A Framework for Context-aware Heterogeneous Group Decision Making in Business Processes

Carmen De Maio[a], Giuseppe Fenza[b], Vincenzo Loia[b], Francesco Orciuoli[b], Enrique Herrera-Viedma[c,d]

[a] *Dipartimento di Ingegneria dell'Informazione, Elettrica e Matematica Applicata, University of Salerno, 84084 Fisciano (SA), Italy*
[b] *Dipartimento di Scienze Aziendali - Management & Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy*
[c] *Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain*
[d] *Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Abstract

In Business Process Management great attention is given to Computational Intelligence for supporting process life-cycle. Several approaches have been defined to support human decision making. The main drawback is that there are no solid criteria determining always optimal decisions since context, matter of discussion, and involved actors may differ at each execution. This work focuses on the definition of a framework to support and trace human decision making activities, in business processes, when heterogeneous decision-makers have to find a consensus to select most promising alternative to follow. The framework relies on Fuzzy Consensus Model and implements Reinforcement Learning algorithm to learn weight of the decision-makers through the analysis of past process executions considering context and performances of business processes. Context awareness relies on semantic web technologies enabling ontological reasoning to evaluate context similarity used to assign the right weight to the involved decision-makers also in the case when more general or more specific context occurs. The framework has been instantiated in the case study of Supply Chain Management. The

Email addresses: cdemaio@unisa.it (Carmen De Maio), gfenza@unisa.it (Giuseppe Fenza), loia@unisa.it (Vincenzo Loia), forciuoli@unisa.it (Francesco Orciuoli), viedma@decsai.ugr.es (Enrique Herrera-Viedma)

Preprint submitted to Elsevier

analysis of the simulation results reveal that the proposed weight learning algorithm and the considered initial weight association strategies (*Starting Weight* and *Training Executions*), even if the cold start, give to decision-makers the chance to fill the gap with respect to more experienced decision makers.

Keywords: Business Processes, Fuzzy Consensus Model , Group Decision Making, Context Awareness, Semantic Web, Reinforcement Learning

---

1. Introduction and Motivation

In complex business environments (e.g., organizations), Business Process Management (BPM) [1] provides an effective tool for managing processes. Business Processes (BP) include person-to-person work steps, system-to-system communications, or combinations of both. BPM integrates several disciplines like, for instance, process modeling, process simulation, process execution, process monitoring, etc. One of the most important concept underlying BPM is that process execution must be monitored in order to detect useful elements to improve next executions and providing value (e.g., for organization, customers, etc.). According to the above aim, a number of research works, focused on automatically or semi-automatically supporting human decision making activities, within a Business Process (BP), are recognizable in the specialized scientific literature. Many of them are based on the analysis of past process executions in order to derive decision criteria, transform them into rules and execute such rules to make decisions. This trend is motivated by the consideration that by means of existing machine learning algorithms it is possible to freeze past decisions (taken by humans) in form of rules and automatically apply them during new executions of the same BP without the intervention of humans. A drawback for the above mentioned approaches is that there are no criteria that determine always optimal decisions since context and matter of decision may differ from situation by situation. Thus, it is crucial to investigate the definition of tools supporting human decision making within BP, which are capable of taking into account the context in which processes run. Moreover, approaches that completely replace human decision makers with automatic rules are not always applicable and, consequently, such approaches cannot be generalized and scaled to different and heterogeneous situations.

According to the aforementioned considerations, there is a need for definition

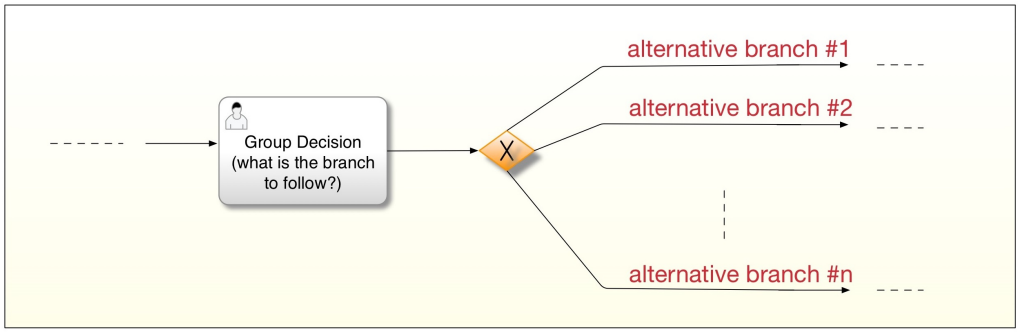

Figure 1: BP fragments describing group decision making activities.

of frameworks capable to support human decision making. In order to be effective, such frameworks must be:

- integrable smoothly into the practices of BPM in order to be portable across organizations which adopt standards and standards de-facto;
- based on a formal model to handle consensus achievement in the group of decision-makers in order to provide a trusted mechanism to moderate conflict resolution and, in general, decision making;
- context-aware in order to drive the consensus achievement process by also taking care of the impact of peculiar context features;
- adaptive with respect to the goodness of past decisions of the decision-makers in order to improve next processes.

These characteristics will foster *organizational learning* in the sense that the organization will learn from the level of success of the past decisions in order to improve new process executions by giving greater importance to those decision-makers who performed better in past processes executed in similar contexts.

Taking care of the above needs, this work focuses on the definition of a framework to support and trace human decision making activities, within BP, when more heterogeneous decision makers have to find a consensus to select one of a set of defined alternatives, as shown in Fig.1.

Specifically, the proposed framework is based on the existing *Fuzzy Consensus Model* and on a *Reinforcement Learning* algorithm. The first one is used to

find a convergence among a set of heterogeneous decision makers' opinions[1]. The second one is applied in order to learn the relative importance of the decision makers.

In literature, approaches assuming different weights for decision makers are recognizable [2]. For instance, authors of [3] introduce a trust-based approach to calculate weights.

This work proposes a novel approach to calculate such weights, by tracing the context of past experiences of people to make decisions. The relative importance of a decision maker is measured by considering the past successful decisions taken by him/her. The concept of relative importance is developed by considering the context in which such decisions take place. Contexts are modeled by means of semantic technologies enabling ontological reasoning and succeed in providing relative importance of decision-makers also in the case of more general or more specific contexts.

The manuscript is organized as follows: Section 2 provides an overview of the overall proposed approach; Section 3 defines fuzzy consensus model to support GDM and describes how it has been applied to face with heterogeneous issues related to the executions of BP in different contexts; Section 4 describes how the context has been modeled and used to assign heterogeneous weights to decision makers according to their previous decisions in similar contexts; and Section 5 describes how reinforcement learning algorithm has been used to update weights assigned to decision makers in the knowledge base according to the outcome of their decision. Section 6 provides the application of the proposed framework to a case study in the domain of Supply Chain Management. The results of such case study have been subsequently analyzed and discussed. Finally, conclusions and future directions close the paper.

2. Overall Approach

The proposed approach is mainly focused on injecting a semi-automatic method to achieve a consensus into business processes when a Group Decision Making (GDM) problem is proposed. Specifically, the main idea in this work is to adopt the Fuzzy Consensus Model proposed in [4],[5] to face with GDM problem during business process execution. From the architectural viewpoint, Fig.2 points out that the proposed Context-aware Group Decision

[1]The convergence happens when the degree of consensus is higher than a fixed threshold, otherwise the decision will not be taken and other rounds take place.

