MISE 3.0: An Agile Support for Collaborative Situation

Frederick Benaben, Matthieu Lauras, Sébastien Truptil, Jacques Lamothe

MISE 3.0: An Agile Support for Collaborative Situation

Frederick Benaben[1], Matthieu Lauras[1], Sébastien Truptil[1], Jacques Lamothe[1],

[1] Ecole des Mines d'Albi-Carmaux, Campus Jarlard, Route de Teillet,
81000 Albi, France
{frederick.benaben, Matthieu.lauras, Sebastien.Truptil, Jacques.Lamothe}@mines-albi.fr

Abstract. Mediation Information System Engineering project is starting its third iteration (MISE 3.0). The main objective of this paper is to introduce that version. MISE 3.0 aims at defining and designing a platform, dedicated to detect, initiate and support any collaboration opportunity among potential partners (obviously based on results inherited from MISE 1.0 and MISE 2.0). This MISE 3.0 platform implements the same model-driven engineering approach than MISE 1.0 and MISE 2.0. This approach is structured according to four layers: (i) gathering of individual and collaborative knowledge, (ii) design of potential collaborative behavior, (iii) deployment of accurate collaborative behavior and (iv) management and adaptation of collaborative behavior. However, this new platform is dedicated to provide improvements such as continuous working, performance measurement, smart monitoring and cloud deployment, which are the scientific backbone of this paper.

Keywords: Model-Driven Engineering, Interoperability, Key Performance Indicator, Decision Support System.

1 Introduction

Organizations (of any kind) embedded in today's economic environment are deeply dependent from their ability to take part into collaborations. Consequently, it is strongly required for them to assume the needed interoperability functions: exchange of information, coordination of functions and orchestration of processes. Furthermore, inside these organizations, Information Systems (IS) and computerized systems are assuming both the roles of interface (external and internal exchanges) and functional engine (driving processes and business activities). Therefore, IS, must be supporting the previously listed interoperability functions. The issue is to ensure that partners' IS will be able to work altogether (thanks to these interoperability functions) in order to constitute a coherent and homogeneous set of IS (the IS of the collaborative situation). Providing organizations with methods, tools and platforms able to ensure these interoperability functions makes therefore sound sense.

The MISE project (Mediation Information System Engineering) has been launched in 2004 and is dedicated to provide an approach (and the associated tools) for Mediation Information System (MIS) design. The so obtained MIS should ensure the interoperability functions (translation of data, sharing of services and orchestration of workflows) in an agile manner. Actually, collaborations are very unstable situations requiring adaptation: context can change (new opportunity, modification of

objectives, etc.), network of partners can change (withdrawal or arrival of partner, lack of resource, etc.) or dysfunction during the collaborative behavior can occur (even if context and partners are still the same, something may not happen as expected). Therefore, the MIS should remain well adapted to the potentially changing needs of the collaboration. Two iterations of the MIS project have already been performed. MISE 1.0 is presented in [1] and [2] while MISE 2.0 is presented in [3] and [4]. The third iteration, MISE 3.0, is ongoing and this article aims at presenting how this version intends to support collaborative networks in the Internet of services.
Second section of this article provides an overview of the three iterations of MISE projects, their links, their specificities and their logical structure. Third section presents specifically the MISE 3.0 iteration and the associated features for each step of MISE structure. Fourth section concerns conclusion and perspectives about MISE.

2 MISE Iterations

2.1 General overview of MISE approach

This overall MISE design approach might be seen as a dive into abstraction layers based on model-driven engineering [5]. The general principle of the MISE approach (whatever the iteration considered) is structured according to three steps:

1. Design of collaboration model: this level concerns the gathering of knowledge about the considered collaborative situation in order to instantiate concepts of the so-called collaborative metamodel (concerning mainly *environment* of the collaboration, *objectives* of the collaboration, *partners* and services of the collaboration).
2. Deduction of collaborative behavior model: the second step deals with the automated deduction of collaborative processes, based on the knowledge collected at the previous level. Schematically, the aim is to select and organize *partners' services* according to *objectives* and *environment* of the collaboration.
3. Deployment of the appropriate MIS: the previously deduced *business* behavior (processes) is translated in a technical behavior (workflows) in order to be implemented. The goal is mainly to match *services* with *activities* and *data* with *information*.

