# A Design-time Data-centric Maturity Model for Assessing Resilience in Multi-Party Business Processes

Andrea Marrella[a], Massimo Mecella[a], Barbara Pernici[b], Pierluigi Plebani[b]

[a] *Sapienza Università di Roma*
*Dipartimento di Ingegneria Informatica Automatica e Gestionale*
*via Ariosto 25, 00185 Roma, Italy*
[b] *Politecnico di Milano*
*Dipartimento di Elettronica, Informazione e Bioingegneria*
*piazza Leonardo da Vinci, 32, 20133 Milan, Italy*

**Abstract**

Nowadays, every business organization operates in ecosystems and cooperation is mandatory. If, on the one hand, this increases the opportunities for the involved organizations, on the other hand, every business partner is a potential source of failures with impacts on the entire ecosystem. To avoid that these failures, which are local to one of the organizations, would block the whole cooperation, *resilience* is a feature that multi-party business processes currently support at *run-time*, to cope with unplanned situations caused by those failures.

In this work, we consider awareness of resilience in multi-party business processes during *design-time*, by focusing on the role of available – as an alternative to unreliable – data as a resource for increasing resiliency, as data exchange usually drives the cooperation among the parties. In fact, a proper analysis of involved data allows the process designer to identify (possible) failures, their impact, and thus improve the process model at the outset. A maturity model for resilience awareness is proposed, based on a modeling notation extending OMG CMMN – Case Management Model and Notation, and it is organized in different resiliency levels, which allow designers *(i)* to model at an increasing degree of detail how data and milestones should be defined in order to have *resilient by-design* process models and *(ii)* to quantify the *distance* between a process model and the complete achievement of a resiliency level.

*Keywords:* Business process resilience, artifact-centric modeling, resilience maturity model, CMMN - Case Management Model and Notation

*Email addresses:* marrella@diag.uniroma1.it (Andrea Marrella), mecella@diag.uniroma1.it (Massimo Mecella), barbara.pernici@polimi.it (Barbara Pernici), pierluigi.plebani@polimi.it (Pierluigi Plebani)

## 1. Introduction

The adoption of service-oriented architectures and workflow automation (a.k.a. orchestration), while enabling and making integration among heterogeneous systems easier, has also reduced the difficulties in digitizing the communications among different organizations. As a result, digital business ecosystems have been proposed as a paradigm for enabling cooperation among these organizations [23]; they can be conceptualized in terms of *multi-party business processes*: every party performs some internal tasks (private view) and communicates with the other parties if some information is needed to perform the internal tasks or if some results have to be notified to make the other parties able to perform their own tasks (external view, also referred to as choreography). Although this communication is a great opportunity for organizations, the resulting inter-dependencies are difficult to manage, especially when some failures occur: a party could stop working for internal reasons and all the parties which depend on the information that the failing one is responsible for might fail as well, thus resulting in a domino effect.

A proper design of *resilient business processes* becomes fundamental. *Resilience concerns the ability of a system to cope with unplanned situations in order to keep carrying out its mission* [5]. In particular, making a multi-party business process resilient means helping organizations to cope with the complexity of the process and avoiding, limiting or mitigating possible failures that might affect the technological infrastructures as well as the involved organizational structures [2]. We therefore consider the *resilience* of the process as a whole, including its informational components that depend on the infrastructural and physical elements.

Satisfying resilience requirements is related to the ability to cope with *unplanned situations*. In the literature [30], several approaches have been proposed to keep business processes running even when some unplanned exceptions occur, by enacting countermeasures. If we focus on *what to do* in case of failures, focusing on run-time flexibility with a reactive approach seems to be the only possibility. Actually, new opportunities come from the recent increase in available data that, in some cases, can be used as alternative sources of data for performing tasks that, otherwise, could fail. Thus, we can shift the focus to *what may be affected* when a failure occurs, where improvements can be made also at *design time*, with the aim of assessing the *level of awareness* with respect to the resilience of the processes while they are being designed.

The goal of this paper is to provide a systematic approach for evaluating the resilience of multi-party business processes, and driving the improvement of resilience by reducing the possible impact of failures caused by missing data due to improper human behavior and/or smart device errors. To this aim, the approach considers the dependency between data and tasks, as well as data and milestones characterizing a process. In particular, our approach is based on assessing how available data or milestone redefinitions can possibly be exploited to design viable alternatives in the process model to make it more resilient. The adoption of a declarative notation, namely OMG CMMN – Case Management

Model Notation [26], to design the process model, introduces an additional degree of flexibility as declarative languages rely on an open-world assumption, thus leaving room for supporting situations that cannot be planned at design-time [9].

As an extension of the work presented in [29], this work introduces a formalization of the approach where data are considered as "first class citizens", because their unavailability could determine the failure of the processes. The proposed formalization includes an extension of OMG CMMN with some additional elements required to express the alternative data and milestones.

The resulting *maturity model*, which constitutes a significant contribution of this paper, takes into account the degree of awareness of process models through *levels of resilience*, which can be computed using the provided formalization. For any resiliency level, we define an indicator that quantifies the percentage of compliance of a process model with respect to the resiliency level of interest, taking into account the different values of criticality of the modeling elements considered as relevant for the resiliency of the model. When a full compliance with a resiliency level is not completely achieved, the indicator allows a better understanding of the impact and the risks of such a non-compliance, returning a percentage value that implicitly measures the distance between the model and the complete achievement of the resiliency level.

The rest of the paper is organized as follows. After a discussion of the state of the art in Section 2, Section 3 introduces a motivating case study – used all the way through the paper – from which resilience requirements are derived. In Section 4 we summarize the CMMN notation that we adopt as a basis for process modeling. Section 5 introduces the proposed maturity model, where five levels of resilience awareness are introduced. Section 6 defines how CMMN has been extended to define process models that can be coupled with the proposed maturity model, in order to identify to which resiliency levels they belong. In Section 7, we present a critical discussion about the general applicability, strengths and limitations of our maturity model, and finally, in Section 8, we conclude the paper by tracing future work.

## 2. Related work

Research on *resilient systems* encompasses several disciplines, such as psychology [38], ecology [11], sociology [1] and engineering [16]. In information systems, *resilience engineering* has its roots in the study of safety-critical systems [16], i.e., systems aimed at ensuring that organizations operating in turbulent and interconnected settings attain high levels of safety despite a multitude of emerging risks, complex tasks, and constantly increasing pressures. A system is considered as resilient if its capabilities can be adapted to new organizational requirements and changes that have not been explicitly incorporated into the design of the existing system [22]. In the BPM field, cf. [22] and [31], this means that respective business processes are able to automatically adapt themselves to such changes.

Over the last years, change management in BPM has been mainly tackled through the notions of *process flexibility* [30] and *risk-aware BPM* [34, 33].

On the one hand, the themes of flexibility and flexibility requirements have been discussed widely in the literature (e.g., see [24] for a summary), as a requirement for enterprise systems of being robust to business changes. Research on process flexibility has focused on four major flexibility needs, namely *(i) variability* [12, 13], *(ii) looseness* [37, 19], *(iii) adaptation* [32, 21], and *(iv) evolution* [6, 7].

- *Variability* requires to incorporate in a process model different *process variants* at design-time such that the selection of the most appropriate variant can be done at run-time for each process instance [13], i.e., the course of actions may vary from variant to variant [12]. Usually, there exists a multitude of variants of a particular process model, whereby each of these variants is valid in a specific scenario; i.e., the configuration of a specific variant depends on requirements of the process context [13].

- *Looseness* is the ability of a process to execute on the basis of a loosely or partially specified model. The full specification of the model may emerge at run-time, is not known a-priori and may be unique to each instance [37]. In this direction, declarative approaches, such as the constraint-based language Declare [28], are emerging with the aim of providing a less restricting way for modeling processes, i.e., anything is possible as long as it is not forbidden [37].

- *Adaptation* relates to the ability of a process to react to exceptional circumstances and to adapt/modify its structure accordingly [32]. Exceptions can be either *anticipated*, i.e., planned at design time and incorporated into the process model, or *unanticipated*. The latter refer to situations, unplanned at design time, that may emerge at run-time and can be detected and manually or automatically tackled in ad-hoc way only during the execution of a process instance, when a mismatch between the computerized version of the process and the corresponding real-world process occurs [21].

- *Evolution* is the ability of an implemented process to change when the corresponding business process evolves [6]. The evolution may be *incremental* as for process improvements [7] (i.e., only small changes are required to the implemented process), or *drastic* as for process innovation or process re-engineering [15]. (i.e., if radical changes are required).

The ability to deal with changes makes process flexibility approaches a required, but not sufficient, means for building resilient BPM systems. In fact, there is a (seemingly insignificant but) relevant gap between the concepts of flexibility and resilience: *(i)* process flexibility is aimed at producing "reactive" approaches that reduce failures from the outset by incorporating remedial strategies at design-time or deal with them at run time if any "known" disturbance arises; *(ii)* process resilience requires "proactive" techniques accepting

and managing change "on-the-fly" rather than anticipating it, in order to allow a system to address new emerging and unforeseeable changes with the potential to cascade. On the other hand, while relatively close to the concept of risk-aware BPM, which evaluates operational risks on the basis of historical threat probabilities (with a focus on the "cause" of disturbances and events), resilient BPM shifts attention to the "realized risks" and their consequences, to improve risk prevention and mitigation, and therefore it aims at complementing conventional risk-aware approaches.

Based on the foregoing discussion, there is only a limited number of research papers investigating the resilience of BPM systems [2, 40, 39], and mainly at conceptual level. For example, the work of Antunes and Mourao [2] derives a set of fundamental requirements aimed at supporting resilient BPM. More recently, the approach of Zahoransky et al. [40] investigates the use of process mining [36] to create probability distributions on the time behavior of business processes. Such distributions can be used as indicators to monitor the level of resilience at run time and indicate possible countermeasures if the level drops. Finally, the work [39] provides a support framework and a set of measures based on the analysis of previous process executions to realize and evaluate resilience in the BPM context.

Closely related to process resilience approaches, it is worth to mention the work [24], which advocates a "flexibility by selection" approach for a priori modeling of the capability of environment change based on the dynamic selection of components from libraries. Along this line, many papers have been advocating modeling dynamic process composition using services as components (see for instance [3] presenting a general approach). However, in these proposals the focus is on modeling the process structure rather than the data used in the range of the process, which is the essential component that is used to tackle resiliency in our approach.

If compared with the aforementioned papers, our research aims to provide concrete indicators to measure the resilience of a multi-party business process by focusing on the data exchanged between the activities composing the process, an aspect neglected in the existing approaches to process resilience. We believe that such indicators can provide a reliable mean for evaluating in advance the impacts of potential disturbances and improving decision making at run time.

## 3. Case study and requirements

Based on a case study taken from a real scenario, this section motivates the presented approach for supporting the resilience at design-time by discussing problem setting and deriving the requirements for resilience that will be analyzed in the rest of the paper.

### *3.1. Description of the case study*

Smart devices have been adopted by several organizations to increase the effectiveness of business processes [18]. For instance, in the logistics domain,

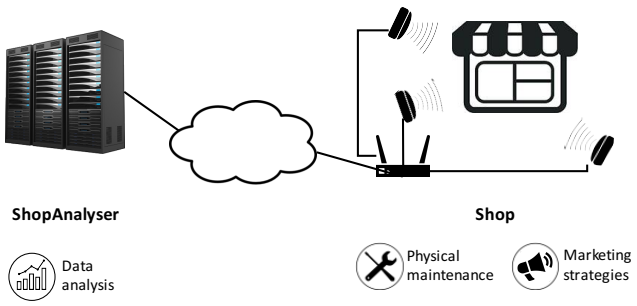



Figure 1: Running example overview.

smart devices provide real-time monitoring of goods transportation in terms of their position or state (e.g., temperature, humidity). Although the advantages of the adoption of smart devices are clear, there are also some side effects in terms of system reliability. In fact, smart devices are prone to failure due to their limitations in terms of computational power and energy autonomy. Moreover, in some cases they operate in extreme conditions (e.g., meteorological stations on top of mountains), thus they might stop working without any previous notice.

Implications of the use of sensors in processes are illustrated through the example shown in Figure 1, which presents a real case study involving the company *ShopAnalyser* and *Shop Inc.*, one of its clients. ShopAnalyser offers products and services to physical shops/commercial centers willing to monitor and analyze the behavior of their customers while they are walking inside their premises. To this aim, ShopAnalyser sells innovative sensors able to capture the probe packets periodically sent by cellphones and to localize and track the position of cellphones. In this way, assuming that a cellphone belongs to exactly one customer, the sensor is able to track the behavior of the customer inside the area and, by correlating MAC addresses, it recognizes when the same customer repeatedly visits the shop. The analytics required to understand the customers' behaviors are offered by ShopAnalyser as a service to all the shops which buy its sensors. More specifically, ShopAnalyser produces one report every week to the shops, and they use these reports as a basis for defining or improving their marketing strategies.

Shop Inc. has the goal of reaching an acceptable conversion rate[1] in its shop, and therefore it decides to acquire sensors and the analytics service from ShopAnalyser to get an insight on its customers' behavior and support its marketing analysis. The owner of Shop Inc., through its maintenance personnel, is responsible for the installation and physical maintenance of the sensors: ShopAnalyser delivers the sensors to Shop Inc., which installs them in the shop and configures them to send collected data to the data center of ShopAnalyser. Some status

[1]In marketing, the conversion rate measures the ratio between visitors and effectively paying customers.

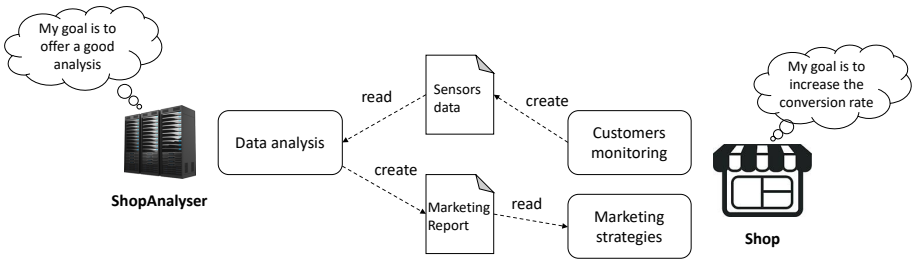



Figure 2: Problem setting.