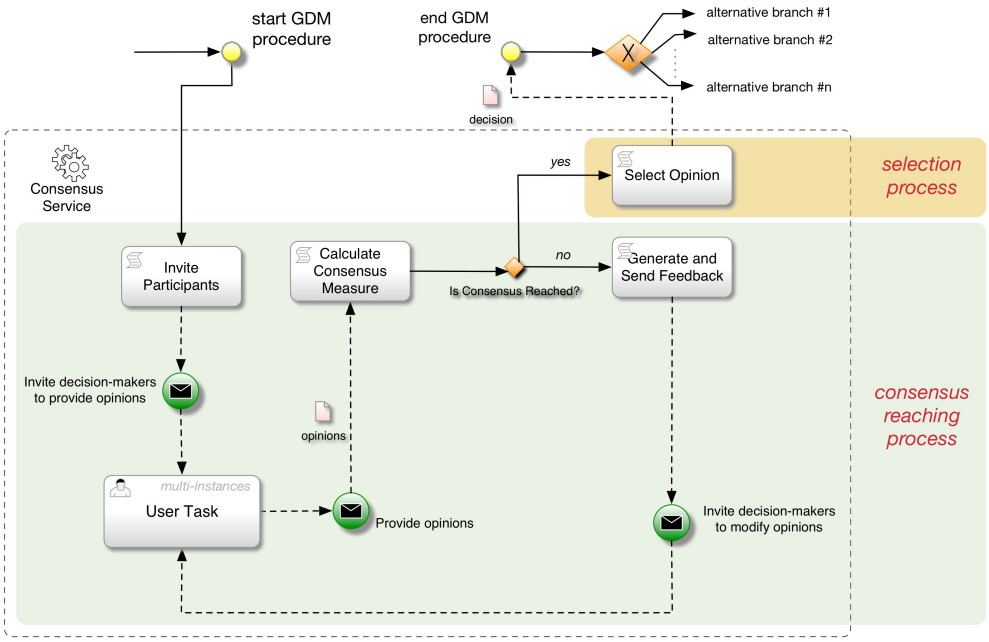

Figure 2: Group Decision Making pattern

Making has been modelled as a business process. Specifically, it consists of some phases/services that are orchestrated as shown in Fig.2 by means of Business Process Model and Notation (BPMN). BPMN is a standard for business process modeling that provides a graphical notation for specifying business processes. So, the idea is aimed at demonstrating that the proposed framework could be easily integrated into the enterprises that already adopt standard systems, languages and practices for business process management for supporting the human decision making activities.

Thus, as shown in Fig.2, it is suitable to model this kind of method by means of BPMN [6] in order to use it as a pattern when needed. In particular, when a GDM problem is proposed and represented, for instance, by means of the pattern provided in Fig.1 it is possible to expand, at the BPMN level during the business process designing phase, the user task of Fig.1 with the sub-process reported in Fig.2. In particular, the aim of such sub-process is to formalize the GDM process and make it executable by means of a set of scripts implementing a structured approach we are going to explain. This approach consists of two different phases called consensus reaching

and selection, which can be generally applied to several and heterogeneous domains. Although it is specific, the pattern of Fig.1 is always possible to reconduce different gateway-based pattern to it in order to apply the approach we are going to propose.

A *consensus reaching process* in a GDM problem is an iterative process composed by several discussion rounds in which involved decision makers receive feedback that induce them to modify their preferences. The feedback is given by a moderator that is in charge to supervise and drive the consensus process to achieve the maximum possible agreement reducing the number of experts outside of the consensus at each iteration. Once the maximum possible agreement has been achieved it is possible to start the *selection process* that is committed to extract a decision from the proposal decisions generated by the group members. Initially, GDM problems were defined in situations where all the decision makers' opinions were considered equally important [7][8]. Otherwise, it is more realistic to assume that the consensus model suitable in the organization requires a heterogeneous model that assigns relative importance to decision makers according to some features. Recently, GDM problems has been defined for heterogeneous decision makers. In [9], the authors identify three kinds of heterogeneity: different preference representation formats; different expression domains (e.g., numeric ones, linguistic ones, etc) [10]; and different backgrounds and levels of knowledge about the problem for each decision maker.

The proposed approach, namely *Context-aware Heterogeneous Group Decision Making* (CaHGDM), introduces a further level of heterogeneity that is related to the different contexts in which the decision activity could take. Specifically, this work focuses on two main heterogeneities: decision maker background (including: history of decision maker in the organization, background, levels of knowledge of the problem, and so on); and context that attains with matter of discussion and organizational environment. Additionally, we assume fuzzy preference relation [11] as the base to uniform the information [12], [13]. Specifically, during the consensus calculus we transform experts' preferences represented in different ways (e.g., preference ordering of the alternatives, utility function) in fuzzy preference relations [14].

More formally, given a set of decision makers $DM = \{dm_1, dm_2, \ldots, dm_m\}$, $(m \geq 2)$

and a set of alternative branches $AB = \{ab_1, ab_2, \ldots, ab_n\}$, $(n \geq 2)$, let us consider the execution of BP which matter of discussion is in a specific current context c_c, then the proposed framework during the consensus process

execution will assign to each involved decision maker $dm_k \in DM$ a relative importance degree that depends on all of the following factors:

- the history of past choices that decision maker taken in the previous participation;
- the context in which decision maker has participated considering greatest matching degree among the current context c_x and the previous ones;

By adopting a continual learning it is possible to improve the weight estimations. Specifically, this work stresses the importance of context-aware heterogeneous fuzzy consensus model learning from the past executions. So, there is a need to feed and maintain a knowledge base storing the associations between contextual variables, decisions, and weights for each decision maker, that is the list of the following 4-tuples $\langle dm_i, c_j, \overline{ab}_{dm_i,c_j}, w_{dm_i,c_j} \rangle$ where:

- $dm_i \in DM$ is the i-th decision maker;
- c_j is the context where decision maker has already participated;
- $\overline{ab}_{dm_i,c_j}$ is the vector of opinions of decision maker dm_i in the process execution context c_j with respect to the available alternative branches (at first iteration of consensus process execution), it is composed of fuzzy preference values $ab^k_{dm_i,c_j} \in [0,1]$ representing the value assigned to k-th alternative branch ab_k ($\forall k \in \{1, 2, \ldots, m\}$);
- w_{dm_i,c_j} is the weight associated to decision maker dm_i in the context c_j.

In order to be context-sensitive, the proposed framework integrates two architectural modules aimed to learn and retrieve the suitable weight for each decision maker during consensus process execution:

- *Context-Aware Weighting.* Contexts and background knowledge of the decision makers are supposed to be modeled with semantic technologies in order to enable context reasoning and to assign relative importance to decision makers also in the case of more general or more specific context occurs. Based on the semantic modeling, weights of decision makers are retrieved in a context-aware manner considering context most similar to the current one. This will be detailed in Section 4.

- *Decision Makers Weights Learning.* The overall workflow implements an algorithm of Reinforcement Learning (RL) that taking into account 4-tuples and the BP outcomes adopts a rewarding or punishing policies aimed to train the system in order to better assign and distribute the weights among decision makers when the process execution runs into the same (or similar) context. More details about RL approach adoption will be provided in Section 5.

In order to define and perform a continuous learning monitoring the BP executions it is needed to evaluate the trend of process results. Specifically, this work assumes that by assigning different weights to decision makers there is a greater probability to move the organization towards the successful decision. So, implementing an algorithm capable to improve weights distribution according to the context of BP execution this work performs an approach fostering organizational learning. In fact, the decision resulting from GDM during process execution impacts on the process performance, that according to [15] can be assessed in two main dimensions, that are 'hard' and 'soft' goals. A binary result indicating whether the process has achieved its 'hard' goal, namely, a state the process intends to achieve (e.g., ordered goods are supplied to the customer). A result which can be evaluated as 'soft' goals [16] by means of Key Performance Indicators (KPI) indicating the extent to which business objectives have been achieved (e.g., time to delivery, quality level, costs). So, the results will be used to estimate the successfulness (or not) of each decision determining an impact on rewarding or punishing of learning policy defined in Section 5.

3. Heterogeneous Group Decision Making

One of the reasons why decision making processes have been widely studied in the literature is the increasing complexity of the social–economic environment. It is obvious that the comparison of different actions according to their desirability in decision problems, in many cases, cannot be done by using a single criterion or a unique person [17]. In fact, models and methodologies to face with GDM find wide application in several domains, like: emergency decision support [18], health management [19], and so on. The proposed framework is aimed to support decision making in the organization in the cases where business processes include the group decision making pattern shown in Fig.1. For instance, the innovation management, when

it is needed to decide whether to start the development of a new product (or service) the process involves decision makers belonging to several areas of the organization (e.g., market, IT, etc.) and the adopted strategy usually depends on the matter of discussion and on the current context of the organization, economy, etc. Analogously, the process of competence assessment developed via 360 degrees feedback, it requires an overall assessment result (needed to make a coherent and fair decision on, for instance, the rewarding) for the assessment we need to consider all the individual evaluations and try to find a convergence among all the opinions; and so on.