Furthermore, these three steps are used in an agile framework, which deal with *detection of evolution* and *adaptation of behavior*. Performing agility of MIS is based on event analysis (according to the received event, is the situation in line with what is expected) and on behavior adaptation (by invoking step 1, step 2 or step 3 depending on the nature of the event analysis). On a technical point of view, MISE project is based on a Service Oriented Architecture (SOA) paradigm and MISE tools are deployed as web-services on an Enterprise Service Bus (ESB). Even if there are some differences and specific features, each of the three iterations of MISE project is structured according to the three previously presented steps and the associated agile framework. Furthermore, on a technical point of view, these iterations are all centered on SOA principles and on web-services. The following picture illustrates the global

MISE approach (three steps in an agile framework) and underlines schematically the specificities of first and second iterations:

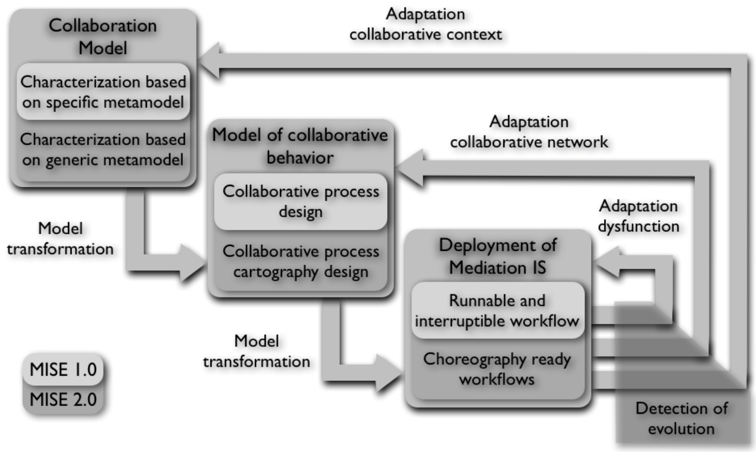

Fig. 1. MISE project overall structure including MISE 1.0 and MISE 2.0 iterations.

On the previous figure, the three steps of MISE approach are represented from MISE 1.0 and MISE 2.0 perspectives. The *three steps* of MISE structure are presented in a waterfall sequence together with *detection* mechanism and *adaptation* loops. For every step, both first MISE iterations specificities are mentioned. It is crucial to notice that there are in fact four "so-called" steps in MISE approach, but, in the previous big picture, the first three steps (dedicated to design-time) are presented as boxes while the last one (dedicated to run-time) is represented through the three looping arrows.

2.2 MISE 1.0 and MISE 2.0 articulation

MISE 1.0 uses domain specific metamodels (crisis management, manufacturing, etc.) to gather the knowledge in a meaningful collaborative situation model. That knowledge is extracted and transformed (according to [2] and [7]) to provide one single appropriate collaborative process dedicated to support the characterized (thanks to the gathered knowledge) collaborative situation. An additional knowledge concerning information about technical services (applications or functions) is then imported to define how activities of this collaborative process model may be concretely achieved and orchestrated. Once that additional knowledge integrated, the process model is transformed into a workflow model that can be run (thanks to an ESB and its workflow engine). There are several drawbacks with that first version of MISE. Most important ones are the following:

- The use of domain specific metamodels does not allow the approach to be relevant for any kind of collaborative situation. Furthermore, there are several associated knowledge bases (one per metamodel), which cannot be used conjointly. Consequently, the concerned knowledge elements and the embedded behavioral schemes should be duplicated (or abandoned).

- Deducing one single collaborative process is not very relevant. First, most organizations are structured according to *decisional*, *operational* and *support* processes (ISO 9000-2001 recommendations [10]). Consequently, it would be significant to structure the deduced behavior according to that schema and to obtain processes covering *decisional*, *operational* and *support* views.
- The transition from business process (embedding business activities and business information) to technical workflows (concerning technical services and technical data) is quite raw: the way the technical description of services is integrated in workflow models is automated (through model transformation) but the precise selection is manual.
- Concerning agility (defined as "*detection + adaptation*"), if the adaptation functionality is assumed by the service-oriented structure, which allows to invoke design-time services at any required moment (in order to re-define the appropriate behavior), the detection functionality is fully manual, based on human analysis of reports and information coming from the situation.