LEDs are embedded in the sensors to make the owners of the shops aware of possible malfunctioning problems in sensor behavior, when the probe packets sent by the cellphones are not collected correctly (in this case the Shop informs ShopAnalyser, which will enact some repair action, such as sending substituting sensors), or to signal connection problems (i.e., the sensors are working, but the data cannot be sent to ShopAnalyser). ShopAnalyser is responsible for the data analysis, which produces a weekly report, and for identifying malfunctioning sensors that cannot be detected directly by the shops, i.e., unrealistic data captured by sensors and sent to the data center (e.g., one hundred cellphones identified in the same tiny shop at the same time).

Although some actions are present to cope with sensor malfunctioning, in the case study the focus is mainly on signaling possible failures: i.e., if a sensor stops working then a replacement is provided; if the network connection is interrupted, then the ISP – Internet Service Provider – is called to resume the connection. Actually, these occurring failures could have a significant impact as they affect the data availability. In fact, during down time, an amount of sensors data is not collected so it is not represented in the data set used for the analysis. As a consequence, the report used for marketing purposes might become unrealistic.

However, during the process enactment, several other unplanned situations may occur. Depending on the nature of the raised issues, the magnitude of their impact varies and one or many activities may be involved. At the same time, different countermeasures can be taken to mitigate these negative effects. In fact, there may be other data available in the shop and its environment that could be exploited to improve the quality of the service.

As an example, the sensors might not be able to communicate with ShopAnalyser for many reasons, an alternative source of information about the number of clients in the shops might be considered, to be able to infer customers' behaviors in the shops. Alternative ways to collect such information can include counting the number of persons entering the shops, which may be available from other unrelated applications, such as video surveillance. In this way, ShopAnalyser will not have gaps in the analysis, but only lower data quality. Other ways to improve the final reports could include algorithms to fill in the gaps of sensor information, based for instance on sales prediction algorithms applied when sensors data have not been collected.

### *3.2. Problem setting*

In order to set the boundaries of our problem, we define a multi-party business process for the running case study in terms of (see Figure 2):

- *Parties*: players involved in the process. All the parties are interested in making the process up and running without problems, as their goals depend on the resilience of the whole process. As an example, Shop Inc. wants to make the marketing strategy more effective by increasing the conversion rate. On the other hand, ShopAnalyser wants to sell a good service to its customers.

- *Tasks*: a task is a unit of work performed by a party, which consumes data as input and produces data as output. The data produced by a task must be needed by at least one other party. As we are more interested in the dependencies among the parties – rather than to internal executions of processes by each party – the Customers Monitoring, Data Analysis, and Marketing strategies are the tasks considered.

- *Data*: units of storage used by the data producer to store/write data and by the data consumer to read such data. Producers and consumers are parties performing tasks and, in the case study, "sensors data" as well as "marketing report" are the relevant data.

- *Milestones*: status of the process for which the verification of some properties ensures a correct evolution of the process. The final milestone represents the goal of the process as no further evolution is required. Different parties could have different milestones. For instance, the ShopAnalyser wants to offer a good sensors data analysis, whereas the Shop wants to increase its visitors conversion rate. While these milestones represent the final goal of the parties, the ability to produce sensors data, as well as the ability to create a report are other milestone that are requested to be reached while executing the process.

### *3.3. Resilience requirements*

Our approach analyzes the multi-party business process resilience from a *data perspective*: data dependencies among the involved parties and relationships between process tasks/milestones and data are taken into account to identify the *sources of possible failures*, and how the process can be better modeled to make it resilient with respect to these failures.

Similarly to what is usually done in emergency management [35, 17, 20, 27], where a *preparedness phase* aims to improve the system by learning from the previous emergencies, we propose an approach that helps the process designers improve their process models by considering the previous experiences in failures generated by data unavailability. The resilience of this type of process depends both on the reliability of the tasks and on the lack of data availability.

The *reliability of the tasks* concerns the possibility that one or more tasks cannot be executed: i.e., the infrastructure required to perform the job is not

available, also including the human resources whose execution of manual tasks can be blocked by data unavailability, is not available.

On the other hand, *lack of data availability* is a situation in which the data consumed by a task are not available. This situation can occur for different reasons:

- Firstly, it may be directly connected to task reliability, as data produced by parties performing tasks may be relevant for that party or other participating parties, and problems on tasks may also have the side effect of making data unavailable.
- Moreover, there are situations in which tasks work properly, but the returned data, although available, do not have a sufficient quality level to enable processing, thus they can be considered unavailable. Completeness, timeliness, and accuracy are some of the quality parameters through which we can define the acceptable level of data quality for considering the data available [4]. For this reason, the definition of the data could be coupled with the definition of quality levels that are considered acceptable for a task uses said data.

Based on this discussion and on the experience gained in the area and lessons learned from our direct involvement in several projects dealing with multi-party business processes (a partial list of such projects is detailed in the Acknowledgments), we can outline the following set of requirements to be satisfied in order to model a resilient process.

**Req #1** A strong requirement for multi-party processes is to explicitly model all relevant data accessed by process tasks and manipulated by them. The unavailability of such data often represents the main reason of failure during process execution, and this may critically affect the overall resilience of the multi-party process.

**Req #2** It is required to represent the availability of alternative sources to primary data. When devising alternatives, focusing only on the control flow is limiting, as processes are often constrained in terms of tasks to be executed and their flow. Conversely, data, which may originate from different sources and may have different quality, are often a practical source of alternatives (as previously discussed in the case study).

**Req #3** It is required to represent alternatives for milestones, where milestones are meant as major intermediate objectives that the process aims at achieving. Sometimes, in particular when alternative data sources are employed in place of primary ones, it is not possible to achieve all the milestones expected for a process. Thus, the ability to model alternative milestones that capture best-effort (relaxed) objectives – still maintaining the meaningfulness of the whole process – is crucial for modeling resilient processes at design time.

**Req #4** It is required to provide a mechanism to increase the process resilience at the modeling level, in order to drive the designer towards modeling resilient-by design processes. Designing a model is an iterative task, in which the designer progresses and continuously takes awareness of the different facets of the phenomenon to be modeled (in our case, the process). Therefore, a practical approach should allow the designer to check iteratively the compliance of the phenomenon with a specific aspect (in our case, the modeling of the resilience) in order to gradually increase its awareness.

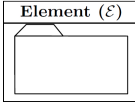
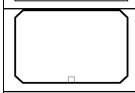
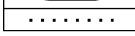
**Req #5** It is required to objectively quantify at design-time the level of modeled resilience, in order to provide an effective tool that allows to detect how much the designer is progressing in the level of awareness of the phenomenon (in our case, the resilient-by-design process). This will allow to measure how far is the design of a process model from achieving an adequate level of resiliency. The presence of a clear indicator of resiliency can be also seen as a way to push the designer to continuously improve the modeling of a resilient-aware process. As a matter of fact, similar indicators that capture the unlocking of achievements are massively used in gamification theories [14] to improve engagement and productivity of end users.

Starting from the above requirements, in the following sections we will provide a modeling approach, based on CMMN, which will allow a designer to specify resilient-by-design processes and assess them against a maturity model structured in five levels of resiliency to be achieved.

## 4. A CMMN primer

To model multi-party business processes, like the one in the case study, activity-centric modeling languages such as BPMN are usually adopted. Even if this type of language results as being more intuitive for the process designers, this approach has some limitations with reference to specifying process resilience. As an example, the order of activities during exception handling is loosely specified: when addressing process resilience, the designer should specify recovery activities, and the order in which they are performed is usually decided at run-time based on considerations about the status of the process. Other approaches, such as declarative modeling, relax some strictly sequencing assumptions, thus leaving room for supporting situations that cannot be planned at design-time [9].

To this purpose, as already discussed in Section 1, in this work we adopt CMMN – Case Management Model and Notation [26]. Being an artifact-based language, the definition of data, and the relationship with tasks and milestones holds a central role in the modeling, making this notation suitable for our objectives.

| Element ($\mathcal{E}$) | Name |
|---|---|
|  | case |
|  | stage |
|  | case file item |
|  | task |
|  | event listener |
|  | milestone |
|  | connector (sentry) |

(a) Key modeling elements

| Type | Annotator ($\mathcal{A}$) | Name |
|---|---|---|
| Decorator |  | collapsed planning table |
| |  | expanded planning table |
| |  | auto complete |
| |  | collapsed |
| |  | expanded |
| |  | manual activation |
| | # | repetition |
| | ! | required |
| Sentry |  | entry criterion |
| |  | exit criterion |
| Marker |  | non-blocking human |
| |  | process |
| |  | case |
| |  | participant |
| |  | timer |

(b) Annotators

Figure 3: CMMN modeling elements and annotators.

CMMN was published in 2014 by the Object Management Group (OMG), with the target of providing a complementary specification to BPMN – Business Process Model and Notation [25]. While BPMN is focused on designing a business process in a *procedural way* with the help of an explicit control flow (which specifies *how* and *in which order* things must happen in a process), CMMN provides a *declarative style* for modeling processes that is targeted at describing *what* is allowed and not allowed in a process. In this section, we briefly introduce in a rigorous way the key concepts of CMMN that are required to understand the rest of the paper. For a complete description of the standard, as well as for a conceptual representation (through Class Diagrams) of its modeling elements and relationships, the reader should refer to the official specification document [26].

**Definition 1** (CMMN model)**.** A CMMN model is a tuple $\mathcal{N} = \langle \mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R} \rangle$ where:

- $\mathcal{E}$ is a set of modeling elements;
- $\mathcal{U}$ is a binary relationship in which two elements $x, y \in \mathcal{E}$ (with $x \neq y$) are related if and only if they belong to the same scope;
- $\mathcal{A}$ is a set of annotators that can be used to indicate the characteristics of elements in $\mathcal{E}$;
- $\mathcal{R}$ is a binary relationship $\langle x, y \rangle$ in which two elements $x, y \in \mathcal{E}$ (with $x \neq y$) are related if and only if an event ($e$) from one ($x$) triggers the other ($y$), i.e., $x \overset{e}{\Rightarrow} y$. Note that $x$ and $y$ must be in the same scope, i.e., $\langle x, y \rangle \in \mathcal{U}$, and that $\langle x, y \rangle \in \mathcal{R}$ is an ordered pair.

As shown in Figure 3a, the set of modeling elements $\mathcal{E} = \mathcal{E}_{\mathcal{C}} \cup \mathcal{E}_{\mathcal{S}} \cup \mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{M}} \cup \mathcal{E}_{\mathcal{D}} \cup \mathcal{E}_{\mathcal{V}} \cup \mathcal{E}_{\mathcal{F}}$ includes:

- A set $\mathcal{E}_{\mathcal{C}}$ of *cases*. A case is a container for all the elements and data of a CMMN model. Note that a CMMN model may have multiple cases, but a case cannot be contained by other modeling elements.
- A set $\mathcal{E}_{\mathcal{S}}$ of *stages*, which are containers for modeling elements and may be organized in a hierarchy (i.e., a stage may contain other stages).
- A set $\mathcal{E}_{\mathcal{D}}$ of *case file items*, which are used to represent all kinds of data involved in the execution of a case, including documents, data values in a database, spreadsheets, etc.
- A set $\mathcal{E}_{\mathcal{T}}$ of *tasks*, i.e., units of business relevant work that are to be performed within a case.
- A set $\mathcal{E}_{\mathcal{V}}$ of *event listeners*, which describe the interaction between case elements and the external environment.
- A set $\mathcal{E}_{\mathcal{M}}$ of *milestones*, which represent accomplishments of business objectives during the execution of a case.
- A set $\mathcal{E}_{\mathcal{F}}$ of *connectors*, which are primarily used to visualize the relationships defined in $\mathcal{R}$ and can be annotated with a label representing which specific events have determined them.

Notice that stages and tasks can be flagged as *discretionary* (from a graphical point of view, their shape is obtained using a dashed line), meaning that the decision to deal (or not to deal) with them is deferred to run time.

Figure 4 presents the CMMN model of the ShopAnalyser case study. The outer box *Shop improvement* represents the *case*, i.e., the complete behavior of the process. A user is provided with access to all information concerning the case and is responsible for controlling how a case evolves. According to the model, the following elements can be identified:

$$
\begin{aligned}
\mathcal{E}_{\mathcal{C}} &= \{\textit{Shop improvement}\} \\
\mathcal{E}_{\mathcal{S}} &= \{\textit{Sensor data acquisition, Data analysis,} \\
&\qquad \textit{Marketing analysis}\} \\
\mathcal{E}_{\mathcal{T}} &= \{\textit{Installing sensors, Reading values,} \\
&\qquad \textit{Data mining, Marketing actions}\} \\
\mathcal{E}_{\mathcal{M}} &= \{\textit{report available}\} \\
\mathcal{E}_{\mathcal{D}} &= \{\textit{sensors data, shop data, marketing report, report}\} \\
\mathcal{E}_{\mathcal{V}} &= \{\textit{on Monday, acceptable conversion rate}\} \\
\mathcal{E}_{\mathcal{F}} &= \{\langle \textit{Data analysis}, \texttt{occur}, \textit{on Monday}\rangle \\
&\qquad \langle \textit{Data analysis}, \texttt{-}, \textit{report available}\rangle \\
&\qquad \langle \textit{Installing sensors}, \texttt{complete}, \textit{Reading values}\rangle \\
&\qquad \langle \textit{Shop improvement, } \texttt{occur}\textit{, acceptable conversion rate}\rangle\}.
\end{aligned}
$$

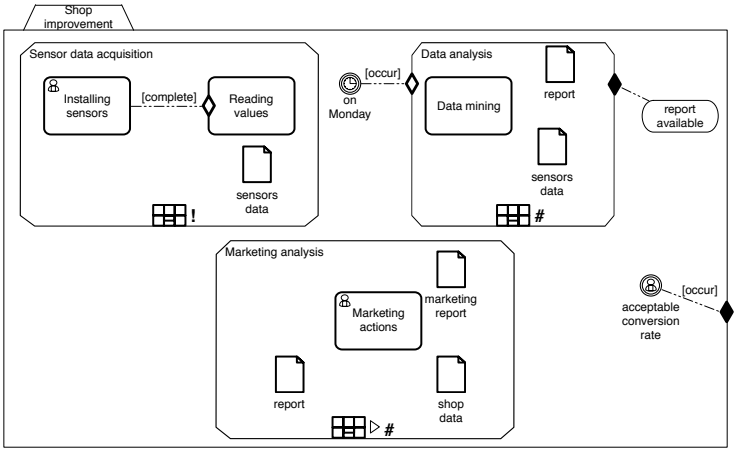



Figure 4: CMMN diagram of the case study process.