Accordingly, the core of the proposed framework is focused on the GDM problem introduced in [4]. Initially, GDM problems were defined in situations where all the decision makers' opinions were considered equally important [7]. Unlike, we focus on the decision processes in the framework of GDM stressing heterogeneity induced by the expertise in such contexts of the involved decision makers. Hence, CaHGDM relies on an extended version of GDM, namely *Heterogeneous Group Decision Making* (HGDM), whose theoretical approach has been introduced in [9]. In particular, we use two proposals of consensus models. The first one, presented in [4], is used because it provides the first approach to develop automatic consensus processes in which we could use the consensus and proximity measures to guide the consensus reaching process without the usual figure of moderator. On the other hand, we use a second approach currently defined in [9], because we apply in our framework the possibility of using the variable importance in the group decision making problems as a way to establish different importance degrees among the user opinions. Usually the importance degrees or weights are introduced in the decision models when the opinions are aggregated or when the consensus measures are computed. However, in [9] they are also used to guide the feedback mechanism of the consensus model which add a new value in the use of the weights.

In general, the consensus model foresees a resolution method composed by two different processes:

1) Consensus process: to obtain the maximum degree of consensus or agreement among the experts on the solution alternatives. In literature several works exploiting Fuzzy Theory in GDM there exist [20], [21], [22]. Analogously, the proposed work implements a soft consensus model in order to deal with vague or imprecise opinions during consensus process detailed in Section 3.1.

2) Selection process: a method to obtain the set of alternatives ranked according to the opinions given by the experts. Let us note that in our case it is not important the whole ranking of alternatives, but only to find consensus about the most valued alternatives.

The main distinguishing feature of HGDM is that the decision makers' opinions are not considered equally important, and so, the aggregation phase to obtain collective preference and feedback mechanism will be guided by the importance degrees of each decision maker. Following subsections provide formal definition of the HGDM process execution in the proposed framework.

3.1. Consensus reaching process

According to the notation introduced in Section 2, given a current process execution context c_c, a set of decision makers $DM = \{dm_1, dm_2, \ldots, dm_m\}$, $(m \geq 2)$ and a set of alternative branches $AB = \{ab_1, ab_2, \ldots, ab_n\}$, $(n \geq 2)$, we assume that experts provide their opinions in terms of fuzzy preferences values $ab^k_{dm_i,c_c} \in [0,1]$ that enable us to carry out the fuzzy preference relations as described in [14].

Definition 3.1. A fuzzy preference relation P on a set of alternatives X is a fuzzy set on the product set $X \times X$, i.e., it is characterized by a membership function $\mu_P : X \times X \to [0,1]$.

When cardinality of alternatives X is small, the preference relation may be conveniently represented by the $n \times n$ matrix $P = (p_{ik})$ being $p_{ik} = \mu_P(x_i, x_k)$ $(\forall i, k \in 1, \ldots, n)$ interpreted as the preference degree of the alternative x_i over x_k. Indifference between x_i and x_k is indicated by $p_{ik} = 1/2$, $p_{ik} = 1$ indicates that x_i is absolutely preferred to x_k, and $p_{ik} \geq \frac{1}{2}$ indicates that x_i is preferred to x_k. Based on this interpretation we have $p_{ii} = 1/2\ \forall i \in 1, \ldots, n$. For the sake of coherence to the well known consensus model introduced in [5], in the following we will refer to X as the set of alternative branches (that was AB) and to x_i to indicate the i-th alternative branch (that was ab^i).

Nevertheless, it is quite natural to consider that different experts will give their preferences in a different way. This leads us to assume that the experts' preferences over the set of alternatives, X, may be represented in different ways (e.g., preference ordering of the alternatives, fuzzy preference relation, utility function) [14]. Exploiting transformation function defined in [14], it is possible to obtain a uniform representation of the preferences into fuzzy preference relations. So, analogously to [12, 13], we consider fuzzy preference

relation as the base to uniform the information. Specifically, during the consensus calculus we transform fuzzy preference values in fuzzy preference relations augmenting the range of applicability of the proposed contribution. Definition 3.2 recalls this transformation function defined in [14].

Definition 3.2. Suppose that we have a set of alternatives, $X = \{x_1, ..., x_n\}$, and $ab^i_{dm_k,c_c}$ represents an evaluation associated to alternative x_i, indicating the performance of that alternative according to a point of view (expert or criteria) dm_k. Then, the intensity of preference of alternative x_i over alternative x_j, $p^k_{i,j}$, for dm_k is given by the following transformation function:

$$p^k_{ij} = \varphi(ab^i_{dm_k,c_c}, ab^j_{dm_k,c_c}) = \frac{1}{2} \cdot (1 + ab^i_{dm_k,c_c} - ab^j_{dm_k,c_c}) \tag{1}$$

Once this uniform representation has been achieved, we can apply a selection process as specified in accordance with the foundation of fuzzy consensus models. Considering the fuzzy preference relation matrices P^{dm} corresponding to the opinions of decision maker $dm \in DM$, next subsections will describe the calculation of the consensus degree reached at each iteration by the involved decision makers, and the feedback mechanism according to the importance degree, in the execution context, of the involved decision makers.

3.1.1. Computing consensus degrees

In order to calculate the consensus degree reached at each iteration, the moderator computes the *similarity matrix* $SM^{kl} = (sm^{kl}_{ij})$ for each pair of decision maker $\{dm_k, dm_l\}$, where $sm^{kl}_{ij} = 1 - |p^k_{ij} - p^l_{ij}|$, and $p_{ij} = \mu_P(x_i, x_j)$ is the intensity of the opinion alternative x_i over x_j. Starting from the similarity matrices, it is possible to compute the consensus degree achieved by aggregating the similarity matrices as follows

$$cm_{ij} = \phi(sm^{kl}_{ij}), k = 1, ..., m-1, l = k+1, ..., m \tag{2}$$

where ϕ is an aggregation function (e.g., arithmetic mean, median). After that, it is possible to calculate the global consensus degree co, as proposed in [21], considering the consensus degree corresponding to each alternative branch, as follows:

$$co = \frac{\sum_{i=1}^{n} ca_i}{n} \tag{3}$$

In equation (3), co is computed as the average of all the consensus degrees on each alternative x_i, namely ca_i, given by the following equation:

$$ca_i = \frac{\sum_{j=1,j\neq i}^{n}(cm_{ij} + cm_{ji})}{2(n-1)} \tag{4}$$

The consensus degree, co, represents the agreement among all the decision makers about the alternative branch to follow in the process execution. As a consequence, closer co is to one, then greater is the agreement among all involved decision makers. In particular, if the consensus degree is higher than a given threshold, i.e. if $co > \gamma_c$, the consensus process ends and the selection process can be applied to find the alternative branch to follow in the process execution. Otherwise, if the level of consensus is lower than the threshold, feedback mechanism will be implemented in accordance with the experts' weight values to identify the preferences that each expert should modify to increase the consensus degree level in the next round of consensus.

3.1.2. Feedback Mechanism

This work implements a feedback mechanism to advice experts on how to change their preferences. Specifically, we implement the following procedure to support moderator activity to determine the feedback to send to decision makers. In particular, Feedback process is achieved by calculating, for each decision maker, a proximity measure that indicates the similarity between the preferences of dm_k and the current global preference $P^c = (p^c_{ij})$, where:

$$p^c_{ij} = \frac{\sum_{k=1}^{m} p^k_{ij} \times \overline{w}_k}{m} \tag{5}$$

where $\overline{w}_k$ is the level of importance of decision maker dm_k normalized in $[0, 1]$ considering the number of decision makers involved in the GDM in the current process execution context c_c, i.e. w_{dm_k,c_c}.
The proximity measure on the relation pr^k, which takes into account the distance between each agent and the global preference is given by:

$$pr^k = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1,j\neq i}^{n} pp^k_{ij} + pp^k_{ji}}{2(n-1)} \tag{6}$$

where $pp^k_{ij} = 1 - |p^k_{ij} - p^c_{ij}|$ is the proximity measure on pairs of alternative (x_i, x_j).