Considering the previous elements, MISE 2.0 aims at reusing MISE 1.0 results and adding some new features. Therefore, one single metamodel (representative of collaborative situations has been defined [4]). This metamodel, the instances of the associated ontology (*i.e* the ontology structured according to this metamodel) and associated deduction rules (defined from concepts of the considered metamodel and dedicated to deal with instances of the associated ontology) can hence be used in any collaborative situation. This structural improvement reduces the first listed drawback. In addition, MISE 2.0 uses an objective typology to deduce a complete collaborative process cartography including several processes, which are typed as *decisional*, *operational* and *support* processes. This point tackles the second drawback. Besides, semantic reconciliation mechanisms have been injected (as described in [3]) in order to deal with the transition from business processes to technical workflow (*i.e.* the third drawback of the previous list). This improvement uses semantic annotations of business activities on the one hand and of technical services on the other hand, in order to select the most appropriate subset of technical services to ensure the behavior described by the considered business activities. Based on semantic annotations of information, these research results also provide on-the-fly data translation in order to assume correct orchestration of the selected technical services. Finally, an event-driven architecture (including a complex-event processing tool [11]) is added to the service-oriented structure of the MIS. This improved technological platform provides two main interests. The first one concerns choreography of multi-processes. Deducing a collaborative process cartography implies to be able to orchestrate each workflow but also to manage the coordination of these workflows. Workflow orchestration is assumed by the SOA structure while coordinating several workflows is assumed by the EDA structure (through choreography). The second one concerns the detection part of agility. Services (but also other devices or sensors) are able to send events. These events might be used by the system to detect any unexpected situation. This diagnosis mechanism is a solution to reduce the fourth identified drawback [12]. The following table summarizes the specificities of MISE 1.0 and MISE 2.0.

MISE 1.0 and MISE 2.0 are associated with some concrete application fields. For instance, ISyCri project concerns MISE 1.0 in crisis management context [6], while ISTA3 project concerns MISE 2.0 in manufacturing scope [13].

Table 1. Specificities of first and second iterations according to steps of MISE approach.

	MISE 1.0	MISE 2.0
Collaboration Model	Domain specific metamodels have been defined, depending on considered business fields (crisis management [6], manufacturing context [7])	One generic metamodel, dedicated to all types of collaborative situations has been defined (including external layers, enclosing domain specific concepts)
Model of Collaborative behavior	One single collaborative process has been deduced from the gathered knowledge.	Decisional, Operational and Support processes have been deduced from the gathered knowledge.
Deployment of Mediation Information System	After manual identification of technical services (or user-interfaces) that would assume identified business activities of the deduced collaborative process, the process is translated in BPEL language in order to be computerizable.	Automatic semantic reconciliation allows selecting subsets of technical services that will be invoked to assume business activities of collaborative processes on a technical point of view. Furthermore, ontological tools ensure “on-the-fly” data conversion [3].
Agility (detection + adaptation)	Detection is a manual task based on the way situation evolves. Once detected a need of adaptation, design-time tools (model editor, process deducing tool, workflow translator) may be invoked on purpose in order to (re)define the collaborative behavior appropriate for the “new” situation.	Detection is based on an EDA. Sensors and services publish their events (reporting on the situation and on workflow progress) that can be used to update situational models. If the current model differs from the expected model, then adaptation must be started based on the same principle than MISE 1.0.

However and obviously, there are still drawbacks in the MISE approach. First, MISE 2.0 only focuses on some main drawbacks. Consequently, there are still “second order” problems. Second, new features potentially bring new drawbacks that should also be considered. Following section presents these complementary drawbacks and introduces MISE 3.0 as a potential way to reduce them.

3 Specific Improvements of MISE 3.0

MISE 1.0 and MISE 2.0 did provide an improved solution for collaborative situation support by deploying a MIS between heterogeneous organizations. However, even if MISE 1.0 provides a first conceptual backbone and a full suit of tools, even if MISE 2.0 provides some tangible improvements and fixes some critical problems, there are still some concrete research avenues to explore.