Due to the declarative nature of CMMN, it is possible to define strong *scope relationships* between its elements. The scope relationship $\mathcal{U}$ can be imposed by using the two kinds of modeling element that can contain other elements, i.e., cases and stages. In a nutshell, a modeling element $x$ is in a scope relationship with an element $y \neq x$, if and only if they are contained in the same case or stage, and no other sub-stage contains them. For example, a portion of the scope relationship of the CMMN model in Figure 4 is the following:

$$\begin{aligned}\mathcal{U} = \{&\ldots,\\ &\langle \textit{Sensor data acquisition}, \textit{Data analysis}\rangle,\\ &\langle \textit{Sensor data acquisition}, \textit{Marketing analysis}\rangle,\\ &\langle \textit{Data analysis}, \textit{Marketing analysis}\rangle,\\ &\langle \textit{Sensor data acquisition}, \textit{on Monday}\rangle,\\ &\langle \textit{Sensor data acquisition}, \textit{report available}\rangle,\\ &\langle \textit{Sensor data acquisition}, \textit{acceptable conversion rate}\rangle,\\ &\ldots,\\ &\langle \textit{Data mining}, \textit{sensors data}\rangle,\\ &\ldots\}\end{aligned}$$

i.e., the three stages of the model are in the same scope, together with the milestone *report available* and the events *on Monday* and *acceptable conversion rate*. Note that the task *Data mining* and the case file item *sensors data* are in a different scope than, e.g., *on Monday*. This means that $\langle \textit{Data mining}, \textit{sensors data}\rangle \in \mathcal{U}$, but $\langle \textit{Data mining}, \textit{on Monday}\rangle \notin \mathcal{U}$ and $\langle \textit{sensors data}, \textit{on Monday}\rangle \notin \mathcal{U}$.

Concerning the annotators, there are three types, named decorators, markers and sentries (cf. Figure 3b). *Decorators* are used to indicate certain characteristics of a case/stage/task. For example, the decorators attached to the *Market-*

*ing analysis* stage in Figure 4 indicate that: *(i)* it must be manually initiated (*manual activation* decorator); *(ii)* it can be repeated multiple times (*repetition* decorator); *(iii)* it is associated to a list of discretionary items that can be dynamically selectable at run-time (*expanded planning table* decorator). *Markers* denote the kind of tasks to be executed within a case. For example, *Installing sensors* is a human task, i.e., a task that must be executed and completed by a (human) case worker.

Among the annotators, *sentries* are particularly important as they allow the definition of temporal-logic dependencies between modeling elements. To be more specific, sentries enable in order to describe when a task, stage, or milestone is available for execution (*entry criterion*), or when a case, stage or task is complete (*exit criterion*). As an example, the *Reading values* task starts only when the sensors have been installed. The *Data analysis* stage opens every week and terminates when a report is produced as defined by the milestone *report available*. Entry/Exit criteria have the form: `on e if cond`, where `e` is an event and `cond` is a condition over data. Both parts are optional, supporting both pure event-based or conditional-based sentries.

The use of sentries allows the specification of *event relationships* between modeling elements. Specifically, a relationship $\langle x, y\rangle \in \mathcal{R}$ relates two modeling elements $x$ and $y$ if and only if an event from one of them triggers an entry or an exit criterion in the other element. For example, in the CMMN model of Figure 4, the event relationship $\langle$*on Monday*, *Data analysis*$\rangle \in \mathcal{R}$ indicates that every Monday the entry criterion of the stage *Data analysis* is triggered, causing its execution. Similarly, the event relationship $\langle$*Shop improvement*, *acceptable conversion rate*$\rangle \in \mathcal{R}$ is used to denote that when an adequate conversion rate is achieved, then the case *Shop improvement* completes. Notice that event relationships cannot cross scope, i.e., if $\langle x, y\rangle \in \mathcal{R}$, then $\langle x, y\rangle \in \mathcal{U}$.

Once the conversion rate obtained by executing all the activities is considered sufficient, then the business process concludes. Finally, *case plan items* (i.e., *sensors data*, *report*, *marketing report* and *shop data*) are included in the stages which use them. It is worth noting that, according to the reported diagram, from the moment when the sensors are installed, the *Reading values* task keeps running until the time in which the expected conversion rate is achieved. At the same time, the marketing analysis is not coordinated with the other activities as it is performed by analyzing the reports produced by ShopAnalyser.

This concludes the exposition of the key concpets underlying the design of a CMMN model. As will be discussed in the following sections, also this language has some limitations when defining the data aspects. For this reason, we will propose some extensions to the language to better represent all the data aspects and dependencies required to concretely employ our maturity model.

## 5. A maturity model for the awareness of resilience

With the aim to classify multi-party business processes in terms of their degree of resilience *awareness*, one of the main contribution of this paper is a

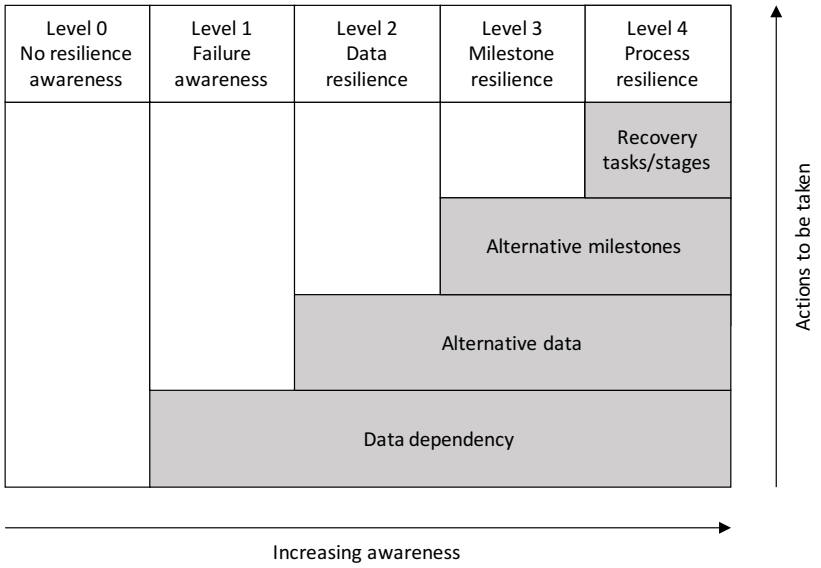



Figure 5: Process resilience awareness maturity model.

maturity model which organizes different levels of resilience awareness, shown in Figure 5, in a coherent framework where the actions to be taken in order to increase the by-design resilience are identified. The levels of resilience are defined on the basis of the ability of the multi-party process to adjust the possible unexpected failures with preparedness strategies developed to increase resilience at design time.

By using the maturity model, the process designer can figure out how much the resilience of the process under investigation has been taken into account. Based on this, having a process classified as *Level 0* means that resilience has not been considered at all in the design, while a process in *Level 4* indicates that the designers considered resilience fully according to several aspects, as detailed in the following. Notably, the proposed maturity model does not provide a specific support for investigating resilience during run-time, as this depends on the availability of some resilience support (e.g., fault-tolerant systems) embedded in the infrastructure that the process relies on, or on being the activities internally designed to be resilient, etc. These are orthogonal aspects and not necessarily seen by the designer at design-time.

As discussed in Section 2, other proposals in the literature have been put forward to define resilience for processes, e.g., [8]. However, here we do not focus on the structure of the process or its components and instances, but we aim at classifying the way resilience can be considered and obtained, in terms of preparedness to unexpected events which might be caused or have an impact on data availability. In particular, the following levels of designed resilience have been identified:

- **Level 0 – No resilience awareness.** At this level, a business process is designed without taking into account the data unavailability that might cause failures during the execution. As a result, countermeasures to be adopted in case of critical situations are not defined. The designed process only reflects the wishful scenario where it is assumed that all the parties correctly execute their tasks and all the data are transferred among them as expected. Although a process design of this type can be useful for defining the agreement between the parties, no support is given to resilience.

- **Level 1 – Failure awareness.** A first step for improving the process design is to make the process aware that there are possible sources of failure, therefore there is the need to make it resilient. In this work, we consider failures caused by data unavailability, which might affect on one or many tasks of the same party that is producing such data, or tasks performed by other parties. For this reason, failure-aware business processes are designed to have a clear map of which relevant data are subject to failures, as well as the impact of these failures. The analysis of potential failures depends on several factors: amount of data, how the data are collected, how the data are stored. As an example, data stored on a local server have a probability of failure that is lower than data stored on a smart device connected to a wireless network. Similarly, if data created by one party and used by several parties becomes unavailable, the impact of this failure will be greater than the one produced by data created and consumed by the same party.

- **Level 2 – Data resilience.** The model of the process makes an initial attempt to overcome possible failures considering *data availability*. On the basis of the information about the sources of failures and the potential impacts of these failures, the designer can decide to include alternative data in the process model. In this way, starting from the data with a higher probability of failure and greater impact, the designer has to specify if there are alternative data sources and how to reach them. A more precise model requires an analysis of the gap between the quality of the data in the original data source with respect to the quality of the data in the alternative data source. For instance, if the sensors installed in Shop Inc. stop working, the process model indicates other services as an alternative source, e.g., an installed door counter and/or Google Popular Times or even historical data stored in a different, but accessible, place. The issue of data quality has been addressed extensively in traditional information systems, e.g., [4], but the quality of big data (which includes sensor-generated data) still has to be precisely defined [10].

- **Level 3 – Milestone resilience.** As the process resilience implies mitigating the effect of a failure, a possible mitigation includes revising the

initial expectations of the process to achieve a given milestone. At this level, resilience awareness implies that the designer defines, for each party, a new milestone that represents a status that can terminate process execution in a reasonable way. If the initial milestone corresponds to the optimal one, the alternative milestone could be considered as best-effort. As an example, ShopAnalyser, realizing that the data coming from the sensors contain errors, can decide to release an incomplete, lower quality report at a reduced price instead of releasing full reports with all details.

It is worth noting that the business process models at this level do not prescribe any specific remedial action to cope with the failures at run-time. For this reason, a model at this level only helps whoever is in charge of executing the process to select, in case of failures, new data sources as well as to decide on considering the result of the execution as satisfactory even if the initial objective implied by a primary milestone cannot be fulfilled, accepting a weaker objective.

- **Level 4 – Process resilience.** At this level, processes have been designed by considering also remedial actions to be taken in case of failures. Design-time mechanisms are conceived to be able to (semi)-automatically move the process to an acceptable state when unexpected or unplanned failures occur. Based on the information about the alternatives (both data and milestones), the designer can embed in the business process how these alternatives could be effectively managed. New recovery stages and tasks can be added to the process to express the activities to be performed in order to improve the quality of the data alternatives to a quality level that is equivalent to the original service. Taking as example the problems of missing data, the previous level suggests including the door counter and the Google Popular Times in the list of possible alternatives. At this level, the process designer should specify if the alternative data should be considered as they are produced, or if additional tasks must be taken, e.g., combining both services into a reliable assessment of the indoor occupancy for Shop Inc.

With the above levels of resilience, we aim at supporting the process designer in understanding whether resilience is modeled, and whether there is room to improve the process model by specifying possible alternative solutions. As an example, once the designer understands that the modeled processes are at Level 0, the first step should be to start considering the evolution of the data in the process (i.e., moving to Level 1 – Failure awareness).

## 6. Modeling and assessing resilience

In this section, for each previously introduced level, we discuss the practical impact of using CMMN as a modeling language. In this way, on the one hand, we are able to highlight which additional required constructs are needed for

modeling resiliency aspects. On the other hand, this will allow us to properly address the requirements discussed in Section 3.3

Therefore, we propose an extension of CMMN able to improve the specification of which data are used and in which way, in order to better analyze the possible failures and their impacts. Such an extension will be specified in a rigorous formal way. This will allow us to provide a non-ambiguous classification framework to be used for checking if a CMMN model is compliant with one of the resiliency levels defined in the previous section.

### *6.1. Level 0 – No resilience awareness*

CMMN makes it possible to express the basic scenario where resilience is not considered at all. The model of the business process for the ShopAnalyser case study, shown in Figure 4, belongs to this level.

### *6.2. Level 1 – Failure awareness*

One of the main shortcomings of CMMN is poor data modeling capability about data. In the current version of the standard [26], data are defined in an "abstract way", in terms of *case file items* with no restrictions in the format and nature of the represented data. If, on the one hand, this allows maximum flexibility in modeling various scenarios, on the other hand, no information about the link between tasks/event listeners and data is provided, unless data are attached to the entry and exit conditions of a modeling element. This means that the existence of case file items is limited to triggering the enactment of a stage/task/milestone, or to recording the outcome of case/stage/task execution through an exit criterion. Any link connecting a case file item to the sentry of another modeling element can be labeled with the specific operation (i.e., *create*, *update*, *delete*, *replace*) to be performed on the case file item; such operations are presented in the CMMN standard in Table 8.2 [26]. In the rest of the paper, we denote them as $OP$. To date, as a matter of fact, the CMMN standard does not allow the expression of any other kind of data dependency in the range of a case.