The feedback mechanism proposed here is based on the supposition that those experts with lower level of importance (e.g., beginner, etc.) will need more advice than others with higher importance. In particular, we compute a customized amount of advice that varies in accordance with the experts' weight values [9].
Specifically, we identify the low-importance set of experts as decision makers whose level of importance is lower than the average of overall set of decision makers involved in the current process execution context c_c. Once the similarity measures have been computed, the feedback mechanism may use them to generate personalized advice to the decision makers. This activity is carried out in two phases:

1) Search for preferences. The consensus level should be improved by suggesting changes in the decision makers' preferences. To do it, the procedure identifies the preferences values where the agreement is smaller than a threshold, α_1, as follows:

$$P = \{(i,j)|cm_{ij} < \alpha_1\} \tag{7}$$

where $\alpha_1 = \sum_{i=1}^{n} \frac{\left(\sum_{j=1,j\neq i}^{n} cm_{ij}\right)}{(n^2-n)}$. The feedback is sent only to $dm_k \in DM_{low} = \{dm_k|w_k <= \tau\}$, where τ is fixed to $\frac{1}{m}$ with m the number of decision makers involved in the GDM execution.

2) Generation of advice. Once the feedback mechanism has identified the preferences pairs to be changed by the decision makers, the model shows right direction of the changes to achieve the agreement suggesting to increase the current assessment if $p_{ij}^k < p_{ij}^c$ or to decrease if $p_{ij}^k > p_{ij}^c$, for $(i,j) \in P$.

3.2. Selection process

The selection process can be applied to find the alternative branch to follow in the process execution, when the consensus degree is higher than a given threshold, i.e. if $co > \gamma_c$. Global preference $P^c = (p_{ij}^c)$ about the alternatives is transformed into a global ranking computing OWA operator [23], and the most preferred alternative branch will be followed to proceed with the execution of the process. Let us note that the selection process carries out a vector ab^* summarizing the final opinion, that is the concrete output of GDM.

4. Context-Aware Weighting

In [24], the concept of context awareness has been defined as the characteristic of a system to re-adapt itself according to the location where it is used, the collection of nearby people and objects, and so on. The authors of [25] use context information to provide relevant information or services to the users according to their tasks.

Recently, the role of context awareness has been emphasized also with respect to business processes. Specifically, the term has been applied to business theory in relation to contextual application design and BPM issues. Ultimately, the research stream concerning business process flexibility would deal with the flexible reconfiguration of processes in enterprise applications and workflow management systems. Reconfiguration should be accomplished in response to specific environmental contingencies. Moreover, in [26], the notion of context-aware process design is associated to an approach to support the engineering and use of flexible business processes in adherence to the underlying context.

Unlike the above mentioned approaches, the proposed framework exploits context awareness techniques in order to support decision making activities, within business processes execution. Thus, continuous monitoring of contextual variables is a prerogative and an integral component of context-awareness in processes. Values for such contextual variables could come from databases, sensors, and so on. It also incorporates the provisioning of near real-time data for decision making through content aggregators and business intelligence software.

Specifically, the proposed framework assigns a relative importance degree to each decision maker involved in the GDM (within a BP) based on an heterogeneous consensus process (see Section 3). The weight depends on the current context configuration (the set of all relevant contextual information) and the knowledge of previous opinions expressed by decision maker.

Firstly, in order to implement a context-aware assignment of weights to decision makers it is needed to define a *context model* and to store it as an element of the 4-tuples $\langle dm_i, c_j, \overline{ab}_{dm_i,c_j}, w_{dm_i,c_j} \rangle$ in the knowledge base including also previous decisions and weights (as introduced in Section 2).

Then, it is necessary to define a *context matchmaking* algorithm to retrieve the weight of a decision maker by comparing the current context to the ones associated to the decision maker. Context matchmaking must be executed for each decision maker involved in a specific GDM activity. Context model

and matchmaking are detailed in the following subsections.

4.1. Context Model

The context are supposed to be modelled by means of semantic technologies. In particular, Semantic Web languages and vocabularies like OWL2 and SKOS will be used.

As illustrated in Fig.3, the hierarchy of contexts (e.g., C_1, C_2, etc.) is induced by the values assumed by the property *hasFeature* that are organizational features represented with domain ontologies specified by means of SKOS vocabulary (e.g., Topic, Business Area, etc.). Thus, context features are characterized in terms of hierarchies allowing reasoning based on *broader* and *narrower* properties of SKOS. Specifically, the property *hasFeature* has been defined as generic relation that associates to each context the set of specific characteristics that belong to different ontologies. It specifies the *rdf:range* as the top class of SKOS vocabulary (i.e., *skos:Concept*):

```
<rdf:Property rdf:ID="hasFeature">
        <rdfs:domain rdf:resource="#Context"/>
        <rdfs:range rdf:resource="skos:Concept"/>
```

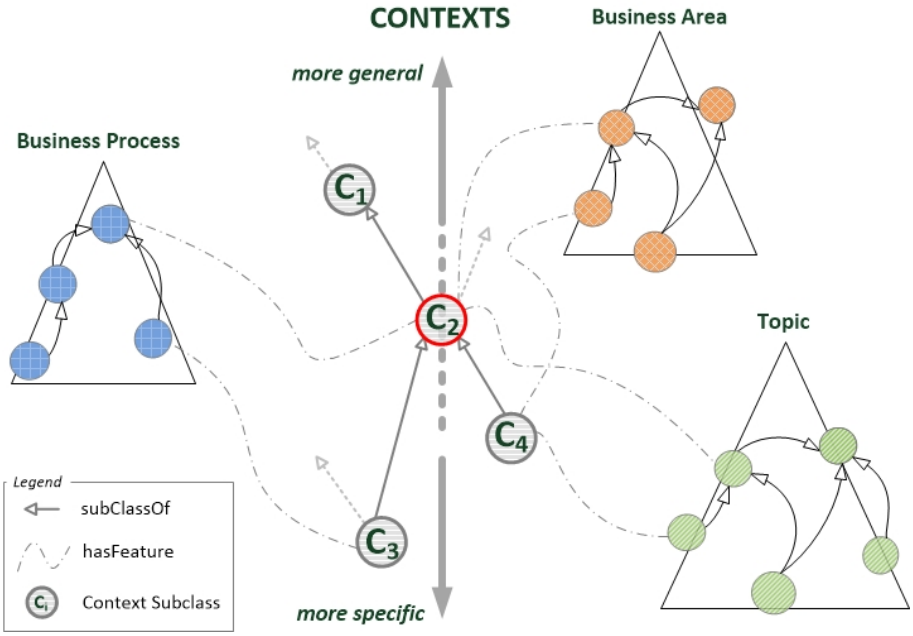

Figure 3: Context Modeling & Related Ontologies.

```
</rdf:Property>
```

Let us note that the relation $hasFeature$ may be specialized to differentiate relations, for instance, with concepts of Topic or Business Area vocabularies by means of $subPropertyOf$ construct of RDF, as follows:

```
<rdf:Property rdf:ID="hasRisk">
        <rdfs:subPropertyOf rdf:resource="#hasFeature"/>
        <rdfs:domain rdf:resource="#Context"/>
        <rdfs:range rdf:resource="#Topic"/>
</rdf:Property>
```

The set of values assumed by the property $hasFeature$ (or its specializations) identifies the underlying context with different level of granularity. In terms of Description Logic, the context (e.g., C_2) is defined as a class equivalent to the intersection of restrictions on the value of property $hasFeature$, that formally means:

$$C_2 \equiv \exists hasFeature.\{y_1\} \cap hasFeature.\{y_2\} \cap \cdots \cap hasFeature.\{y_L\} \quad (8)$$

where $y_i \in O_i$ for $i \in \{1, 2, \ldots, L\}$ are specific values of $hasFeature$ expected for C_2, these values are defined in the organizational domain context ontologies O_i (e.g., Topic, Business Area, etc.).
Considering the number of features and *broader* (or *narrower*) relations that may exist among values of $hasFeature$ properties, the contexts available in the knowledge base may be related each other by subsumption relation, for instance C_3 subsumes C_2, as illustrated in Fig.3. Thus, the context model enables to retrieve the weight to assign to each decision maker considering not only the same current context, but, if available in the knowledge base, it is possible to retrieve also the most similar ones where decision maker has participated in the past executions of GDM as described in the next section.

