Design-time models for resiliency

Andrea Marrella[1], Massimo Mecella[1], Barbara Pernici[2], and Pierluigi Plebani[2]

[1] Sapienza Università di Roma, Italy
[2] Politecnico di Milano, Italy

Abstract. Resiliency in process-aware information systems is based on the availability of recovery flows and alternative data for coping with missing data. In this paper, we discuss an approach to process and information modeling to support the specification of recovery flows and alternative data. In particular, we focus on processes using sensor data from different sources. The proposed model can be adopted to specify resiliency levels of information systems, based on event-based and temporal constraints.

1 Introduction

As information systems (ISs) are becoming more and more complex and interconnected, the information provided by the system and by other networked businesses and components can be of varying quality depending on the functioning of the modules of the IS itself, both at the hardware and the software level.

According to the *error-chain* paradigm described in [5], an erroneous situation in a system is not always evident, and becomes apparent when a *failure* occurs; such a failure may originate from different *error states*, which in turn are possibly originated by different *faults* in system components. Faults may be transient or permanent and they may be difficult to diagnose, in particular in the case of intermittent faults. Therefore, different ways of managing possible failures have to be considered, depending on the state of the system and on the possibly originating fault, and the different effects of the faults must be taken into account, with the goal of resuming normal functioning, or at least guaranteeing some limited functionality.

In this paper, we focus on designing ISs to make them resilient by design, i.e., considering the improvement of their reliability in case of expected and unexpected faults. In [26], we analyzed the characteristics needed for designing a *resilient information system* using a process-based representation of the IS. As in most process-based approaches to IS modeling, the data model is considered only marginally and often detached from the process model. However, in the design of a resilient IS, one important aspect are data and their temporal characteristics. As in the case of faults, also data faults may be transient or permanent, and different design approaches have to be considered depending on the type of fault and on the desired resiliency level.

Temporal aspects in conceptual modeling have been studied extensively in the literature (e.g., [18,7,24,25]), and in particular in connection with the representation of events. In this work, inspired by the classification of temporal features of entity types and relationships proposed by Olivé et al. in [7], we discuss its application in the different phases of designing resilient information systems.

The goal of this paper is to discuss the temporal characteristics of information in resilient ISs and to propose a model to support resilient IS design.

The rest of the paper is organized as follows. In Section 2, we introduce the motivations and our approach to IS resiliency, presenting their characteristics in Section 3. In Section 4, we introduce the proposed process and data models, discussing their application for designing systems of different resiliency levels in Section 5. Finally, in Section 6 we relate our work to the state of the art.

2   The approach

The approach underlying this work consists of the following building blocks:

- A designer provides a specification of the process s/he is dealing with in some process specification language, e.g., BPMN - Business Process Model and Notation, CMMN - Case Management Model and Notation, etc.
- In addition to such process specification, the designer provides an information model, detailing the data used in the process, both for routing the control flow (process data) and needed during tasks' execution.
- Such process model and information model are the main artifacts over which a *design-time resilience analysis* should be conducted, in order to identify possible breaches, propose alternative specifications, etc.

From an abstract point of view, our approach is a method that, given a process and an information model specifications, returns a quantification of how much such specifications allow the deployment of a resilient process, and possible guidelines for improving such specifications are offered. The availability of such a conceptual tool is quite important: organizations operate in ecosystems, in which each actor is a potential source of failures, and therefore the *awareness on resilience* is a critical element during the design of an IS and its applications.

Some previous work has attempted to analyze the issues of resilience, but generally satisfying resilience requirements is considered mainly as a run-time issue, as it is related to the ability to cope with unplanned situations: several approaches [27] have been proposed to keep business processes running even when some unplanned exceptions occur, by enacting countermeasures. Clearly, if we focus on what to do in case of a failure, this approach seems to be the only possibility. However, if we focus on what is affected when a failure occurs, some improvements can be done also at design-time. In [26] we proposed an approach for assessing the resilience of business processes modeled with CMMN, and assigning to the specification a value in the range [0..3] in order to measure how much the specification has been designed with resilience in mind. In this

work, we focus on providing more specific details on the conceptual modeling languages to be used during the specification, focusing in particular on representing temporal aspects which play an important role for resilience, as it will be shown in the following examples.

3   Resiliency scenarios

In this section, we describe some characteristics of IS resilience discussing some example scenarios (shown in Table 1), covering a broad variety of typologies of information systems with resiliency requirements. We take into consideration the information sources needed by the scenarios and their characteristics and possible actions to improve resilience. In the following of the paper we will discuss how to model and evaluate impacts of possible failures, focusing on resilient IS design.