To overcome this limitation and allow the design of CMMN models that are compliant with Level 1 of the maturity model, we propose to extend CMMN through a more rigorous specification of case file items.

**Definition 2** (Case File Item)**.** A case file item is a tuple $\mathcal{D} = (nm(\mathcal{D}), ds(\mathcal{D}), cn(\mathcal{D}), lb(\mathcal{D}))$ where:

- $nm(\mathcal{D})$ is a label that identifies $\mathcal{D}$ in the range of a CMMN model;
- $ds(\mathcal{D})$ is the data schema (i.e., the information model) that captures the data maintained by $\mathcal{D}$;
- $cn(\mathcal{D})$ is the subset of modeling elements $\mathcal{E}_x \subseteq (\mathcal{E}_\mathcal{T} \cup \mathcal{E}_\mathcal{V})$ that perform an operation on $\mathcal{D}$;

- $lb(\mathcal{D}) \subseteq cn(\mathcal{D}) \times \texttt{expr}$ is a total function that associates any element of $cn(\mathcal{D})$ with an expression $\texttt{expr}$ denoting the operation to be performed on $\mathcal{D}$ (or on its data schema $ds(\mathcal{D})$).

In the above definition, we firstly make explicit which tasks (or event listeners) can affect (or are affected by) a case file item $\mathcal{D}$, see $cn(\mathcal{D})$. Secondly, we associate $\mathcal{D}$ with a data schema $ds(\mathcal{D})$. For example, we can imagine that the case file item *sensors data* in the CMMN model of Figure 4 corresponds to a relational database schema including two object classes ("Sensor" and "Reading") and a relationship between them. We can assume that $ds(\mathcal{D})$ is represented as a UML class diagram.[2] The class "Sensor" can be used to list the physical sensors installed in the shop, and the class "Reading" to collect the single readings of the sensors. However, a CMMN model can also include case file items that have no data schema associated with them, i.e., such that $ds(\mathcal{D}) = \emptyset$. For example, the case file item *marketing report* can be seen as a traditional text document with plots and graphics.

Thirdly, the connection between tasks (or event listeners) and $\mathcal{D}$ must be labeled with an expression $\texttt{expr}$, which denotes the kind of operation that is performed on $\mathcal{D}$, see $lb(\mathcal{D})$. Let us denote with $OP^+$ the same collection of operations included in $OP$, plus two further operations: *read* and *predicate on*. If $ds(\mathcal{D}) = \emptyset$, then $\texttt{expr}$ corresponds to an operation (selected from $OP^+$) that can be performed on $\mathcal{D}$. For example, if we consider the CMMN model of Figure 4, the case file item *marketing report* can be defined as follows:

$$\begin{aligned}\mathcal{D} = (&\mathit{marketing\ report}, \emptyset,\\ &\{\mathit{Marketing\ actions}, \mathit{acceptable\ conversion\ rate}\},\\ &\{\langle \mathit{Marketing\ actions}, \mathit{create}\rangle,\\ &\ \langle \mathit{acceptable\ conversion\ rate}, \mathit{predicate\ on}\rangle\}).\end{aligned}$$

This basically means that the *Marketing actions* task leads to the creation of a new marketing report (i.e., $\texttt{expr} = \mathit{create}$) that predicates on the event listener *acceptable conversion rate* (i.e., $\texttt{expr} = \mathit{predicate\ on}$). In addition, if $ds(\mathcal{D}) \neq \emptyset$, then it is possible to create complex expressions that predicate on the single components of $ds(\mathcal{D})$. For example, if $ds(\mathcal{D})$ is expressed as a UML Class Diagram, $\texttt{expr}$ can be formulated as an OCL (Object Constraint Language) rule, e.g., for reading or updating single rows of the "Reading" class associated to (see previously) the case file item *sensors data*.

The above rigorous specification of case file items allows us to extend traditional CMMN, as follows:

**Definition 3** (CMMN-1 model)**.** A CMMN-1 model is a tuple $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}})$ where:

- $(\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R})$ is a CMMN model as specified in Definition 1, with the following modifications:

[2] We notice that our approach can easily be transferred to other data modeling languages.

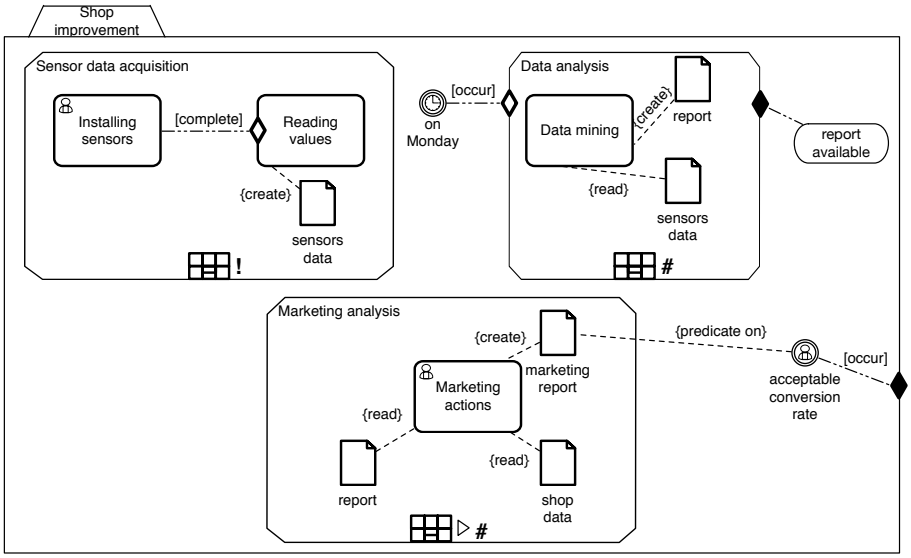



Figure 6: Level 1 (Failure awareness) compliant process model.

- $\mathcal{E}_{\mathcal{D}} \in \mathcal{E}$ contains case items specified as in Definition 2;
- $\mathcal{E}_{\mathcal{F}} \in \mathcal{E}$ allows the presence of connectors linking directly case file items with event listeners and tasks.
- $\mathcal{R}$ includes direct relationships between case file items and tasks (or event listeners). Such relationships are not possible in the standard specification of a CMMN model, cf. Section 4.

- $\mathcal{J}_{\mathcal{D}} : \mathcal{E}_{\mathcal{D}} \not\to (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$ is a binary relationship that relates a case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ to a modeling element $e_x \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$. Note that $\mathcal{J}_{\mathcal{D}} \subset \mathcal{R}$.

- $\mathcal{W}_{\mathcal{D}} : \mathcal{E}_{\mathcal{D}} \to z$ is a binary relationship that associates to any case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ a continuous value $z \in [0, 1]$ reflecting its level of criticality within the process.

Notice that $\mathcal{J}_{\mathcal{D}}$ does not need to be total, as certain case file items are not associated with a task (or event listener) that interacts with them. In the latter case, if $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ is a case file item with $cn(e_{\mathcal{D}}) = \emptyset$, then for each $e_x \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$ no relationship $j \in \mathcal{J}_{\mathcal{D}}$ exists such that $j = \langle e_{\mathcal{D}}, e_x \rangle$.

On the other hand, $\mathcal{W}_{\mathcal{D}}$ is total and is used to indicate the criticality of a case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ with respect to the degree of resiliency that a designer wants to achieve for the process. For the sake of simplicity, in the rest of the paper we assume the existence of a function $\mathcal{K}$ that bounds the levels of criticality $c_L$ of a CMMN modeling element to only five possible numeric values, as shown in Equation 1.

$$\mathcal{K}(c_L) = \begin{cases} 0 & \text{if } c_L = none, \\ 0.2 & \text{if } c_L = low, \\ 0.5 & \text{if } c_L = medium, \\ 0.8 & \text{if } c_L = high, \\ 1 & \text{if } c_L = critical \end{cases} \tag{1}$$

Designing a CMMN-1 model allows the connections between tasks/event listeners and case file items, annotated with the operations performed on the data. The use of this extension in the case study is shown in Figure 6. The new elements in the model as well as the presence of a relationship that explicitly couples case file items with a criticality value allow the designer to identify the data that might have more impact in case of their unavailability. For example, to express that the lack of sensors data will have more impact than the lack of another kind of data (as the former can cause a domino effect affecting all the tasks/event listeners in the process), we can perform the following assignment:

- $\mathcal{W}_{\mathcal{D}}(sensors\ data) = \mathcal{K}(critical) = 1$
- $\mathcal{W}_{\mathcal{D}}(shop\ data) = \mathcal{K}(medium) = 0.5$
- $\mathcal{W}_{\mathcal{D}}(marketing\ report) = \mathcal{K}(medium) = 0.5$
- $\mathcal{W}_{\mathcal{D}}(report) = \mathcal{K}(low) = 0.2$

(2)

Given the above ingredients, we can define CMMN-1 models that fully cover the Level 1 of the maturity model.

**Definition 4** (Level-1 compliant model)**.** Let $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}})$ be a CMMN-1 model. $\mathcal{N}$ is said to be a "Level-1 compliant model" if and only if, for each $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ with $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$, there exist $j \in \mathcal{J}_{\mathcal{D}}$ and $e_x \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$ such that $j = \langle e_{\mathcal{D}}, e_x \rangle$.

Notice that, by Definition 2, if a case file item $e_{\mathcal{D}}$ is involved in a relationship $j = \langle e_{\mathcal{D}}, e_x \rangle \in \mathcal{J}_{\mathcal{D}}$, this automatically means that $cn(e_{\mathcal{D}}) \neq \emptyset$ and that exists $\langle e_x, \textbf{\textit{expr}} \rangle \in lb(e_{\mathcal{D}})$, i.e., $j$ will be implicitly associated with the operation ***expr*** to be performed on $e_{\mathcal{D}}$ by $e_x$.

If a CMMN-1 model $\mathcal{N}$ is not Level-1 compliant, one may want to quantify the percentage of compliance ($pc_{L1}$) between the model and the rules specified in Definition 2. This can be achieved through Equation 3, where $||cn(e_{\mathcal{D}})|| = 1$ if $|cn(e_{\mathcal{D}})| > 0$, and $||cn(e_{\mathcal{D}})|| = 0$ if $|cn(e_{\mathcal{D}})| = 0$, being $|cn(e_{\mathcal{D}})|$ the *number* of tasks (or event listeners) related to a case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$. In a nutshell, $||cn(e_{\mathcal{D}})||$ indicates the absence/presence of (at least) a connection between $e_{\mathcal{D}}$ and a task/event listener $e_x \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$.

| Cases | Case File Items | | | | % of Compliance |
|---|---|---|---|---|---|
| | $\|\|cn(\textbf{\textit{sensors data}})\|\|$ $\mathcal{W}_{\mathcal{D}} = 1$ | $\|\|cn(\textbf{\textit{shop data}})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.5$ | $\|\|cn(\textbf{\textit{marketing report}})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.5$ | $\|\|cn(\textbf{\textit{report}})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.2$ | |
| case 1 | 1 | 1 | 1 | 1 | 100% |
| case 2 | 1 | 0 | 1 | 1 | 77% |
| case 3 | 1 | 0 | 1 | 0 | 68% |
| case 4 | 0 | 1 | 1 | 1 | 54% |

Table 1: Percentage of compliance of CMMN model of Fig. 6 wrt. different values of $||cn(e_{\mathcal{D}})||$

$$pc_{L1}(\mathcal{N}) = \begin{cases} 100\% \quad if \quad \sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) = 0 \\ \\ \frac{\sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \left( \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||cn(e_{\mathcal{D}})|| \right)}{\sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}})} \cdot 100 \quad if \quad \sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0 \end{cases} \tag{3}$$

On the one hand, the first part of Equation 3 shows the trivial cases in which: *(i)* no case file item has been defined for the CMMN model $\mathcal{N}$; *(ii)* each case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ included in $\mathcal{N}$ has a criticality value equal to 0, i.e., such that $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) = \mathcal{K}(none) = 0$. In both cases, $\mathcal{N}$ is considered to be trivially Level-1 compliant, i.e., $pc_{L1}(\mathcal{N}) = 100\%$. In particular, in the second case *(ii)* the designer is explicitly declaring that all case file items of $\mathcal{N}$ do not affect its resiliency.

On the other hand, the second part of Equation 3 aims at checking if any case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ having $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$ is associated to (at least) a task (or event listener) through the relationship $\mathcal{J}_{\mathcal{D}}$, and this can be verified by simply evaluating if $||cn(e_{\mathcal{D}})|| = 1$. If so, Definition 4 is satisfied and, consequently, we can state that a CMMN-1 model $\mathcal{N}$ is Level-1 compliant when $pc_{L1}(\mathcal{N}) = 100\%$.

When a full compliance with Level 1 is not completely achieved, the equation allows a better understanding of the impact and the risks of such a non-compliance, returning a percentage value that implicitly measures the *distance* between the model and the complete achievement of the resiliency level. Specifically, the percentage of compliance varies in presence of case file items $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ for which $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$ and $||cn(e_{\mathcal{D}})|| = 0$. Higher levels of criticality will correspond to lower percentages returned by Equation 3.