4.2. Context Matchmaking

The architectural module of Context-Aware Weighting implements the workflow described in Fig.4 integrating an algorithm of Context Matchmaking in order to retrieve the weight for each decision maker involved in the GDM activity. The weight is taken by considering the context where decision maker took decisions (for his/her organization) in the past experiences (executions of business processes). If this context exists, the corresponding weight is retrieved, otherwise it is needed to look for the closer context associated

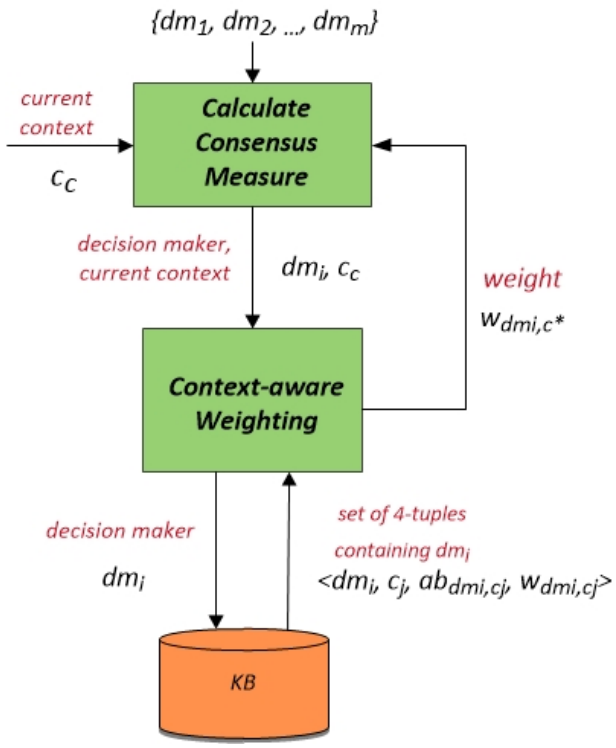

Figure 4: Context-Aware Weighting Workflow.

to such decision maker that exists in the knowledge base. Moreover, the corresponding weight is adjusted by applying a function that takes into account the hierarchy of contexts (broader or narrower contexts).
Specifically, given a decision maker dm_i and the current context c_c, it is needed to employ a context reasoning operation to select, from the knowledge base, the context that is most similar to the current context by analyzing the set of tuples corresponding to dm_i.

More formally, let us consider the following set of 4-tuples introduced in Section 2 corresponding to dm_i are listed following:

- $\langle dm_i, c_1, \overline{ab}_{dm_i,c_k}, w_{dm_i,c_k} \rangle$
- $\langle dm_i, c_2, \overline{ab}_{dm_i,c_k}, w_{dm_i,c_k} \rangle$
- ...
- $\langle dm_i, c_k, \overline{ab}_{dm_i,c_k}, w_{dm_i,c_k} \rangle$

For the sake of clarity, the 4-tuples are simplified as $\langle c_j, w_{c_j} \rangle_{dm_i}$ for any decision, because the context matchmaking uses only the associations between contexts c_j ($j \in \{1, 2, \ldots, k\}$) and weights w_{dm_i,c_j} that the system has stored in the knowledge base, to determine the weight of dm_i in the current context c_c of GDM.
In particular, the context matchmaking consists of finding the set C^* of contexts such that:

$$C^* = argmax_{dm_i} match(c_c, c_j) \qquad j \in \{1, 2, \ldots, k\} \tag{9}$$

where *match* is a function that given a pair of contexts, c_c and c_j, and their corresponding features, calculates a value in the range $[0; 1]$ that measures how much close are the input contexts. Specifically, for each pairs of feature values f_{c_c} and f_{c_j}, describing c_c and c_j respectively, the *match* function is defined as follows:

$$match(c_c, c_j) = \sum_{f_{c_c}, f_{c_j}} sim(f_{c_c}, f_{c_j}) \tag{10}$$

where $sim(f_{c_c}, f_{c_j})$ is the similarity measure such that f_{c_c}, f_{c_j} are two concepts in *is-a* taxonomy. Let us assume that f_{c_3} is the least common superconcept of f_{c_c} and f_{c_j}. $|\widetilde{f_{c_c}, f_{c_3}}|$ is the number of nodes on the path from f_{c_c} to f_{c_3}. $|\widetilde{f_{c_j}, f_{c_3}}|$ is the number of nodes on the path from f_{c_j} to f_{c_3}. $|\widetilde{f_{c_3}, root}|$ is the number of nodes on the path from f_{c_3} to *root*. Then, Wu and Palmer similarity [27] is exploited to calculate $sim(f_{c_c}, f_{c_j})$ as follows:

$$sim(f_{c_c}, f_{c_j}) = \frac{2 \times |\widetilde{f_{c_3}, root}|}{|\widetilde{f_{c_c}, f_{c_3}}| + |\widetilde{f_{c_j}, f_{c_3}}| + 2 \times |\widetilde{f_{c_3}, root}|} \tag{11}$$

Intuitively, the Wu & Palmer distance is proportional to the shortest path that connects two terms and presents good performances compared to the other similarity measures [28]. Finally, the algorithm retrieves the weight of dm_i such that:

$$w_{dm_i,c_c} = \min_{c^* \in C^*} w_{dm_i,c^*} \tag{12}$$

and, if not exists, the corresponding tuple $\langle dm_i, c_c, \overline{ab}_{dm_i,c_c}, w_{dm_i,c_c} \rangle$ is added to the knowledge base. Let us note that in the case there is a new decision maker, and C^* is empty or $w_{dm_i,c_c} = 0$, the proposed approach foresees the definition of an initial weight association strategy to determine how to deal with new decision maker.

5. Decision Makers Weights Learning

The proposed framework is aimed at improving the support to group decision making during business process execution assuming that the ability to make successful decisions is predicated upon the previous involvement in similar contexts. In particular, as introduced in Section 2, the proposed work update weights of decision makers in the knowledge base according to a learning model that considers the outcome and the context of each BP execution. Specifically, we are assuming that there is a way to evaluate the business process outcome by means specific KPIs at each execution of BP, and so we are able to evaluate if a decision taken each time is successful or unsuccessful.

Let us note that in the most cases it is not possible to know the consequences of performed decisions in terms of business process outcomes, so there is a need to perform an on-line learning algorithm. Analogously, reinforcement learning differs from standard supervised learning in that correct input/output pairs are never presented. Further, there is a focus on on-line performance, which involves finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge). So, reinforcement learning (RL) approach is a suitable theory for the proposed aims.

In general, RL relies on the fact that an agent can make an action u that applied to the environment changes its state from x to x'. The agent receives a reinforcement r. In our case the action is the opinion expressed by dm_i in the context c_c, that is the vector of preferences $\overline{ab}_{dm_i,c_c}$ (one element for each alternative branch in the business process), the state is represented by the weight assigned to $w^t_{dm_i,c_c}$ at time t that will be updated according to a specific rewarding or punishment functions and become the weight that will be used at time $t+1$, i.e. $w^{t+1}_{dm_i,c_c}$.

In more detail, the following formula summarizes the algorithm and represent how the weight w_{dm_i,c_c} at time t+1 will be updated:

$$w^{t+1}_{dm_i,c_c} = \alpha w^t_{dm_i,c_c} + \beta R(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) - \gamma P(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) \tag{13}$$

where R and P are rewarding and punishing functions respectively, α is parameter that controls the influence of the original weight, β and γ regulate the learning rate of the system. In particular, the functions R and P are calculated by comparing opinion of dm_i in the context c_c (i.e., $\overline{ab}_{dm_i,c_c}$) with the decision of GDM $\overline{ab}^*$. Specifically, if $\overline{ab}^*$ is a successful decision then R

and P are evaluated as follows:

$$R(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) = K(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) \tag{14}$$

and,

$$P(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) = 1 - K(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) \tag{15}$$

where $K(\overline{ab}_{dm_i,c_c}, \overline{ab}^*)$ is a function that implements a distance between two rankings [29] and it is inspired to the coefficients of Kendall [30] and Spearman [31] that evaluates the degree of similarity between ranked set of the alternative branches. This coefficient depends upon the number of inversions of pairs of objects which would be needed to transform one rank order into the other. More formally:

$$K(\overline{ab}_{dm_i,c_c}, \overline{ab}^*) = 1 - \frac{\sum_{j \in \{1,\ldots,n\}, rank(\overline{ab}^j_{dm_i,c_c}) \neq rank(\overline{ab}^{*j})} |\overline{ab}^j_{dm_i,c_c} - \overline{ab}^{*j}|}{n} \tag{16}$$

Vice versa, if $\overline{ab}^*$ is a not successful decision, R and P are inverted.