3.1 Knowledge Gathering: Collaboration Model

In MISE 1.0 and MISE 2.0, knowledge gathering is based on a specific filling (by the user) of the instantaneous information available concerning the collaborative situation (its objectives, its specificities and the means available to achieve these objectives). In

MISE 3.0, the ambition is to use Event Driven Architecture (EDA) to continuously gather the knowledge (about organizations and situation) and continuously update the models (describing organizations and situation). The principle is to use an *event market place*, where each service and each device of the considered ecosystem publish its own events (*i.e.* reports, messages and information describing its status). By watching this *event market place* the system obtains a continuous *image* of the considered ecosystem. Moreover, the collected *events* are used to instantiate the collaboration metamodel and to create the specific instances of the *model* of the courant situation. By observing this *model* the system can diagnose any collaboration opportunity (for instance by checking some specific variables or detecting some significant patterns). Furthermore, when diagnosing any collaboration opportunity, the required *collaboration model* is already fulfilled, available and operational, thanks to this event-based principle.

3.2 Behavior Design: Model of Collaborative Behavior

In MISE 1.0 and MISE 2.0 the collaborative process(es) deduction is "binary": the apparently most appropriate structure of activities is built and is the result of the deduction step. However, MISE 3.0 includes a more soft principle, which (i) provides several models of potential behavior (depending on different options, different priorities and different layouts of relevant activities) and (ii) integrates decision support system to assist the user in selecting the most suitable one.

Regarding the decision support system, an important feature concerns Key-Performance Indicators (KPI). Because, the idea is to deduce not only the adequate collaborative behavior but also the associated indicators we propose to define two sets of KPI. The first one (inspired by [13]) allows comparing objectively the different scenarios of collaboration (on business and technical points of view) during second and third steps of MISE. The second one consists in designing a performance measurement system able to support the control of the most relevant collaborative workflows (inspired by [9]) during the fourth step of MISE.

Finally, second step of MISE 3.0 deduces several potential business behaviors (collaborative process cartography), the "design-time" decision support system and its associated KPI (to be used to select the appropriate business behavior, but also the appropriate technical behavior) and the "run-time" KPI (to support decision-makers to control "manually" the business and technical behaviors).

At the end of this second step, the user obtains (i) a set of "design-time" indicators defining expected performances, (ii) the adequate collaborative behavior to support the considered situation (collaborative processes selected among the deduced ones thanks to "design-time" KPI) and (iii) a set of "run-time" indicators (performance measurement system) to control this collaborative behavior during execution.

3.3 Implementation: Deployment of Mediation IS

Similarly to second step, in MISE 1.0 and particularly in MISE 2.0, the translation of collaborative workflows (from deduced collaborative processes) is a "binary" task:

semantic reconciliation (information/data and activities/services) select the most fitting technical elements to implement the deduced business collaborative behavior. In MISE 3.0, the idea is also to use non-functional requirements extracted from previously deduced “design-time” indicators during the semantic reconciliation step. By this way, the design of technical workflows (based on services and data) to implement business processes (based on activities and information) rests on functional and non-functional requirements. Concretely, instead of selecting technical services only on the basis of expected function (for instance “weather measurement”), non-functional requirements (such as *response time*, *reliability*, *security*, etc.) are also taken into account (for instance “weather measurement within 2s with encoded data).

3.4 Agility: Detection and Adaptation

This step is really based on the MISE 1.0 and particularly MISE 2.0 principles: detection through EDA system and adaptation through a new run of one of the design-time steps (function of the nature of the problem detected). But in the previous versions of MISE, the detection was based only on a comparison of models (current model differs from expected one). In MISE 3.0, we propose to add to this, a way that allows to the decision-maker to detect himself an abnormal situation through the use of the performance measurement system defined in step 2 (“run-time” KPI). Actually, the interpretation of such system is quite “human” and very difficult to automatize due to the interdependency between KPI. In other words, MISE 3.0 proposes a combination of automatic detection and human detection in order to improve responsiveness (and consequently agility) of the overall collaborative system.