**Example 6.1.** Let us consider the CMMN-1 model $N$ in Figure 6. As shown in *case 1* of Table 1, independently by the level of criticality of its case file items, such a model has a percentage of compliance of 100%, since each case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ is related to (at least) a task (or event listener) via relationship $\mathcal{J}_{\mathcal{D}}$, i.e., $||cn(e_{\mathcal{D}})|| = 1$. Let us discuss now other possible cases in which compliance is not reached. First, let us suppose that case file item *sensors data*, which is considered critical for the resilience of the process, i.e., $\mathcal{W}_{\mathcal{D}}(sensors\ data) = \mathcal{K}(critical) = 1$, is not linked to any task (or event listener) through relationship $\mathcal{J}_{\mathcal{D}}$, i.e., $cn(e_{\mathcal{D}}) = \emptyset$. If we assume that the level of criticality of the other case file items is as indicated in the assignment performed in (2), Equation 3 returns

$pc_{L1}(\mathcal{N}) = 54\%$ (cf. *case 4*). By analyzing also *case 2* and *case 3* of Table 1, it is interesting to notice that the level of criticality of case file items plays a key role in the quantification of the percentage of compliance, e.g., in *case 4*, the absence of connections for a (single) critical case file item has a greater impact than the absence of connections for two case file items at the same time (such as in *case 3*), but with a lower level of criticality assigned to them.

### *6.3. Level 2 – Data resilience*

To cope with *alternative data*, we propose to associate case file items with one or more alternative data sources, called *alternative case file items*, to be considered when the quality of information provided by the primary case file item is low.

**Definition 5** (Case File Item with alternatives)**.** A case file item with alternatives is a tuple $\mathcal{D} = (nm(\mathcal{D}), ds(\mathcal{D}), cn(\mathcal{D}), lb(\mathcal{D}), al(\mathcal{D}))$ with the following possibilities:

- $(nm(\mathcal{D}), ds(\mathcal{D}), cn(\mathcal{D}), lb(\mathcal{D}))$ is a case file item as specified in Definition 2;
- $al(\mathcal{D})$ is a binary relationship $\langle e_{\mathcal{D}}, e_x \rangle$ in which two elements $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ and $e_x \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$ are related if and only if $e_x \in cn(\mathcal{D})$ and $e_{\mathcal{D}} \neq \mathcal{D}$ is the alternative case file item for $\mathcal{D}$. Notice that $\mathcal{D}$, for any $e_x \in cn(\mathcal{D})$, can provide only a single alternative case file item for $e_x$.

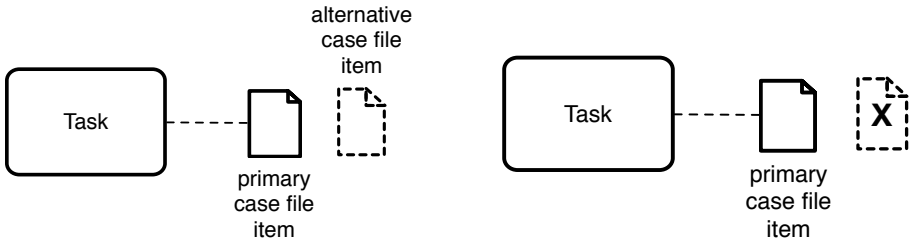
According to the previous definition, a case file item $\mathcal{D}$ can act as the *primary* data source for some tasks/event listeners and as an *alternative* data source for other case file items in the CMMN model. To be more precise, we state that:

- $\mathcal{D}$ is *primary* for a task/event listener $e_x \in cn(\mathcal{D})$ if and only if it does not exist any case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ such that $e_D \neq \mathcal{D}$, $e_x \in cn(e_{\mathcal{D}})$ and $\langle \mathcal{D}, e_x \rangle \in al(e_{\mathcal{D}})$.
- $\mathcal{D}$ is *alternative* for a case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ and a task/event listener $e_y \in cn(\mathcal{D})$ if and only if $e_{\mathcal{D}} \neq \mathcal{D}$, $e_y \in cn(e_{\mathcal{D}})$ and $\langle \mathcal{D}, e_y \rangle \in al(e_{\mathcal{D}})$.

The above definition enables a process designer to build different versions of a case file item $\mathcal{D}$ where:

1. no alternative case file items are specified for $\mathcal{D}$, i.e., $al(\mathcal{D}) = \emptyset$;
2. alternative case file items are specified just for a subset of the tasks/event listeners interacting with $\mathcal{D}$, i.e., there exists some $e_y \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}})$ such that $e_y \in cn(\mathcal{D})$, but it does not exist any $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$, such that $e_{\mathcal{D}} \neq \mathcal{D}$ and $\langle e_{\mathcal{D}}, e_y \rangle \in al(\mathcal{D})$;
3. no alternative case file item has been identified for some of the tasks/event listeners interacting with $\mathcal{D}$. In this case, a special keyword "NOP" can be used to make this aspect explicit. For example, if a designer does not identify an alternative case file item for $\mathcal{D}$ related to its interaction with $e_y \in cn(\mathcal{D})$, then $\langle NOP, e_y \rangle \in al(\mathcal{D})$;

4. each task (or event listener) belonging to $cn(\mathcal{D})$ is provided with an alternative case file item (or a NOP) in $al(\mathcal{D})$.




(a) A case file item with an alternative.

(b) No alternative identified for a case file item.

Figure 7: Example of alternative case file items.

For graphically representing an alternative case file item, we propose to add a new icon with a shape identical to a "traditional" case file item, but with a dashed border strictly attached to the original data source (see Figure 7a). If the process designer is aware that no alternative case file item is possible for a primary one, then the dashed border icon is explicitly labeled with an 'X' (which corresponds to NOP, see Figure 7b).

In the example in Figure 8, two alternative sources are defined: *public data* as an alternative to *sensors data* and the *Data mining* task, and *market analysis* to be used instead of the *report* produced by the *Marketing actions* task. For the other case file items included in the model, no alternative has been identified, and this is made explicit by using alternative case file items labeled with an 'X'.

For example, according to Definition 5, the primary case file item called *sensors data* and one of its alternatives called *public data* can be specified as follows:

$$\begin{aligned}\mathcal{D}_1 =&(\textit{sensors data}, \emptyset,\\ &\{\textit{Reading values}, \textit{Data mining}\},\\ &\{\langle \textit{Reading values}, \textit{create}\rangle, \langle \textit{Data mining}, \textit{read}\rangle\},\\ &\{\langle \textit{NOP}, \textit{Reading values}\rangle, \langle \textit{public data}, \textit{Data mining}\rangle\}).\end{aligned}$$

$$\begin{aligned}\mathcal{D}_2 =&(\textit{public data}, \emptyset,\\ &\{\textit{Data mining}\}, \{\langle \textit{Data mining}, \textit{read}\rangle\}, \{\emptyset\}).\end{aligned}$$

As previously discussed, a case file item can act as primary for some tasks/event listeners and as alternative for other case file items in the CMMN model. It is also possible to define several alternative case file items for a given task/event listener. This latter case is possible if for a primary case file item $e_{\mathcal{D}_1} \in \mathcal{E}_{\mathcal{D}}$ and a task (or event listener) $e_x \in cn(e_{\mathcal{D}_1})$ there exists an alternative case file item $e_{\mathcal{D}_2} \in \mathcal{E}_{\mathcal{D}}$ such that $e_{\mathcal{D}_2} \neq e_{\mathcal{D}_1}$ and $\langle e_{\mathcal{D}_2}, e_x\rangle \in al(e_{\mathcal{D}_1})$, and

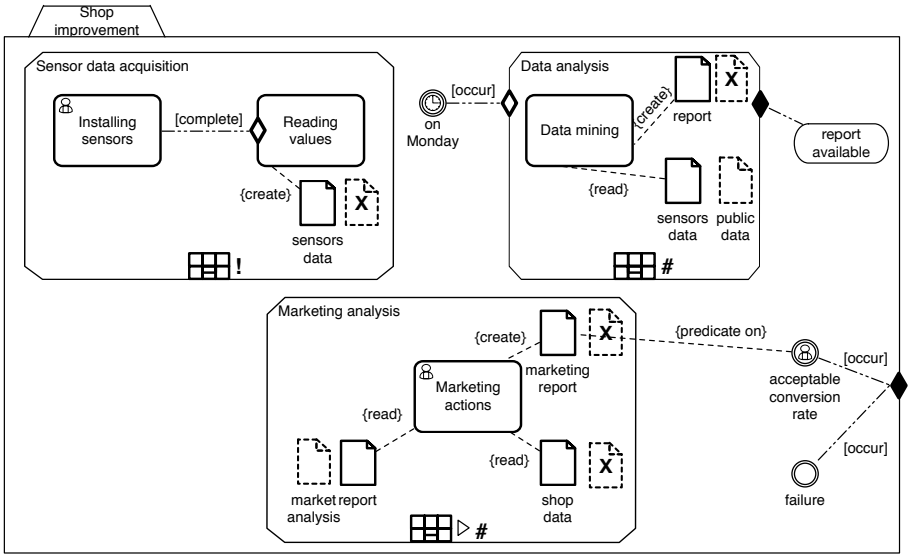



Figure 8: Level 2 compliant process model.

for which there exists a third case file item $e_{\mathcal{D}_3} \in \mathcal{E}_{\mathcal{D}}$ such that $e_{\mathcal{D}_3} \neq e_{\mathcal{D}_2}$, $e_{\mathcal{D}_3} \neq e_{\mathcal{D}_1}$ and $\langle e_{\mathcal{D}_3}, e_x \rangle \in al(\mathcal{D}_2)$, and so on. Therefore, Definition 5 allows building "priority chains" of alternative case file items associated to a primary one. Their "priority" in being chosen as alternative data sources is made graphically evident through an integer number on the top left corner of their icon, with '1' meaning maximum priority.

The introduction of alternative case file items allows us to further extend CMMN as follows:

**Definition 6** (CMMN-2 model)**.** A CMMN-2 model is a tuple $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}})$ where:

- $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}})$ is a CMMN-1 model as specified in Definition 3 with the only modification that $\mathcal{E}_{\mathcal{D}} \in \mathcal{E}$ contains case file items as specified in Definition 5.

Note that the relationship $\mathcal{J}_{\mathcal{D}}$ does not change. Thus, for CMMN-2 models, it will record the relationships between tasks (or event listeners) and primary/alternative case file items. From a graphical point of view, to avoid overloading the CMMN model, alternative case file items will be only connected to their primary counterpart. For the sake of understandability, in the rest of this section, we will use the abbreviation $||\rho(\mathcal{D}, e_x)||$ to evaluate if a case file item $\mathcal{D}$ is primary or not with respect to a task (or event listener) $e_x$. Specifically, $||\rho(\mathcal{D}, e_x)|| = 1$ means that $\mathcal{D}$ is primary for $e_x$, otherwise $||\rho(\mathcal{D}, e_x)|| = 0$.

**Definition 7** (Level-2 compliant model)**.** Let $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}})$ be a CMMN-2 model. $\mathcal{N}$ is said to be a "Level-2 compliant model" if and only if:

- $\mathcal{N}$ is a Level-1 compliant model (cf. Definition 4);
- for each pair $\langle e_{\mathcal{D}}, e_x \rangle \in \mathcal{J}_{\mathcal{D}}$ such that $||\rho(e_{\mathcal{D}}, e_x)|| = 1$ and $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$, then: *(i)* there exists $e_{\mathcal{D}_2} \in \mathcal{E}_{\mathcal{D}}$ such that $e_{\mathcal{D}_2} \neq e_{\mathcal{D}}$ and $\langle e_{\mathcal{D}_2}, e_x \rangle \in al(e_{\mathcal{D}})$, or *(ii)* $\langle NOP, e_x \rangle \in al(e_{\mathcal{D}})$.

In a nutshell, a CMMN-2 model is compliant with Level 2 of the maturity model if any primary case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ included in the model provides (at least) an alternative data source (or explicitly declare its absence with the NOP label) for each task/event listener that is associated to it via relation $cn(e_{\mathcal{D}})$.

When a full compliance of a CMMN-2 model $\mathcal{N}$ with Level 2 is not completely achieved, we can leverage on Equation 4 to quantify such a non-compliance. In that case, the percentage of compliance depends on the presence of primary case file items $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ for which $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$ but that do not provide any alternative for some of the tasks/event listeners connected to them, i.e., such that $||\rho(e_{\mathcal{D}}, e_x)|| = 1$ and $||al(e_{\mathcal{D}}, e_x)|| = 0$. The latter is an abbreviation used to check if $e_{\mathcal{D}}$ provides an alternative case file item for $e_x$. If this happens, $||al(e_{\mathcal{D}}, e_x)|| = 1$, otherwise it will be equal to 0.

$$pc_{L2}(\mathcal{N}) = \begin{cases} 100\% \;\; if \;\; \sum_{\langle e_{\mathcal{D}}, e_x \rangle \in \mathcal{J}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\rho(e_{\mathcal{D}}, e_x)|| = 0 \\ \\ \frac{\sum_{\langle e_{\mathcal{D}}, e_x \rangle \in \mathcal{J}_{\mathcal{D}}} \left( \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\rho(e_{\mathcal{D}}, e_x)|| \cdot ||al(e_{\mathcal{D}}, e_x)|| \right)}{\sum_{\langle e_{\mathcal{D}}, e_x \rangle \in \mathcal{J}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\rho(e_{\mathcal{D}}, e_x)||} \cdot 100 \\ \\ if \;\; \sum_{\langle e_{\mathcal{D}}, e_x \rangle \in \mathcal{J}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\rho(e_{\mathcal{D}}, e_x)|| > 0 \end{cases} \tag{4}$$

The first part of Equation 4 shows the trivial cases in which: *(i)* no case file item has been defined for $\mathcal{N}$; *(ii)* each primary case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ included in $\mathcal{N}$ has a criticality value equal to 0, i.e., such that $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) = \mathcal{K}(none) = 0$. In both cases, $\mathcal{N}$ is considered to be trivially Level-2 compliant, i.e., $pc_{L2}(\mathcal{N}) = 100\%$.

Then, the second part of Equation 4 verifies that any primary case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ with $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$, and associated to a task (or event listener) $e_x$ through the relationship $\mathcal{J}_{\mathcal{D}}$, provides an alternative case file item for $e_x$, i.e., such that $||al(e_{\mathcal{D}}, e_x)|| = 1$. If so, Definition 7 is satisfied and, consequently, we can state that a CMMN-2 model $\mathcal{N}$ is Level-2 compliant when $pc_{L2}(\mathcal{N}) = 100\%$ (and $pc_{L1}(\mathcal{N}) = 100\%$ as well).