Let us note that, the decision maker that expresses the successful opinion, even if the cold start, has the chance to fill the gap with respect to more experienced decision makers because their weights will be punished with a greater value when their decision are unsuccessful for the organization. The underlying intuition of the above formula is to incrementally (execution by execution of BP) move the weight vector of each decision maker toward the cluster of successful decisions in the context c_c and away from the wrong ones. In this sense, we are implementing an organizational learning approach assuming that assigning different weights to decision makers there is a greater probability to move the organization towards the successful decision.

6. Case Study and Discussion

In this section a case study, executed to early evaluate the proposed approach, is described and its results are discussed. The case study has been simulated by means of the software prototype whose implementation details are described below.

6.1. Domain description

The domain we selected for the case study is the Supply Chain Management (SCM) and in particular we focused on those business processes including the *supplier selection*.

A supply chain is a set of organizations directly linked by one or more upstream and downstream flows of products, services, finances, or information from a source to a customer. SCM is the management of such a chain. SCM software includes tools or modules used to execute supply chain transactions, manage supplier relationships, and control associated business processes. SCM is concerned with the optimization of the resources of organizations in a network that delivers value to end customers. In the context of a supply chain, *supplier selection* is a highly important multi-criteria group decision making problem, which requires a trade-off between multiple criteria exhibiting vagueness and imprecision with the involvement of a group of experts (also with different background knowledge).

Different criteria can be considered during the decision making process for supplier selection decision include qualitative and quantitative factors. Improper selection of suppliers may unfavorably affect the company's competiveness strategy. Thus, this problem is naturally a multi-objective decision making problem with several conflicting factors.

Multiple decision makers are usually preferred to a single decision maker, because there is less chance of mistake when there are more than one decision makers. Moreover, it is preferable to build the group of decision makers by considering integration (heterogeneous experts) rather specialization (similar experts) for several motivations. For instance, it is important that more stakeholders (involved in the business process including the decision making activity) are represented and their interest are considered in the decision process. Lastly, people with different and heterogeneous specializations can develop a greater number of potential options and more creative options.

6.2. Technological implementation details

In order to simulate the above case study, we adopted jBPM[2], a Business Process Management System enabling modeling, design, implementation, execution, monitoring, simulation and optimization of business processes. In particular, we have designed and implemented the pattern described in

[2] `http://www.jbpm.org`

Fig.2. From the design/implementation viewpoint, a multiple-instance user task has been defined to provide a form to decision makers in order to allow them to express their opinions on a set of suppliers. The tasks called *Calculate Consensus Measure*, *Select Opinion* and *Generate and Send Feedback* have been realized as script tasks. Scripts are implemented in Java Programming Language and invoke the classes of the CaHGDM (Context-aware Heterogeneous Group Decision Making) library implementing the approach. In this case study, success measures of business processes are calculated off-line and are passed to a separate software component implementing the reinforcement learning algorithm and interacting with the knowledge base by means of Apache Jena. The simulation knowledge base has been developed by using Apache Jena and its embedded triple store, namely TDB[3].

6.3. Scenario description and objectives

In the context of the proposed case study we will provide one scenario, to be simulated, that is configured in the following way:

- five decision makers involved in all business process instances;
- seven suppliers, corresponding to the different alternative choices the decision makers have to make on each business process instance;
- four contexts in which decisions have to be made;
- all considered business processes instances includes only one decision making activity in which it is needed to select only one supplier among the set of seven ones.

The objective of the simulation is to study and analyze the insertion of a new (with no previous experience in taking decisions) decision maker with respect to the existing ones, considering a fixed context. More in details, the evolution (analytic curve) of the weights associated to the new decision maker in such a context is compared to those of a good decision maker, a fair decision maker and a poor decision maker in such a context. This analysis is repeated by assuming two initial weight association strategies to the new decision makers or to a decision maker that has not yet been involved in any similar context (i.e., C^* is empty, see Section 4.2):

[3] `https://jena.apache.org/documentation/tdb/`

Table 1: Knowledge Base at time t_0.

dm	c_1	c_2	c_3	c_4
dm_1	0.1	?	?	?
dm_2	0.0	?	?	?
dm_3	0.60	0.35	0.8	0.35
dm_4	0.40	0.35	0.4	0.7
dm_5	0.40	0.9	0.35	0.1

- *Starting Weight*. This strategy foresees an experimentally fixed starting weight to the decision maker, that is 0.1 in our test case. The weight will change according to the learning weights policy (see Section 5);
- *Training Executions*. This strategy foresees an experimentally fixed number of executions in which the decision maker has any influence (i.e., the weight is frozen to 0.0), that is set to 5 process executions in our test case. During these executions the decision maker will express his opinions and the system will trace them without any update to the weight in the knowledge base. After 5 participations, the system will assign to the decision maker the weight corresponding to the weight matured from the previous opinions.

The following subsections provide, respectively: details about a single process execution in terms of consensus calculus and knowledge base update (see Section 6.4); discussion about the decision makers' weights evolution in the knowledge base after some process executions (see Section 6.5) according to the strategy introduced above.

6.4. Executions

The initial situation is provided by assigning to all decision makers a weight for the corresponding context. Table 1 represents the knowledge base at time t_0 and shows the weights associated to each decision maker for the corresponding context c_i (for $i \in \{1, \ldots, 4\}$). Let us note that two of the five decision makers have no past experience, therefore their weights will be differently treated according to the initial weight association strategies (i.e., 0.1 if it is the *Starting Weight* strategy, 0.0 if it is *Training Executions* strategy).

Specifically, in the following simulation we apply *Starting Weight* strategy for decision maker dm_1, and *Training Executions* strategy for decision maker

Table 2: Parameters of 10 process executions.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
C_C	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1
Outcome	1	1	-1	0	1	1	-1	1	1	1

dm_2. Furthermore, let us consider ten executions according to the parameters shown in Table 2, where: t_i is the execution time; C_C is the process execution contexts at the corresponding execution time; and *Outcome* represents the value ranging in the set $\{1, 0, -1\}$ whose values stand for successful, neutral or unsuccessful decision taken at each execution, respectively. For the sake of clarity, we have simulated process execution only in the context c_1.

The execution at time t_1 involves all decision makers in the knowledge base and the process execution context is c_1. The weights of each involved decision maker are normalized (in $[0, 1]$) to satisfy the convex combination according which all coefficients are non-negative and sum is equal to 1, so they becomes:

$$w_{c_1,dm_1} = 0.066^4, \quad w_{c_1,dm_2} = 0.0, \quad w_{c_1,dm_3} = 0.4,$$
$$w_{c_1,dm_4} = 0.266, \quad w_{c_1,dm_5} = 0.266$$

Let us suppose that the consensus reaching process loops until the consensus degree *co* is lower than $\gamma_c = 0.80$. Furthermore, let us consider a group decision making problem with respect to four alternative branches (namely, $\{x_1, \ldots, x_4\}$), the execution will carry out the alternative branch to follow in order to proceed with the process execution. Specifically, let us suppose the decision makers' opinions illustrated in Table 3. Then, the derived preference relation matrices are:

P1	x1	x2	x3	x4
x1	0.5	0.45	0.55	0.4
x2	0.55	0.5	0.6	0.45
x3	0.45	0.4	0.5	0.35
x4	0.6	0.55	0.65	0.5

P2	x1	x2	x3	x4
x1	0.5	0.35	0.25	0.05
x2	0.65	0.5	0.4	0.2
x3	0.75	0.6	0.5	0.3
x4	0.95	0.8	0.7	0.5

[4]According to the Starting Weight strategy, the weight associated to dm_1 is 0.1 and becomes 0.066 after normalization.

