: Specifying and implementing social web services operation using commitments. In: *The Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC 2012)*, Riva del Garda, Trento, Italy (2012)
20. Tan, W., Blake, M.B., Saleh, I., Dustdar, S.: Social-network-sourced big data analytics. *IEEE Internet Comput.* **17**(5), 62–69 (2013)

Web Information Systems and Technologies  
10th International Conference, WEBIST 2014,  
Barcelona, Spain, April 3-5, 2014, Revised Selected  
Papers  
Monfort, V.; Krempels, K.-H. (Eds.)  
2015, XV, 380 p. 113 illus. in color., Softcover  
ISBN: 978-3-319-27029-6