When a full compliance with Level 2 can not be guaranteed, similarly to what happens with Equation 3 at Level 1, also in this case the equation returns a percentage value that implicitly allows to measure how far is $N$ from being Level-2 compliant. Specifically, both the criticality of primary case file items and the presence/absence of alternative case file items play a crucial role for determining the value of $pc_{L2}(\mathcal{N})$.

**Example 6.2.** Let us consider the CMMN-2 model $N$ in Figure 8. As shown in *case 1* of Table 2, $pc_{L2}(\mathcal{N}) = 100\%$, since each primary case file item $e_{\mathcal{D}}$ that is

| Cases | Case File Items | | | | | | % of Compliance |
|---|---|---|---|---|---|---|---|
| | $\|\|al(\textbf{sensors data}, \textbf{Reading values})\|\|$ $\mathcal{W}_{\mathcal{D}} = 1$ | $\|\|al(\textbf{sensors data}, \textbf{Data Mining})\|\|$ $\mathcal{W}_{\mathcal{D}} = 1$ | $\|\|al(\textbf{shop data}), \textbf{Marketing actions})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.5$ | $\|\|al(\textbf{marketing report}), \textbf{Marketing actions})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.5$ | $\|\|al(\textbf{report}), \textbf{Data mining})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.2$ | $\|\|al(\textbf{report}), \textbf{Marketing actions})\|\|$ $\mathcal{W}_{\mathcal{D}} = 0.2$ | |
| case 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100% |
| case 2 | 1 | 1 | 1 | 1 | 1 | 0 | 94% |
| case 3 | 1 | 1 | 0 | 1 | 1 | 1 | 85% |
| case 4 | 1 | 0 | 1 | 1 | 1 | 1 | 70% |
| case 5 | 0 | 0 | 1 | 1 | 1 | 1 | 41% |

Table 2: Percentage of compliance of CMMN model of Fig. 8 with respect to different values of $||al(e_{\mathcal{D}}, e_x)||$

related to a task or event listener $e_x$ is also associated to an alternative version of the data source, or with a NOP value, i.e., such that $||al(e_{\mathcal{D}}, e_x)|| = 1$. On the other hand, if we assume that the level of criticality of case file items is the one indicated in the assignment performed in (2), we notice that in *cases 2*, *3*, *4* and *5* the full compliance with Level-2 is not achieved. It is important to point out that the decrease of the percentage of compliance strongly depends by the absence of alternatives for case file items that are considered critical for the resilience of the process. In particular, in *cases 4* and *5* the absence of one alternative (*case 4*) or two alternatives (*case 5*) for case file item *sensors data* has an evident greater impact than the absence of alternatives for case file items *report* (*case 2*) and *shop data* (*case 3*), whose level of criticality is *low* and *medium*, respectively.

### *6.4. Level 3 – Milestone resilience*

Similarly to alternative data, alternative milestones can be defined to improve the level of resilience of CMMN models. In CMMN, the concept of "milestone" is used to capture the completion of a major deliverable event necessary to make progress toward the goals implied by a successful execution of a CMMN model. To be more precise, there are two ways to codify the milestones in a CMMN model. On the one hand, a milestone can be associated with a specific entry condition (attached to the milestone itself), whose fulfillment indicates that the milestone has been achieved. On the other hand, a milestone can be triggered when an exit condition of another modeling element is satisfied.

When it is not possible to trigger a milestone, it would be desirable to provide some alternative "best-effort" milestone that enables the completion of a portion of work in a reasonable way. While CMMN allows specifying several "primary" milestones in the range of a case, it neglects the concept of secondary milestones that are alternative to primary ones. Therefore, in order to define CMMN models compliant with Level 3 of the maturity model, we first need to clarify this concept.

**Definition 8** (Milestone with alternatives)**.** A milestone with alternatives is a tuple $\mathcal{M} = (nm(\mathcal{M}), ec(\mathcal{M}), cn(\mathcal{M}), xc(\mathcal{M}), al(\mathcal{M}))$ where:

- $nm(\mathcal{M})$ is a label that identifies $\mathcal{M}$ in the range of a CMMN model;
- $ec(\mathcal{M})$ is the expression denoting the entry condition required to activate the milestone;

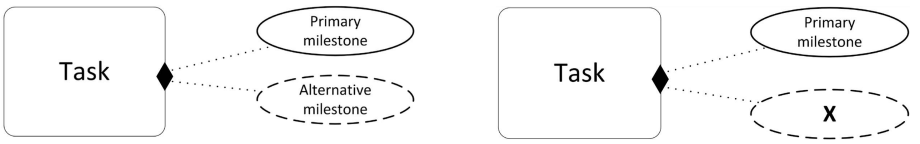
- $cn(\mathcal{M})$ is the subset of modeling elements $\mathcal{E}^{'} \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{V}} \cup \mathcal{E}_{\mathcal{S}})$ that predicate on $\mathcal{M}$ via $ec(\mathcal{M})$;
- $xc(\mathcal{M})$ is the subset of modeling elements $\mathcal{E}^{''} \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{C}} \cup \mathcal{E}_{\mathcal{S}})$ that are connected to $\mathcal{M}$ via an exit condition;
- $al(\mathcal{M})$ is the alternative milestone of $\mathcal{M}$, such that $al(\mathcal{M}) \in \mathcal{E}_{\mathcal{M}}$ and $al(\mathcal{M}) \neq \mathcal{M}$.

According to the above definition, a milestone *(i)* can be associated to an entry condition, *(ii)* can be linked to tasks, event listeners or stages that predicate on its entry condition, *(iii)* can be associated to the exit conditions of tasks, cases or stages, and *(iv)* can be coupled with a specific alternative milestone. If no alternative milestone has been identified, a special keyword "NOP" can be used to make this aspect explicit, i.e., $al(\mathcal{M}) = NOP$.

Based on the foregoing, we enforce a separation between *primary* and *alternative* milestones. Specifically, given a milestone $\mathcal{M} \in \mathcal{E}_{\mathcal{M}}$, we state that $\mathcal{M}$ is *primary* if and only if it does not exist any other milestone $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$ such that $e_M \neq \mathcal{M}$ and $\mathcal{M} \in al(e_{\mathcal{M}})$. Conversely, if such a milestone $e_{\mathcal{M}}$ exists, then $\mathcal{M}$ is said to be *alternative* for $e_{\mathcal{M}}$.

Notice that, differently from primary case file items, which may be associated with several alternatives when they are related to more tasks (or event listeners), a primary milestone can be associated just to a single "direct" alternative (nonetheless, a priority chain of alternatives attached to a primary milestone is allowed, see later). This rigidity is intimately tied to the notion of milestone itself. Since a milestone represents a critical achievement in the range of a process, the "meaning" of its achievement is well-defined and is reflected by a certain "business value" that does not depend by the specific process path followed to reach it or by the number of modeling elements that insist on it.

From a graphical point of view, alternative milestones are represented with a shape identical to a "traditional" milestone, but with a dashed border strictly attached to the original milestone (see Figure 9a). Similarly to alternative case file items, if no alternative milestone can be identified for a certain primary milestone, we allow the process designer to associate it with a dashed border icon marked with an 'X' (see Figure 9b).




(a) A milestone with an alternative version associated to it.

(b) No alternative identified for a milestone.

Figure 9: Example of alternative milestones.

In addition, according to Definition 8, we allow a chain of alternative milestones associated to a primary one to be built; their priority is graphically

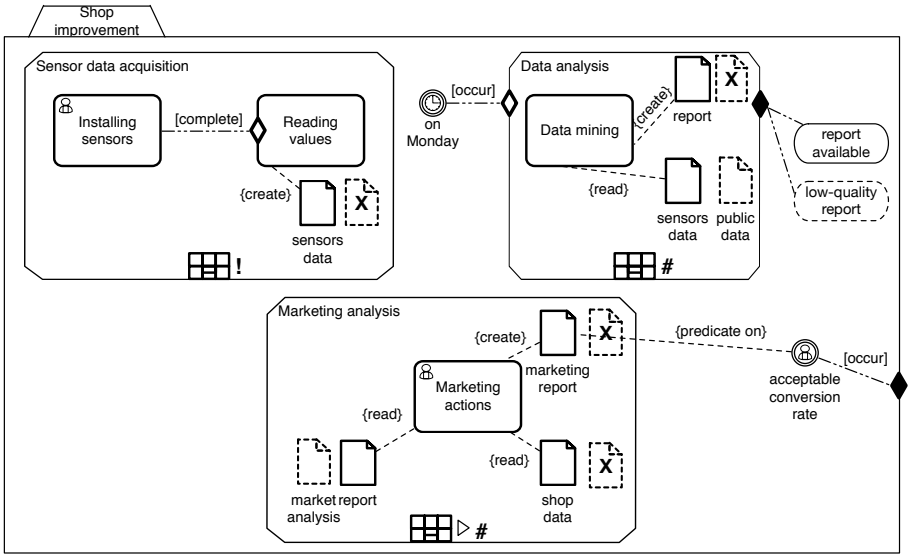



Figure 10: Level 3 compliant process model.

provided through an integer number in the top left corner of their icon, with '1' meaning maximum priority.

In the example of Figure 10, an alternative milestone called *low quality report* is coupled with the primary milestone *report available*. This specifies the fact that if it is not possible to produce a quality report after the completion of the *Data analysis* stage, then it would be desirable to produce (at least) some kind of report, even if of lower quality. For example, according to Definition 8, the two milestone introduced before can be specified as follows:

$$\mathcal{M}_1 = (\textit{report available}, \emptyset, \emptyset, \{\textit{Data analysis}\}, \{\textit{low-quality report}\}) \quad .$$

$$\mathcal{M}_2 = (\textit{low-quality report}, \emptyset, \emptyset, \{\textit{Data analysis}\}, \emptyset) \quad .$$

With the introduction of alternative milestones, we can further extend CMMN as follows:

**Definition 9** (CMMN-3 model)**.** A CMMN-3 model is a tuple $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_\mathcal{D}, \mathcal{W}_\mathcal{D}, \mathcal{J}_\mathcal{M}, \mathcal{W}_\mathcal{M})$ where:

- $(\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_\mathcal{D}, \mathcal{W}_\mathcal{D})$ is a CMMN-2 model as specified in Definition 6, with the only modification that $\mathcal{E}_\mathcal{M} \in \mathcal{E}$ contains milestones as specified in Definition 8.
- $\mathcal{J}_\mathcal{M} : \mathcal{E}_\mathcal{M} \rightarrow (\mathcal{E}_\mathcal{T} \cup \mathcal{E}_\mathcal{C} \cup \mathcal{E}_\mathcal{V} \cup \mathcal{E}_\mathcal{S})$ is a binary relationship that relates a milestone $e_\mathcal{M} \in \mathcal{E}_\mathcal{M}$ to a modeling element $e_x \subseteq (\mathcal{E}_\mathcal{T} \cup \mathcal{E}_\mathcal{C} \cup \mathcal{E}_\mathcal{V} \cup \mathcal{E}_\mathcal{S})$ in one of the ways specified in Definition 8. Note that $\mathcal{J}_\mathcal{M} \subset \mathcal{R}$.

- $\mathcal{W}_{\mathcal{M}} : \mathcal{E}_{\mathcal{M}} \rightarrow z$ is a binary relationship that associates to any milestone $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$ a continuous value $z \in [0, 1]$ reflecting its level of criticality within the process.

Notice that: *(i)* $\mathcal{J}_{\mathcal{M}}$ is total, i.e., all milestones in the CMMN model must be triggered by another modeling element $e_x \subseteq (\mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{C}} \cup \mathcal{E}_{\mathcal{V}} \cup \mathcal{E}_{\mathcal{S}})$ that interacts with them; *(ii)* $\mathcal{W}_{\mathcal{M}}$ is total and is used to indicate the criticality of a milestone $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$ with respect to the degree of resiliency that a designer wants to achieve for the process. For the sake of simplicity, we assume that the interpretation of $\mathcal{W}_{\mathcal{M}}$ is constrained through the (already discussed) function $\mathcal{K}$, which bounds the levels of criticality $c_L$ of a CMMN modeling element to only five possible numeric values, as shown in Equation 1.

In the following, we use abbreviation $||\rho(e_{\mathcal{M}})||$ to evaluate if a milestone $e_{\mathcal{M}} \in \mathcal{M}$ is a primary one. Specifically, $||\rho(e_{\mathcal{M}})|| = 1$ means that $e_{\mathcal{M}}$ is a primary milestone, otherwise $||\rho(e_{\mathcal{M}})|| = 0$.

**Definition 10** (Level-3 compliant model)**.** Let $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}}, \mathcal{J}_{\mathcal{M}}, \mathcal{W}_{\mathcal{M}})$ be a CMMN-3 model. $\mathcal{N}$ is said to be a "Level-3 compliant model" if and only if:

- $\mathcal{N}$ is a Level-2 compliant model (cf. Definition 7);
- for each $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$ such that $||\rho(e_{\mathcal{M}})|| = 1$ and $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{M}}) > 0$, then $al(e_{\mathcal{M}}) = NOP$ or there exists $e_{\mathcal{M}_2} \in \mathcal{E}_{\mathcal{M}}$ such that $e_{\mathcal{M}_2} \neq e_{\mathcal{M}}$ and $al(e_{\mathcal{M}}) = e_{\mathcal{M}_2}$.

That is, a CMMN-3 model $\mathcal{N}$ is compliant with Level 3 of the maturity model if each primary milestone specified in $\mathcal{N}$ is coupled with (at least) an alternative milestone, which expresses the achievement of a relaxed (sub-)goal condition or reflects the awareness that no alternative (sub-)goal is possible.