In the first type of scenario, *periodic reporting* based on sensor data is considered. In this case, e.g., considered in [26], the input source is a stream of collected sensor data, and the system is based on the data collection interval and it can be influenced also by its frequency. Where data sources are considered unreliable, there is the need of investigating if other sources of information (current or historical) are available. In general, this is done at run time, to recover from failures on an ad-hoc basis, while anticipating failures at design time designing alternatives could make the recovery phase more rapid and systematic.

In the second scenario, inspired from the work presented in [13], *self-healing processes* with mechanisms for recovering from failures through partial rework are discussed. The approach, which uses planning techniques to derive repair plans, is based on modeling dependencies among data being used in business processes. In this case, the impact of erroneous data is evaluated, and pre-designed compensation and recovery tasks are used to dynamically generate recovery plans.

In *alerting systems*, such as for monitoring production in factories or in smart buildings (e.g., [37]), the focus is on monitoring the current situation to detect anomalies. As in the first case, alternative sources of information should be planned, and the timeliness of data is important.

Finally, a *data movement* scenario is considered, where moving large quantities of data takes time, and resilience could be weakened due to late/missing data. In this case, preparation strategies to anticipate possible failures can be envisioned, such as transferring data in advance, pre-selecting data, changing data location and redundancy policies [10].

At design time "what if" analyses of possible failures in data availability can support the design of more resilient systems. Other data quality properties could also be considered (data which are not accurate, that cannot be trusted, incomplete, and so on), however in the present paper we focus on their time-dependent characteristics and we attempt to derive a meta-model to support resilient IS design and to study the properties of the system being designed.

Type of scenario	Description	Critical input sources	Characteristics of sources	Compensation
Periodic reporting	Analysis of sensor data	Sensors data collection	Interval of collection, frequency	Alternative source, using data from previous periods
Self-healing processes	Tasks exchanging info / Results of previous tasks	External sources	Any type	Reworking tasks
Alerting system	E.g., factory monitoring (device)	Sensors data collection	Interval of collection, frequency	Alternative source
Data movement	Transferring large documents	Documents available from different locations	Documents do not vary in time	Alternative location

Table 1. Scenarios characteristics.

4 Modeling processes and data for resiliency

4.1 Modeling processes

Process modeling usually relies on imperative/procedural notations where the control flow represents the element around which the activities and the data are organized. In the recent period, BPMN[3] has emerged as the standard de facto notation used to model processes according to this activity-centric approach. As also mentioned in [11], activity-centric modeling notations, although intuitive and close to the way of thinking of the process modelers, suffers from some drawbacks especially when resilience needs to be captured. In fact, activity-centric modeling makes a clear distinction between the normal execution and the exceptional executions. In some way, the modeler is forced to identify which is the right way to enact the business process and to decorate this process with additional activities that might be enacted only if some particular situations occur. In case of resilient processes, the normal execution could be more complex and depends on several variables that could also be unknown at design time. For this reason, we suggest to revert to artifact-centric notations as they provide the required flexibility with the main limitation that the modeling could become cumbersome.

Among the several alternatives, like GSM, Declare, and many others [21], in this paper we adopt CMMN[4] as artifact-centric modeling language. With a lot of synergies with BPMN (they are both proposed by OMG), CMMN provides a set of constructs that help the modeler to design a business process in terms of states in which the activities can be (or cannot be) performed. Instead of imposing a

---

[3] cf. http://www.bpmn.org/
[4] cf. http://www.omg.org/spec/CMMN/1.0

specific control flow and of considering such flow as the ideal one, the modeler can define conditions under which the state of system can be considered acceptable and to enable, or disable, activities.
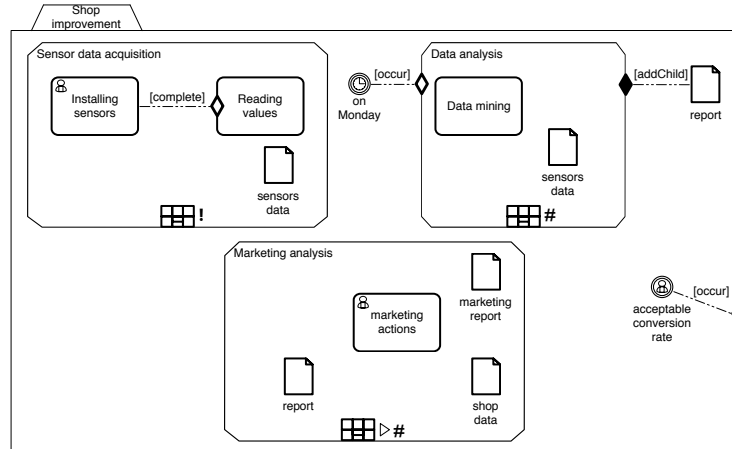

Fig. 1. CMMN model of an IoT-based case study [26].