P3	x1	x2	x3	x4
x1	0.5	0.6	0.7	0.85
x2	0.4	0.5	0.6	0.75
x3	0.3	0.4	0.5	0.65
x4	0.15	0.25	0.35	0.5

P4	x1	x2	x3	x4
x1	0.5	0.45	0.3	0.2
x2	0.55	0.5	0.4	0.25
x3	0.65	0.6	0.5	0.35
x4	0.8	0.75	0.65	0.5

P5	x1	x2	x3	x4
x1	0.5	0.35	0.25	0.05
x2	0.65	0.5	0.4	0.2
x3	0.75	0.6	0.5	0.3
x4	0.95	0.8	0.7	0.5

By performing the equation of the consensus calculus (see Section 3.1.1) we obtain the following results:

- Computing consensus degrees: $co = 0.78$.
- Controlling the consensus process: as $co < \gamma_c = 0.80$, the feedback mechanism is activated.

In order to achieve the agreement, the feedback mechanism is performed (see Section 3.1.2). It foresees the following activities: selection of the decision makers whose weight is lower than average of the experts involved in the group decision making; computing of proximity matrices and consequently generation of the advice. Specifically, we obtain:

- The average weight of involved experts is $ave = 0.2$, and so the decision makers that will receive the advice are ones such that:

$$\{dm_i | w_{c_1,dm_i} \leq ave = 0.2\} = \{dm_1, dm_2\}$$

Table 3: Decision makers' opinions.

	x_1	x_2	x_3	x_4
$\overline{ab}_{dm_1,c_1}$	0.2	0.3	0.1	0.4
$\overline{ab}_{dm_2,c_1}$	0.1	0.4	0.6	1
$\overline{ab}_{dm_3,c_1}$	1	0.8	0.6	0.3
$\overline{ab}_{dm_4,c_1}$	0.4	0.5	0.7	1
$\overline{ab}_{dm_5,c_1}$	0.1	0.4	0.6	1

- The generated advice suggesting to each decision maker (with $w_{c_1,dm_i} < w^*_{c_1}$ for $i \in \{1, 2, \ldots, 5\}$) the right direction of the changes to approach the agreement for the corresponding preferences pairs, are:

dm_1	x_1	x_2	x_3	x_4
x_1	-	-	-	-
x_2	-	-	-	-
x_3	-	-	-	-
x_4	-	-	-	-

dm_2	x_1	x_2	x_3	x_4
x_1	-	-	-	-
x_2	-	-	-	-
x_3	-	-	-	-
x_4	-	-	-	-

Let us suppose that after 3 rounds the consensus is achieved for alternative branch x_4 with the following opinion vector $\overline{ab}^*$:

	x_1	x_2	x_3	x_4
$\overline{ab}^*$	0.47	0.49	0.49	0,54

Now according to the input decision makers' opinions, it is possible to update in the knowledge base the weights of decision makers corresponding to the context c_1 performing the process described in Section 5, taking into account the similarity between opinions provided and $\overline{ab}^*$ considering that according to Table 2 the $\overline{ab}^*$ at time t_1 is assumed to induce a successful decision (i.e., 1). In particular, after the learning procedure the resulting knowledge base is detailed in Table 4, assuming that learning and punishing parameters are set as follows:

$$\alpha = 1, \quad \beta = 0.2, \quad \gamma = 0.15.$$

Analogously to the described execution, other 9 executions of the business process have been performed, accordingly to Table 2 and the weight evolution curves are shown in Figure 5.

Table 4: Knowledge Base at time t_1.

dm	c_1	c_2	c_3	c_4
dm_1	**0.52**	?	?	?
dm_2	**?**	?	?	?
dm_3	**0.5**	0.35	0.8	0.35
dm_4	**0.6**	0.35	0.4	0.7
dm_5	**0.6**	0.9	0.35	0.1

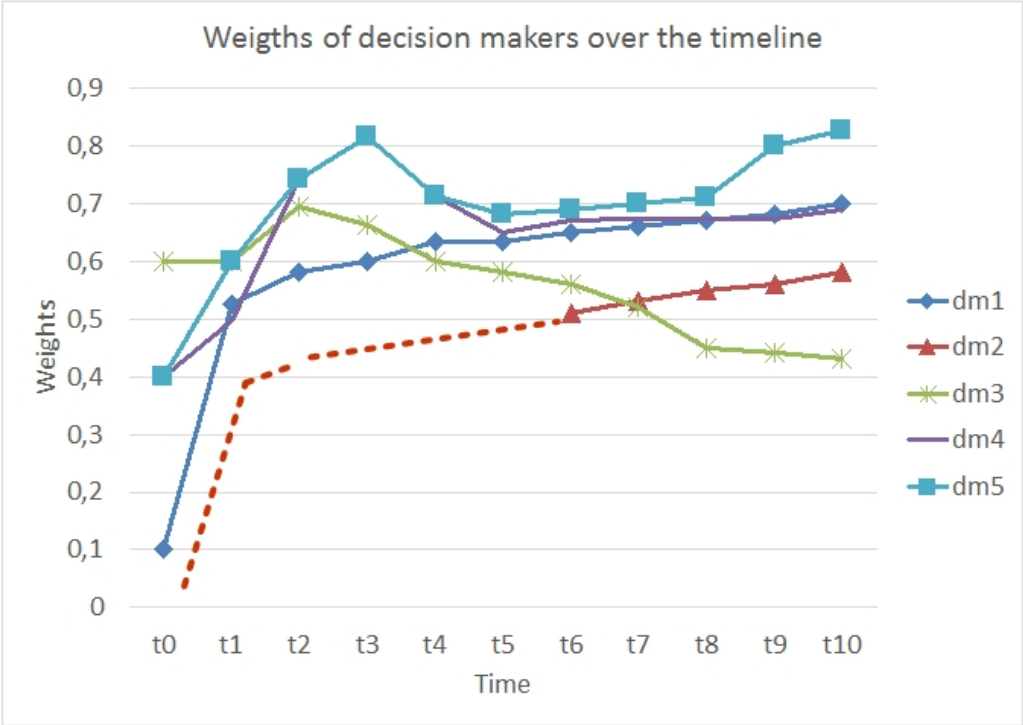

Figure 5: Weights learning curves.

6.5. Discussion

As shown in Figure 5, reporting the results of the simulation described in the previous sections, the decision maker dm_1 reaches the weight assigned to the medium performance decision maker, namely dm_4, at time t_8 and their weights are almost the same already from time t_5. Thus, it is clear that if a new decision maker is introduced with the *Starting Weight* approach and his/her decisions are good he/she takes almost 5 decisions to reach the importance of a medium performance decision maker. Moreover, take a look to dm_1 with respect to the poor performance decision maker dm_3. In this case, dm_1 overcomes dm_3 by taking only 4 iterations (decisions). Furthermore, dm_1 takes 5 decisions to reach the minimum distance from the weight of dm_5 who is the good performance decision maker and if you consider the situation at time t_{10} you can observe that dm_1 does not reach the weight of dm_5. Now, let focus on decision maker dm_2 who is introduced in the decision making processes with initial weight calculated by using the

Training Executions approach. Let us note that dashed curve associated to dm_2 means that the weight of decision maker evolve but it is not considered in the decision making process until t_6. dm_2 overcomes the poor performance of decision maker dm_3 in 6 iterations and starts to have comparable weight at time t_{10}.

It is clear that the *Training Executions* approach is more conservative than the *Starting Weight* approach although also the first one allows new decision makers, who demonstrates good decision performances, to acquire a fairly significant weight in the GDM activities by taking only few iterations.

In general, one of the merit of proposed approach and, more specifically, of the weight learning algorithm is that it implements an organizational learning approach assuming that assigning different weights to decision makers there is a greater probability to move the organization towards the successful decision. Nevertheless, when a decision maker is involved for the first time into a GDM activity may happen that there are no past decision information in the knowledge base (neither for similar contexts). So, the initial weight association strategy has to implement the right trade-off avoiding to quickly assign high influence in the decision making process, but also it has to guarantee the growth of influence according to the evaluated performances. Indeed, thinking to the decision where more experienced people and less ones participate, it is important for the organization to guarantee a correct and effective turn over management. In this sense, the simulation results reveal that, even if the cold start (*Starting Weight* and *Training Executions*), the framework gives decision-makers the chance to fill the gap with respect to more experienced decision makers. Furthermore, monitoring and analyzing the weights of decision makers and the outcomes of BP executions in different contexts, it is possible to:

- observe the growth of the employees, not only in terms of technical competencies, but also with respect to cross skills such as decision making;
- improve the selection (when it is possible) of decision makers according to the context in which specific decisions have to be made with the right distribution of the weights.