The percentage of compliance $pc_{L3}$ of a CMMN-3 model $\mathcal{N}$ with Level 3 can be quantified leveraging on Equation 5. The equation is built using the abbreviation $||al(e_{\mathcal{M}})||$, which is equal to 1 if $e_{\mathcal{M}}$ is associated to an alternative milestone, otherwise it will be equal to 0.

$$pc_{L3}(\mathcal{N}) = \begin{cases} 100\% \;\; if \;\; \sum_{e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}} \mathcal{W}_{\mathcal{M}}(e_{\mathcal{M}}) \cdot ||\rho(e_{\mathcal{M}})|| = 0 \\ \\ \frac{\sum_{e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}} \left( \mathcal{W}_{\mathcal{M}}(e_{\mathcal{M}}) \cdot ||\rho(e_{\mathcal{M}})|| \cdot ||al(e_{\mathcal{M}})|| \right)}{\sum_{e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}} \mathcal{W}_{\mathcal{M}}(e_{\mathcal{M}}) \cdot ||\rho(e_{\mathcal{M}})||} \cdot 100 \\ \\ \qquad if \;\; \sum_{e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}} \mathcal{W}_{\mathcal{M}}(e_{\mathcal{M}}) \cdot ||\rho(e_{\mathcal{M}})||) > 0 \end{cases} \tag{5}$$

To sum up, Equation 5 states that in the cases in which: *(i)* no milestone has been defined for the CMMN model $\mathcal{N}$; *(ii)* each primary milestone $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$ included in $\mathcal{N}$ has a criticality value equal to 0, i.e., such that $\mathcal{W}_{\mathcal{M}}(e_{\mathcal{M}}) = \mathcal{K}(none) = 0$, then $\mathcal{N}$ is considered to be trivially Level-3 compliant, i.e., $pc_{L3}(\mathcal{N}) = 100\%$.

The second part of Equation 5 verifies that each primary milestone $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$ with $\mathcal{W}_{\mathcal{M}}(e_{\mathcal{M}}) > 0$ provides an alternative milestone, i.e., such that $||al(e_{\mathcal{M}})|| = 1$. If so, Definition 10 is satisfied and, consequently, we can state that a CMMN-3 model $\mathcal{N}$ is Level-3 compliant when $pc_{L3}(\mathcal{N}) = 100\%$ (and, of course, $pc_{L2}(\mathcal{N}) = 100\%$). For example, this is exactly the case of the CMMN model in Figure 10, which is Level-3 compliant.

When a full compliance with Level 3 can not be guaranteed, the equation returns a percentage value that implicitly allows to measure how far is $N$ from being Level-3 compliant. Similarly to what happens with Equation 4 at Level 2, in this case the two variables that play a crucial role for determining the value of $pc_{L3}(\mathcal{N})$ are the level of criticality of primary milestones and the presence/absence of an alternative milestone for a primary one.

### *6.5. Level 4 – Process resilience*

In the previous level of the maturity model, we have discussed how the presence of alternative milestones allows to capture best-effort process (sub-) goals to be achieved when primary data sources are missing or not available. However, the quality of the alternative case file items is usually lower than their original counterpart, and sometimes this makes very complex (also) the achievement of alternative milestones. In order to mitigate this issue, the final level of our maturity model pushes a process designer to explicitly model the additional work required to improve the quality of alternative case file items to a degree that is comparable to their original counterpart and that allows to meet some process (primary or alternative) milestone.

To this aim, we need to introduce two further modeling elements: *error events* and *recovery stages*.

**Definition 11** (Error Event)**.** An error event $\mathcal{V}_{err} \in \mathcal{E}_{\mathcal{V}}$ is a tuple $(nm(\mathcal{V}_{err}), cn(\mathcal{V}_{err}), st(\mathcal{V}_{err}))$ where:

- $nm(\mathcal{V}_{err})$ is a label that identifies $\mathcal{V}_{err}$ in the range of a CMMN model;
- $cn(\mathcal{V}_{err}) \in \mathcal{E}_{\mathcal{D}}$ is the alternative case file item that triggers $\mathcal{V}_{err}$;
- $st(\mathcal{V}_{err}) \in \mathcal{E}_{\mathcal{S}}$ is the recovery stage that is activated once $\mathcal{V}_{err}$ is thrown.

An error event $\mathcal{V}_{err}$ represents a situation in which the use of an alternative case item causes the enactment of some recovery actions, embedded in a special recovery stage. We represent an error event with a lightning bolt marker within the event shape (see Figure 11).

According to Definition 11, $\mathcal{V}_{err}$ is strongly coupled to the alternative case file item that triggers it. Notice that $\mathcal{V}_{err}$ can only be triggered by a single $cn(\mathcal{V}_{err})$ and must be associated to a single recovery stage $st(\mathcal{V}_{err})$.

**Definition 12** (Recovery Stage)**.** A recovery stage $\mathcal{S}_{rec} \in \mathcal{E}_{\mathcal{S}}$ is a tuple $(nm(\mathcal{S}_{rec}), ev(\mathcal{S}_{rec}), ct(\mathcal{S}_{rec}), mt(\mathcal{S}_{rec}))$ where:

- $nm(\mathcal{S}_{rec})$ is a label that identifies $\mathcal{S}_{rec}$ in the range of a CMMN model;

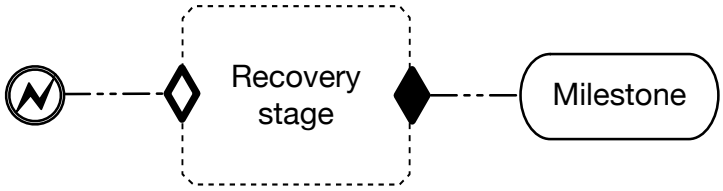



Figure 11: An example of an error event linked to a recovery stage.

- $ev(\mathcal{S}_{rec})$ is a non-empty subset of error events that cause the activation of $\mathcal{S}_{rec}$;
- $ct(\mathcal{S}_{rec})$ is the content of $\mathcal{S}_{rec}$, i.e., the set of modeling elements belonging to $\mathcal{S}_{rec}$ and their relationships.;
- $mt(\mathcal{S}_{rec}) \in \mathcal{E}_{\mathcal{M}}$ is the milestone that is achieved once the enactment of $\mathcal{S}_{rec}$ is completed.

A recovery stage $\mathcal{S}_{rec}$ is a special kind of discretionary stage (from a graphical point of view, the shape is identical but with a dashed outline) triggered by one or more error events trough an (always true) entry condition. The content of a stage is determined by the set of modeling elements belonging to $ct(\mathcal{S}_{rec})$. Potentially, a recovery stage allows to employ (and implicitly combine) also existing primary and alternative case file items. If a designer identifies no recovery stage for a specific error event, then $ct(\mathcal{S}_{rec}) = NOP$. If the process designer is aware that no recovery stage exists for a given error event, then the content of the stage is explicitly labeled with an 'X' (which corresponds to NOP). To avoid recursive situations, alternative case file items are not allowed in recovery stages. For the same reason, primary case file items in recovery stages are associated with a criticality value equal to 0, i.e., they can not affect negatively the resiliency of the process, as they are considered remedial sources whose existence can only improve the quality of data sources. Finally, an (always true) exit condition is associated to $\mathcal{S}_{rec}$, which is connected to a milestone $mt(\mathcal{S}_{rec})$, be it alternative or not.

In the example shown in Figure 12, in case the quality of *public data* is not considered sufficient, a recovery strategy – defined by a recovery stage – is required. In our example, the goal of the recovery stage is to support the achievement of the alternative milestone *low-quality report* by providing a *Data fixing task* able to increase the low-quality public data in a set of *revised public data* which will be used by the *Data mining* task.

For example, according to definitions 11 and 12, the error event *low-quality public data* and the recovery stage that is attached to it, called *Data fixing*, can be specified as follows:

$\mathcal{V} = (\textit{low-quality public data}, \textit{public data}, \textit{Data fixing})$

$\mathcal{S} = (\textit{Data fixing}, \{\textit{low-quality public data}\}, \{\ldots \textit{stage content} \ldots\}, \textit{low-quality report})$

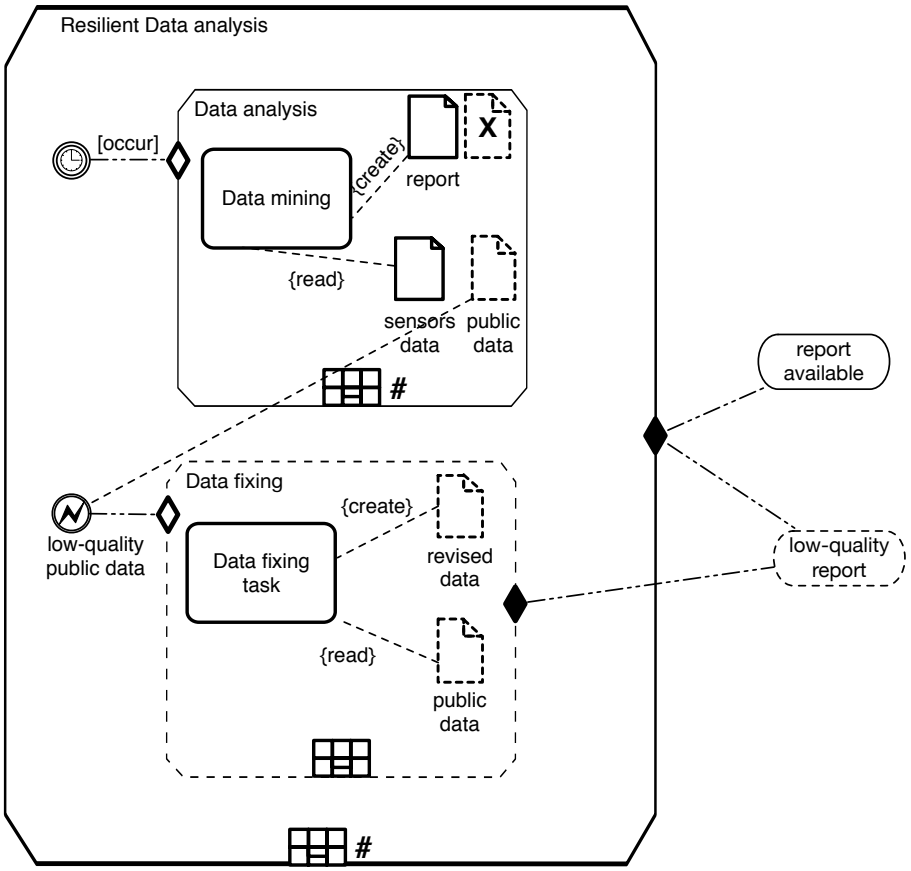



Figure 12: Level 4 compliant process model.

**Definition 13** (CMMN-4 model)**.** A CMMN-4 model is a tuple $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}}, \mathcal{J}_{\mathcal{M}}, \mathcal{W}_{\mathcal{M}})$ where:

- $(\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}}, \mathcal{J}_{\mathcal{M}}, \mathcal{W}_{\mathcal{M}})$ is a CMMN-3 model as specified in Definition 9, with the only variant that:
  - $\mathcal{E}_{\mathcal{V}} \in \mathcal{E}$ contains error events as specified in Definition 11.
  - $\mathcal{E}_{\mathcal{S}} \in \mathcal{E}$ contains recovery stages as specified in Definition 12.

For the sake of simplicity, in the rest of this section we use the following abbreviations:

- $||\alpha(e_{\mathcal{D}})||$ to check if a case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ is the first alternative source of a primary case file item. Specifically, $||\alpha(e_{\mathcal{D}})|| = 1$ means that $e_{\mathcal{D}}$ is an alternative case file item, otherwise $||\alpha(e_{\mathcal{D}})|| = 0$.

- $||\omega(e_{\mathcal{D}})||$ to check if a case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ is associated to a recovery stage $e_{\mathcal{S}} \in \mathcal{E}_{\mathcal{S}}$ (whose exit condition is linked to a milestone $e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$) via an error event $e_{\mathcal{V}} \in \mathcal{E}_{\mathcal{V}}$. Specifically, $||\omega(e_{\mathcal{D}})|| = 1$ means that there exists $e_{\mathcal{V}} \in \mathcal{E}_{\mathcal{V}}$ such that: *(i)* $cn(e_{\mathcal{V}}) = e_D$; *(ii)* $st(e_{\mathcal{V}}) = e_{\mathcal{S}} \in \mathcal{E}_{\mathcal{S}}$; *(iii)* $e_{\mathcal{V}} \in ev(e_{\mathcal{S}})$; *(iv)* $ct(e_{\mathcal{S}}) \neq \emptyset$ or $ct(e_{\mathcal{S}}) = NOP$; and *(v)* $mt(e_{\mathcal{S}}) = e_{\mathcal{M}} \in \mathcal{E}_{\mathcal{M}}$. If this is not the case, $||\omega(e_{\mathcal{D}})|| = 0$.

**Definition 14** (Level-4 compliant model)**.** Let $\mathcal{N} = (\mathcal{E}, \mathcal{U}, \mathcal{A}, \mathcal{R}, \mathcal{J}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}}, \mathcal{J}_{\mathcal{M}}, \mathcal{W}_{\mathcal{M}})$ be a CMMN-4 model. $\mathcal{N}$ is said to be a "Level-4 compliant model" if and only if:

- $\mathcal{N}$ is a Level-3 compliant model (cf. Definition 10);
- for each alternative case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ such that $||\alpha(e_{\mathcal{D}})|| = 1$ and $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$, then $||\omega(e_{\mathcal{D}})|| = 1$.

In a nutshell, in order to be compliant with Level 4 of the maturity model, for any alternative case file item of a CMMN-4 model whose criticality level is grater than 0, it is required to specify a recovery strategy (via an error event plus a recovery stage attached to a milestone), which indicates how to reduce the quality gap between the alternative data sources and their original counterpart with respect to the milestones achieved.