In the next few paragraphs, the basic elements of CMMN useful to understand the content of this paper are introduced. To this aim, we refer to the example shown in Fig. 1 representing a real case study [26] concerning a process in charge of collecting data coming from a set of sensors. These sensors monitor the behavior of the customers inside a shop. Every week, these data are analyzed to create a report that constitutes the basis for creating marketing reports useful to identify marketing strategies (e.g., how to better distribute the items in the shelves, to identify the best products, and so on).

The main concept of CMMN is the *case* that is defined by the *case file* (data managed in a case), the *case plan* (how the case evolves), and the *case roles* (the stakeholders). Focusing on the first two aspects, CMMN does not focus on the order in which the activities are performed, but only on the dependencies between the different states of execution of the process based on information stored in the case file.

In more detail, a case plan (represented as a manilla folder and which must be unique for a given model) is a composition of stages (represented by a rectangle shape with angled corners). The stages represent the episodes of a case which, in turn, could contain other stages or tasks, i.e., atomic units of work. Stages and tasks can be defined as mandatory (with a solid border) or discretionary (with a dashed border) to identify which are the elements of the case that actors must or could execute. Tasks and stages can be further characterized by the entry and exit criteria represented by, namely, white and black diamonds. These criteria define when a task or a stage opens and when they can be considered

as closed. Finally, listeners (represented by circles) represent events that might occur during the execution of the case plan and that could determine the start or the end of a task or stage.

Concerning the information model, although this is a crucial element of artifact-based modeling notations, CMMN simply includes the possibility to specify data objects (typical document shape) without any specific restriction on the format or the content. If, from the one side, this leaves the freedom to consider any type of data, on the other side the model cannot express any type of data semantics. For this reason, as also proposed in [26], an extension of the information model where also discretionary data and a more rich set of association types between stage/task and objects are proposed.

Having quickly introduced the main elements of CMMN, and moving back to the case study in Fig. 1, here the case plan is composed by three main stages (i.e., sensor data acquisition, data analysis, and marketing analysis). While the data analysis starts every Monday and closes when a report is produced, the other two stages always run as neither entry nor exit criteria are defined. The entire case closes when the conversion rate (the ratio between people entering into the shop w.r.t. the people that buy some goods) becomes acceptable for the shop owner. Finally, as defined in the sensor data acquisition stage, it is possible to express some dependencies between the tasks. In fact, the sensor reading tasks start only when the sensors have been completely installed.

As previously mentioned, the information model provided by CMMN is not so rich. For this reason, we can simply add data objects to the stages to clarify which are the data that are considered (without any possibility to specify the nature of the operations on them) when a stage is running.

4.2 Modeling time varying information

As discussed above, CMMN provides a high level view on the data objects needed in the different stages. For supporting resiliency at design time, it is clear that more information is needed on the data being used in the process. We adopt a notation based on UML class diagrams to represent data objects, their components, and their relationships.

However, as discussed by Olivè et al. [7], the notation of UML class diagrams assumes that "the information base contains the current instances of entity and relationship types". On the other hand, temporal information about the available data for the process is essential in processes such as the one shown in Fig. 1. In fact, for instance, the report is produced periodically from sensor data that must be available for the period considered in the report. Sensor data also need to be defined, as they are acquired from sensors within the considered period, and therefore they are a time series taken as an input for a given stage. It has also to be noted that the process is continuing, producing reports periodically, until its exit goal is reached. Therefore, in general, several reports will be produced periodically, starting from different sensor data.

In Olivè et al. [24], two dimensions are considered: durability and frequency. The *durability* feature is used to describe if an entity type is instantaneous or
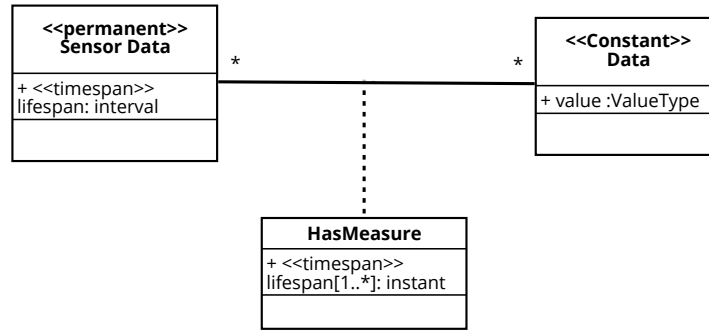
Fig. 2. Time series representation for sensor data.

durable. The *frequency* feature is defined so that "entity type E is single if all its entities can only be instance of E during one classification interval. Otherwise, E is intermittent."