7. Conclusions and Future Works

The process execution monitoring is an important requirement to detect elements enabling improvements for next executions. Many research works define approaches to automatically or semi-automatically support human decision making activities, within a Business Process (BP). Last trend emphasize the importance of past execution analysis to derive decision criteria. Nevertheless, these criteria have to be context-sensitive since context and matter of decision may differ from situation by situation. The main contribution of the proposed work is the definition of a framework to support group decision making during business process execution considering heterogeneous importance level of each human decision maker according to the occurring context where they are involved. The framework implements a novel approach to learn weight of the decision-makers through the analysis of past process executions considering context and performances of business processes. The framework has been instantiated in the case study of Supply Chain Management. The analysis of the simulation results reveal that the proposed weight learning algorithm, even if the cold start (*Starting Weight* and *Training Executions*), give to decision-makers the chance to fill the gap with respect to more experienced decision makers.

In particular, the distinctive features of the proposed work are summarized as follows:

- Definition of framework supporting context-aware decision making into BP considering the specific environment and matter of discussion in which processes run;
- Definition of Fuzzy Consensus Model to Heterogeneous Group Decision Making problem extending the approach proposed in [9] and focusing essentially on two kinds of heterogeneity, that are: context of BP execution and different backgrounds and levels of knowledge about the problem for each expert;
- Application of Reinforcement Learning algorithm to weight the relative importance of the decision makers, considering the past successful decisions taken by them.

Definitely, the proposed framework aims at providing significant results with respect to the area of BP improvement and, in particular, it is focused on those BP that include human decision making activities. Future directions about the proposed framework mainly regard the following aspects:

- Exploitation of decision makers' competencies, in addition to the context, to assign weights before starting a group decision making process. Competencies will be represented by means of Semantic Web technologies enabling ontologycal reasoning as well as presented in this work for the context;

- More formally representation of the enterprise strategies and goals in order to provide a finer way to assess the result of a business process and, consequently, empower the whole proposed approach;

- Automation of decision makers selection for the occurring process execution context considering with the aim to optimize the consensus reaching procedure.

Furthermore, some limitations about the proposed work are related to the weight learning process that could be overcame in the future: updating the weight of the decision maker corresponding, not only to the occurring context, but also for the similar contexts; and taking into account how many times decision maker participates in a GDM in a specific context and the elapsed time from the last involvement.

Acknowledgments

This work was supported by the FEDER funds under Grant TIN2013-40658-P and Andalusian Excellence Project TIC-5991.

References

References

[1] M. Hammer, What is business process management?, in: Handbook on Business Process Management 1, Springer, 2010, pp. 3–16.

[2] J. Morente-Molinera, I. Pérez, M. Ureña, E. Herrera-Viedma, Building and managing fuzzy ontologies with heterogeneous linguistic information, Knowledge-Based Systems 88 (2015) 154–164.

[3] J. Wu, F. Chiclana, E. Herrera-Viedma, Trust based consensus model for social network in an incomplete linguistic information context, Applied Soft Computing (2015).

[4] F. Herrera, E. Herrera-Viedma, et al., A model of consensus in group decision making under linguistic assessments, Fuzzy sets and Systems 78 (1996) 73–87.

[5] E. Herrera-Viedma, S. Alonso, F. Chiclana, F. Herrera, A consensus model for group decision making with incomplete fuzzy preference relations, Fuzzy Systems, IEEE Transactions on 15 (2007) 863–877.

[6] G. Aagesen, J. Krogstie, Bpmn 2.0 for modeling business processes, in: Handbook on Business Process Management 1, Springer, 2015, pp. 219–250.

[7] F. J. Cabrerizo, J. M. Moreno, I. J. Pérez, E. Herrera-Viedma, Analyzing consensus approaches in fuzzy group decision making: advantages and drawbacks, Soft Computing 14 (2010) 451–463.

[8] F. J. Cabrerizo, F. Chiclana, R. Al-Hmouz, A. Morfeq, A. S. Balamash, E. Herrera-Viedma, Fuzzy decision making and consensus: challenges, Journal of Intelligent & Fuzzy Systems 29 (2015) 1109–1118.

[9] I. J. Pérez, F. J. Cabrerizo, S. Alonso, E. Herrera-Viedma, A new consensus model for group decision making problems with non-homogeneous experts, Systems, Man, and Cybernetics: Systems, IEEE Transactions on 44 (2014) 494–498.

[10] J. Morente-Molinera, R. Wikström, E. Herrera-Viedma, C. Carlsson, A linguistic mobile decision support system based on fuzzy ontology to facilitate knowledge mobilization, Decision Support Systems 81 (2016) 66 – 75.

[11] S. Alonso, F. J. Cabrerizo, F. Chiclana, F. Herrera, E. Herrera-Viedma, Group decision making with incomplete fuzzy linguistic preference relations, International Journal of Intelligent Systems 24 (2009) 201–222.

[12] F. Chiclana, F. Herrera, E. Herrera-Viedma, Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations, Fuzzy sets and Systems 97 (1998) 33–48.

[13] F. Chiclana, F. Herrera, E. Herrera-Viedma, Integrating multiplicative preference relations in a multipurpose decision-making model based on fuzzy preference relations, Fuzzy sets and systems 122 (2001) 277–291.

[14] E. Herrera-Viedma, F. Herrera, F. Chiclana, A consensus model for multiperson decision making with different preference structures, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 32 (2002) 394–402.

[15] J. Ghattas, P. Soffer, M. Peleg, Improving business process decision making based on past experience, Decision Support Systems 59 (2014) 93–107.

[16] I. Bider, P. Johannesson, P. Soffer, Y. Wand, On the notion of soft-goals in business process modeling, Business Process Management Journal 11 (2005) 663–679.

[17] J. Morente-Molinera, I. Pérez, M. Ureña, E. Herrera-Viedma, Creating knowledge databases for storing and sharing people knowledge automatically using group decision making and fuzzy ontologies, Information Sciences 328 (2016) 418–434.

[18] A. Choudhury, R. Shankar, M. Tiwari, Consensus-based intelligent group decision-making model for the selection of advanced technology, Decision Support Systems 42 (2006) 1776–1799.

[19] L. Yu, K. K. Lai, A distance-based group decision-making methodology for multi-person multi-criteria emergency decision support, Decision Support Systems 51 (2011) 307–315.

[20] F. J. Cabrerizo, R. Ureña, W. Pedrycz, E. Herrera-Viedma, Building consensus in group decision making with an allocation of information granularity, Fuzzy Sets and Systems 255 (2014) 115 – 127. Theme: Decision and Optimisation.

[21] F. J. Cabrerizo, J. A. Morente-Molinera, I. J. Pérez, J. López-Gijón, E. Herrera-Viedma, A decision support system to develop a quality management in academic digital libraries, Information Sciences 323 (2015) 48–58.

[22] E. Herrera-Viedma, F. J. Cabrerizo, J. Kacprzyk, W. Pedrycz, A review of soft consensus models in a fuzzy environment, Information Fusion 17 (2014) 4 – 13. Special Issue: Information fusion in consensus and decision making.

[23] R. R. Yager, On ordered weighted averaging aggregation operators in multicriteria decisionmaking, Systems, Man and Cybernetics, IEEE Transactions on 18 (1988) 183–190.

[24] B. Schilit, N. Adams, R. Want, Context-aware computing applications, in: Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on, IEEE, pp. 85–90.

[25] A. K. Dey, Providing architectural support for building context-aware applications, Ph.D. thesis, Georgia Institute of Technology, 2000.

[26] M. Rosemann, J. C. Recker, Context-aware process design: Exploring the extrinsic drivers for process flexibility, in: The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium, Namur University Press, pp. 149–158.

[27] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 133–138.

[28] D. Lin, An information-theoretic definition of similarity., in: ICML, volume 98, pp. 296–304.

[29] R. Kumar, S. Vassilvitskii, Generalized distances between rankings, in: Proceedings of the 19th international conference on World wide web, ACM, pp. 571–580.

[30] M. G. Kendall, A new measure of rank correlation, Biometrika 30 (1938) 81–93.

[31] C. Spearman, The proof and measurement of association between two things, The American Journal of Psychology 15 (1904) 72–101.