The percentage of compliance $pc_{L4}$ of a CMMN-4 model $\mathcal{N}$ with Level 4 can be quantified leveraging on Equation 6.

$$pc_{L3}(\mathcal{N}) = \begin{cases} 100\% \quad if \quad \sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\alpha(e_{\mathcal{D}})|| = 0 \\ \\ \frac{\sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \left( \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\alpha(e_{\mathcal{D}})|| \cdot ||\omega(e_{\mathcal{D}})|| \right)}{\sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\alpha(e_{\mathcal{D}})||} \cdot 100 \\ \\ \qquad if \quad \sum_{e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}} \mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) \cdot ||\alpha(e_{\mathcal{D}})|| > 0 \end{cases} \tag{6}$$

Similarly to the previous resiliency levels, the first part of Equation 6 deals with trivial cases. Specifically, if *(i)* no alternative case file item is defined for the CMMN model $\mathcal{N}$ or *(ii)* each alternative case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ included in $\mathcal{N}$ has a criticality value equal to 0, i.e., such that $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) = \mathcal{K}(none) = 0$, then $\mathcal{N}$ is considered to be trivially Level-4 compliant, i.e., $pc_{L4}(\mathcal{N}) = 100\%$.

The second part of Equation 6 checks that each alternative case file item $e_{\mathcal{D}} \in \mathcal{E}_{\mathcal{D}}$ with $\mathcal{W}_{\mathcal{D}}(e_{\mathcal{D}}) > 0$ is associated to a recovery stage via an error event, i.e., such that $||\omega(e_{\mathcal{D}})|| = 1$. If so, Definition 14 is satisfied and, consequently, we can state that a CMMN-4 model $\mathcal{N}$ is Level-4 compliant when $pc_{L4}(\mathcal{N}) = 100\%$ and, of course, $pc_{L3}(\mathcal{N}) = 100\%$ from the previous resiliency level.

## 7. Threats to validity

This section discusses factors that may call the proposal presented in this paper into question or diminish the meaningfulness of the results. These factors are denoted as threats to validity.

The first threat to validity is about the meaningfulness and applicability of the approach, and the lack of an extensive empirical evaluation. The proposed approach has been derived and assessed with respect to the running example that has been taken from a real scenario. Although adopting a real case study gives the opportunity to really test our assumptions and to get feedbacks from the final user, not all the possible causes of failures can be considered in this paper. Similarly, a general acceptance of the levels of awareness that have been proposed in this paper needs to be assessed with additional case studies.

In future work, we plan to evaluate how much the approach proposed in this paper effectively support designers to build more resilient models in specific scenarios. Empirically, this can be evaluated by taking groups of practitioners and performing a modeling experiment to compare produced models with and without adopting the approach (possibly supported by a design tool, see further). To facilitate this, the groups will get the same process modeling assignment, but some groups will be instructed to use systematically an assessment of the levels, as proposed in this paper, and other ones not to consider it. The factors to be compared might be, for example, the resilience of the produced model, speed of development, and user feedback. The hypothesis is that groups adopting the approach, and being able to measure at each step of the modeling how far away their model is from the achievement of a certain level, will produce better models, being driven toward resilience awareness.

A second threat to validity, related to the previous one, is the lack of a modeling tool realizing the approach. We envision such a tool as a CMMN editor in which the designer can evaluate the resilience level the model under construction is currently when editing. Such a tool would enforce awareness, by the designer, about the resilience level, and therefore would push the designer to use all the specific constructs we have defined in our approach. Notably, in the paper all the functional specifications for developing such a tool are provided, and formulas provided in Section 6 are exactly those ones to be realized in the tool for performing the level quantification over the model under construction. The formalization of the approach, which has lead to the exact specification of the way of computing the levels, is the only precise way to define the logic of such a tool, and has been provided in this paper exactly for this purposes.

A third threat to validity is about the practical conditions and assumptions under which the approach is effective. The existence of alternatives might not be always guaranteed; analogously, resilience might also be affected by other factors different from data, like resource unavailability, temporal constraint violations, or non-compliant behavior of certain parties. In these cases, even if the designer is aware of the resilience and would like to achieve a certain level, s/he cannot find alternatives and the resilience of the model is hindered. In the present paper we have focused on the data as main sources affecting process resiliency. Covering all the potential factors affecting resilience is out of the scope of this work. However, the investigation of such factors is in the list of the future extensions of this paper.

## 8. Discussion and concluding remarks

The resilience maturity model presented in this paper, based on an extension of CMMN , is a concrete tool to support process designers so they become aware of how resilient the processes they are working on are. At design time, it is important to be aware of failures, and to identify data and milestones alternatives, in order to be able to design alternative actions. From one point of view, flexible approaches cope with exceptional situations during run time, but only a deep awareness during design time can make really the process *resilient-by-design*.

One of the strengths of the proposed approach is the use of a formal specification of CMMN (and of its extensions) to provide a rigorous conceptualization of the maturity model and to build a classification framework that assesses the compliance of CMMN models against the resiliency levels. The following advantages can be identified:

- Firstly, to have a framework that is formally specified allows to remove any vagueness and ambiguity that may derive from an informal (textual) description, which reflects at most the intuition that is behind the framework itself. Conversely, the presence of a formal framework has allowed us to carefully define the aspects of process resiliency relevant for our research and those aspects that have not been investigated in the paper.

- Secondly, the formal specification of the maturity model has allowed us to precisely quantify the percentage of compliance of a CMMN model with respect to a resiliency level.

- Thirdly, the presence of a formal specification of the framework will easily allow us to realize a design tool that concretely implements the framework.

Based on the latter consideration, future work will comprise the design and development of a plug-in of some CMMN modeling tools, where the designer will have the possibility to verify at which maturity level the process s/he is currently designing can be classified. Calculating the maturity model of a CMMN model, and also providing a degree of how far a model is from the achievement of a desired level, is easily computable on the basis of formalization presented in Section 6. Such a tool would easily make the designer aware of the resilience of the process under modeling. Proper usability and the effectiveness of such a tool with practitioners will be evaluated as well.

*Funding.* The work of Andrea Marrella and Massimo Mecella was partly supported by the Italian projects *Social Museum and Smart Tourism* (CTN01_00034_23154), *NEPTIS* (PON03PE_00214_3), *RoMA - Resilience of Metropolitan Areas* (SCN_00064), and by the Sapienza project *Data-aware Adaptation of Knowledge-intensive Processes in Cyber-Physical Domains through Action-based Languages*. The work of Pierluigi Plebani and Barbara Pernici was partly supported by the Italian project *ITS2020* (CTN01_00176_166195).

## References


[1] Adger, W. N. Social and ecological resilience: are they related? *Progress in Human Geography 24*, 3 (2000), 347–364.

[2] Antunes, P., and Mouro, H. Resilient Business Process Management: Framework and services. *Expert Systems with Applications 38*, 2 (2011), 1241 – 1254.

[3] Ardagna, D., Baresi, L., Comai, S., Comuzzi, M., and Pernici, B. A service-based framework for flexible business processes. *IEEE Software 28*, 2 (2011), 61–67.

[4] Batini, C., and Scannapieco, M. *Data and Information Quality - Dimensions, Principles and Techniques*. Springer, 2016.

[5] Caralli, R. A., Allen, J. H., and White, D. W. *CERT Resilience Management Model: A Maturity Model for Managing Operational Resilience*. Addison-Wesley, 2010.

[6] Casati, F., Ceri, S., Pernici, B., and Pozzi, G. Workflow evolution. *Data & Knowledge Engineering 24*, 3 (1998), 211–238.

[7] Dadam, P., and Rinderle, S. Workflow evolution. In *Encyclopedia of Database Systems*. Springer, 2009, pp. 3540–3544.

[8] de Vrieze, P., and Xu, L. Resilience analysis of collaborative process management systems. In *17th IFIP Conf. on Virtual Enterprises (PRO-VE)* (2016), pp. 124–133.

[9] Fahland, D., Lübke, D., Mendling, J., Reijers, H. A., Weber, B., Weidlich, M., and Zugal, S. Declarative versus imperative process modeling languages: The issue of understandability. In *10th Int. Workshop on Business Process Modeling, Development, and Support (BPMDS)* (2009), pp. 353–366.

[10] Firmani, D., Mecella, M., Scannapieco, M., and Batini, C. On the Meaningfulness of "Big Data Quality". *Data Science and Engineering 1*, 1 (2016), 6–20.

[11] Gunderson, L. H. Ecological resilience–in theory and application. *Annual Review of Ecology and Systematics* (2000), 425–439.

[12] Hallerbach, A., Bauer, T., and Reichert, M. Capturing variability in business process models: the Provop approach. *Journal of Software Maintenance and Evolution: Research and Practice 22*, 6-7 (2009).

[13] Hallerbach, A., Bauer, T., and Reichert, M. Configuration and Management of Process Variants. In *Handbook on Business Process Management vol. 1*, International Handbooks on Information Systems. Springer Berlin Heidelberg, 2010.

[14] Hamari, J. Do badges increase user activity? a field experiment on the effects of gamification. *Computers in Human Behavior* (2015).

[15] Hammer, M. *The Reengineering Revolution*. HarperCollins, 1995.

[16] Hollnagel, E., Woods, D. D., and Leveson, N. *Resilience Engineering: Concepts and Precepts*. Ashgate Publishing, Ltd., 2007.

[17] Humayoun, S. R., Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Bortenschlager, M., and Steinmann, R. The WORKPAD User Interface and Methodology: Developing Smart and Effective Mobile Applications for Emergency Operators. In *5th International Conference on Universal Access in Human-Computer Interaction (UAHCI 2009)* (2009), Springer Berlin Heidelberg, pp. 343–352.

[18] Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Gal, A., Kannengiesser, U., Mannhardt, F., Mendling, J., Oberweis, A., Reichert, M., Rinderle-Ma, S., Song, W., Su, J., Torres, V., Weidlich, M., Weske, M., and Zhang, L. The internet-of-things meets business process management: Mutual benefits and challenges. *CoRR abs/1709.03628* (2017).

[19] Marrella, A., and Lespérance, Y. Synthesizing a library of process templates through partial-order planning algorithms. In *14th Int. Conf. on Business Process Modeling, Development, and Support (BPMDS)* (2013), pp. 277–291.

[20] Marrella, A., Mecella, M., and Russo, A. Collaboration on-the-field : Suggestions and beyond. In *8th Int. Conf. on Information Systems for Crisis Response and Management (ISCRAM)* (2011).

[21] Marrella, A., Mecella, M., and Sardiña, S. Intelligent process adaptation in the SmartPM System. *ACM Transactions on Intelligent Systems and Technology 8*, 2 (2017), 25:1–25:43.

[22] Müller, G., Koslowski, T. G., and Accorsi, R. Resilience - A new research field in business information systems? In *16th Int. Conf. on Business Information Systems (BIS)* (2013), Springer, pp. 3–14.

[23] Nachira, F., Nicolai, A., Dini, P., Le Louarn, M., and L. Rivera Len, L. e. *Digital Business Ecosystems*. European Commission, 2007.

[24] Nurcan, S. A survey on the flexibility requirements related to business processes and modeling artifacts. In *41st Hawaii International International Conference on Systems Science (HICSS-41 2008), Proceedings, 7-10 January 2008, Waikoloa, Big Island, HI, USA* (2008), p. 378.

[25] OMG. Business Process Modeling and Notation, Version 2.0.2, Jan 2014.

[26] OMG. Case Management Model and Notation, Version 1.0, May 2014.

[27] Penadés, M. C., Núñez, A. G., and Canós, J. H. From planning to resilience: The role (and value) of the emergency plan. *Technological Forecasting and Social Change* (2016).

[28] Pesic, M., Schonenberg, H., and van der Aalst, W. M. P. DECLARE: Full support for loosely-structured processes. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference* (2007), pp. 287–300.

[29] Plebani, P., Marrella, A., Mecella, M., Mizmizi, M., and Pernici, B. Multi-party business process resilience by-design: A data-centric perspective. In *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings* (2017), E. Dubois and K. Pohl, Eds., Springer International Publishing, pp. 110–124.

[30] Reichert, M., and Weber, B. *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. Springer, 2012.

[31] Rosemann, M., and Recker, J. Context-aware process design exploring the extrinsic drivers for process flexibility. In *7th Int. Workshop on Business Process Modeling, Development, and Support (BPMDS)* (2006).

[32] Sadiq, S., and Orlowska, M. On capturing exceptions in workflow process models. In *3rd Int. Conf. on Business Information Systems (BIS)*. Springer, 2000, pp. 3–19.

[33] Suriadi, S., Weiss, B., Winkelmann, A., et al. Current research in risk-aware business process management: Overview, comparison, and gap analysis. *Communications of the Association for Information Systems 34*, 1 (2014), 933–984.

[34] Tjoa, S., Jakoubi, S., Goluch, G., Kitzler, G., Goluch, S., and Quirchmayr, G. A formal approach enabling risk-aware business process modeling and simulation. *IEEE Transactions on Services Computing 4*, 2 (2011), 153–166.

[35] Van De Walle, B., Turoff, M., and Hiltz, S. R. *Information Systems for Emergency Management*. M.E. Sharpe, 2009.

[36] van der Aalst, W. M. P. *Process Mining: Data Science in Action*. Springer, 2016.

[37] van der Aalst, W. M. P., Pesic, M., and Schonenberg, H. Declarative workflows: Balancing between flexibility and support. *Computer Science - R&D 23*, 2 (2009).

[38] Yates, T. M., and Masten, A. S. *Fostering the Future: Resilience Theory and the Practice of Positive Psychology.* John Wiley & Sons Inc, 2004.

[39] Zahoransky, R. M., Brenig, C., and Koslowski, T. Towards a process-centered resilience framework. In *10th Int. Conf. on Availability, Reliability and Security (ARES)* (2015), IEEE, pp. 266–273.

[40] Zahoransky, R. M., Koslowski, T., and Accorsi, R. Toward Resilience Assessment in Business Process Architectures. In *Computer Safety, Reliability, and Security: SAFECOMP 2014 Workshops* (2014), Springer, pp. 360–370.