3.5 MISE 3.0 Synthesis

According to the previous points, the third iteration of MISE provides improvements that may be summarized according to the following table:

Table 2. Specificities of the third iteration according to steps of MISE approach.

	MISE 1.0	MISE 2.0	MISE 3.0
Collaboration Model	*Domain specific metamodels have been defined, depending on considered business fields (crisis management [6], manufacturing context [7])*	*One generic metamodel, dedicated to all types of collaborative situations has been defined (including external layers, enclosing domain specific concepts)*	Based on an event-driven architecture one (or many) systems may be supervised in order (i) to detect any collaboration opportunity and (ii) to be immediately informed of all potential partners status (thanks to a continuous watching of the overall system)
Model of Collaborative behavior	*One single collaborative process has been deduced from the gathered knowledge.*	*Decisional, Operational and Support processes have been deduced from the gathered knowledge.*	Deducing several process cartographies (and associated sets of KPI) is a first improvement. Besides, associating a decision-support system (in order to assist the user in selecting the right one) is a second improvement.

Deployment of Mediation Information System	*After manual identification of technical services (or user-interfaces) that would assume identified business activities of the deduced collaborative process, the process is translated in BPEL language in order to be computerizable.*	*Automatic semantic reconciliation allows selecting subsets of technical services that will be invoked to assume business activities of collaborative processes on a technical point of view. Furthermore, ontological tools ensure "on-the-fly" data conversion [3].*	The main feature at this step is to include non-functional requirements in the semantic reconciliation step. Characteristics such as *reliability, latency* or *security* might then be taken into account in the workflow definition process in order to improve the quality of the selected technical services. Furthermore, decision-support system should also be integrated in that step in order to support efficiently the final selection.
Agility (detection + adaptation)	*Detection is a manual task based on the way situation evolves. Once detected a need of adaptation, design-time tools (model editor, process deducing tool, workflow translator) may be invoked on purpose in order to (re)define the collaborative behavior appropriate for the "new" situation.*	*Detection is based on an EDA. Sensors and services publish their events (reporting on the situation and on workflow progress) that can be used to update situational models. If the current model differs from the expected model, then adaptation must be started based on the same principle than MISE 1.0.*	The most important feature concerns the automated detection of evolution on the base of performance indicators (*i.e.* not only on the base of expected functions but also on the quality of these functions).

3.6 Application Domains

MISE project, is dedicated to provide a support framework for collaborative situation by deploying an agile mediation information among partners. Currently, there are mainly three application domains (but there might me really more): support of logistics systems, support of health care systems, support of crisis management systems. We can illustrate concretely the way MISE 3.0 might be used thanks to the last domain mentioned (crisis management): a geographical area may be watched through an EDA platform, in order to gather all events (from sensors, services, people, devices, etc.) in order to build and maintain a global picture of that area. According to some unexpected (or expected) negative changes (such as a lot of tweets mentioning the same problem, a lot of GPS data showing that a lot of vehicles are stopped, some abnormal values of temperature sensors, etc.), the MISE 3.0 platform could start the behavior deduction based on (i) information concerning the situation (risk, facts, etc.) and (ii) information concerning rescue means (resource, potential actors, etc.) both extracted from the global picture. Thanks to the implementation step a MIS may be deployed among the potential partners. Agility of this MIS could be performed thanks to models based on the global picture.

4 Conclusion

MISE project, through its three iterations provide a way to concretely connect *Internet of Things* (sensors, devices and any event providers) with *Internet of Knowledge* (ontologies and knowledge management systems) to run *Internet of Services* (technical services connected on the ESB). MISE principle is the following: any organization may be connected to the MIS, thus giving an access to its "public part" (mainly business capabilities and information). Thanks to EDA, all "public parts" of all connected organizations may publish events on the platform. Detecting any collaboration opportunity (thanks to events), the platform could push to potential partners a suggested collective business behavior (as an automatically deduced and selected collaborative process cartography). Once accepted or modified (through a dedicated decision support system), that collaborative behavior could be run onto the MIS (as an automatically generated set of workflows associated with a set of relevant KPI in charge of controlling the collaborative behavior) through orchestration and choreography. During that run-time, events (that are continuously sent to the EDA platform by invoked services and performance monitoring tool) update a permanent "picture" of the collaborative situation. That "picture" and KPI monitoring provide status knowledge useful to detect any adaptation need. If such a requirement appears, the orchestrated/choreographed workflows may be adapted on the fly by invoking design-time tools. The following picture illustrates this principle:

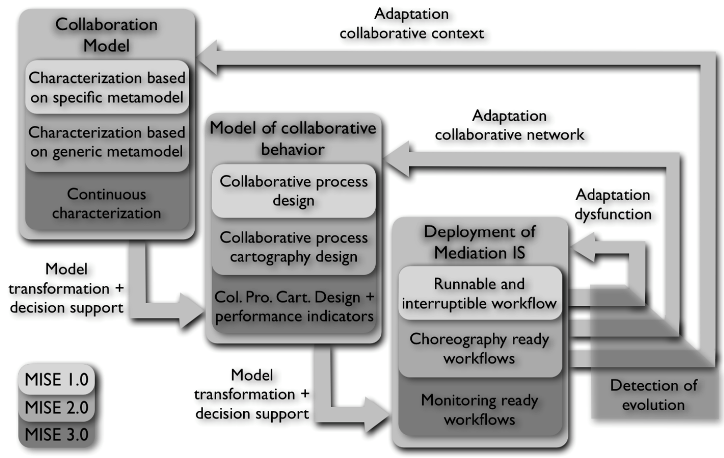

Fig. 2. MISE project overall structure including MISE 1.0, MISE 2.0 and MISE 3.0 iterations.

Similarly with figure 1, it is important to notice that there are in fact four "so-called" steps in MISE approach (whatever the selected iteration), however, the first three steps (dedicated to design-time) are presented as boxes while the last one (dedicated to run-time) is represented through three looping arrows.

5 References

1. Touzi, J., Bénaben, F., Pingaud, H., and Lorré, J.-P.: A Model-Driven approach for Collaborative Service-Oriented Architecture design. International Journal of Production Economics (IJPE), Elsevier, Volume 121, issue 1, pp. 5--20 (2009)
2. Rajsiri, V., Lorré, J.-P., Bénaben, F., and Pingaud, H.: Knowledge-based system for collaborative process specification. Computers in Industry (CII), Elsevier, Volume 61, issue 2, pp. 161—175 (2010)
3. Bénaben, F., Boissel-Dallier, N., Pingaud, H., Lorré, J.-P.: Semantic issues in model-driven management of IS interroperability. International Journal of Computer Integrated Manufacturing (IJCIM), Taylor&Francis. DOI:10.1080/0951192X.2012.684712 (2011)
4. Mu, W., Bénaben, F., Pingaud, H., Boissel-Dallier, N., Lorré, J.-P.: A model-driven BPM approach for SOA mediation information system design in a collaborative context. IEEE International Conference on Service Computing (SCC), IEEE, pp. 747--748. IEEE Press. Washington, USA (2011)
5. OMG, MDA Guide Version 1.0.1, omg/2003-06-01, (2003)
6. Truptil, S., Bénaben, F., and Pingaud, H. A Mediation Information System to help to coordinate the response of a crisis. Pro-VE'10, Springer. Saint-Etienne, France. (2010).
7. Rajsiri, V., Lorré, J.-P., Bénaben, F. and Pingaud, H. Collaborative process definition using an ontology-based approach. Pro-VE'08, Springer. Poznan, Poland. (2008).
9. Rongié, C., Lauras, M., Galasso, F., Gourc, D., Towards a Crisis Performance Measurement System. Proceedings of Pro-VE'10, Springer. Saint-Etienne, France. (2010).
10. NF EN ISO 9000 X50-130 Systèmes de management de la qualité. (2005).
11. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications Company (2010).
12. Truptil, S., Barthe, A.-M., Bénaben, F., Stuehmer, R.: Nuclear Crisis use-case management in an event-driven architecture. In edBPM workshop in conjunction with BPM conference. Clermont-Ferrand, France. (2011).
13. Lauras, M., Galasso, F., Rongier, C., Gourc, D., Ducq Y. Towards a more effective interoperable solution through an a priori performance measurement system. Proceedings of PRO-VE'11, Springer, IFIP. Sao Paulo, Brasil. (2011).