Starting from the dimensions described above, six ways for classifying an entity or a relationship type wrt. its temporal features are proposed in [7]:

- *Instantaneous, single*, if two entity types can be related only once, at a given time.
- *Instantaneous, intermittent*. If two entity types can be related several times, at different time instants.
- *Durable, single*. Two entity types can be related for a period of time, only once.
- *Durable, intermittent*. Two entity types can be related for a period of time, several times.
- *Permanent*. A relationship once established does not change.
- *Constant*. For unchanging entities.

The representation of input sources for data in our case study of Fig. 1 can be represented using the notation proposed in [7], to be able to represent the temporal characteristics of entities used in the model. In Fig. 2, we show how the Sensor data source is formed: as it is a stream of data, it is considered a durable intermittent entity type in our case study, where the intervals indicate the periods in which data are gathered. Its lifespan is indicated as a set of intervals. Note that if the stream is always correctly working, it will contain a single interval from the beginning of the measurement to the current date. If an interruption in the stream occurs, a new lifespan interval is created instead. So the intervals indicate all periods in which Sensor data measurements are actually collected.

5   Analyzing resiliency data properties

During a generic process enactment, unplanned situations might occur. Depending on the nature of the raised issues, the magnitude of their impact varies and one or more activities may be involved. At the same time, different counter-measures can be taken to mitigate these negative effects. As an example, in a cyber-physical process for many reasons the sensors might not be able to send data, and an alternative source of information might be considered, to be able to equally infer relevant behaviors to be monitored, or at least in a slightly degraded form. Another example is about frequencies (temporal constraints): a sensor might not be able to send information very frequently, still alternative techniques can be adopted to infer missing data or to calculate aggregated measurements. Similarly to what is usually done in emergency management [32,20], where a *preparedness phase* aims to improve the systems by learning from the previous emergencies, in [26] we propose an approach which helps the process designers in improving their process models by considering the previous experiences in failures generated by data unavailability. For this, the ability to model the process and the information model and possible temporal constraints for the purpose of resilience awareness is crucial.

Having clearly modeled the process and the information model, allows the development of an approach to categorize resilience characteristics, then to define resiliency levels, and to model the resilience improvement aspects from a modeling perspective. With the modeling approach previously introduced, it is possible to analyze a (possibly multi-party) business process resilience from a *data perspective*: data dependencies among the involved parties and relationships between process activities 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. We are able to consider *Parties*, i.e., actors involved in the process. Then we deal with *Tasks*, i.e., units of work performed by parties, which consume data as input and produce data as output. The data produced by a task must be required by at least another party, and *Data*, i.e., 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. Data can also be used to verify the entry and exit conditions, thus to realize when a stage or task starts or terminates. As previously introduced, the temporal constraints over read/write of data are crucial in modeling resilient processes.

Resilience of processes depends on both the reliability of the tasks and the lack of data availability. The *reliability of the task* concerns the possibility that one or more tasks cannot be executed: i.e., the required infrastructure to perform the job is not available, also including the human resources for which the unavailability of data can block the execution of manual tasks. On the other side, *lack of data availability* (including a wrong frequency of the data) 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 the task reliability, as all the tasks by definition produce data and these data are relevant for at least

one of the participating parties, and problems on tasks may have also the side effect to make data unavailable. Moreover, there are situations in which tasks are properly working, 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 quality parameters through which we can define the acceptable level of data quality for considering the data available [6]. For this reason, the definition of the data could be coupled with the definition of quality levels that are considered acceptable for a task that is using such data.

Having modeled the process and the information model, it is then possible to define levels of resilience on the basis of the ability of the process to adjust the possible unexpected failures. We aim to classify the way resilience can be considered and obtained, in terms of preparedness to unexpected events which might be caused or have impact on data availability. In particular, the following four levels of designed resilience have been identified:

- **Level 0 – None.** At this level business processes are designed without taking into account the data unavailability that might cause failures during the execution. As a consequence, also 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 to define the agreement between the parties, no support is given to the 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, so there will be the need to make it resilient. In this work, we consider failures caused by data unavailability, which might impact on one or more 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 are the relevant data 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 – Identifying alternatives for data and goals.** For processes classified in this level, the model of the process makes an initial attempt to overcome possible failures, whose nature and impact have been defined with the previous level. In more detail, there are two aspects to be taken into account:

- *Alternative Data*: based on the information about the source of failures and the potential impact of these failures, the designer can decide to include in the process model the alternative data. In this way, starting from the data having more probability of failures 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. The issue of quality of data has been extensively addressed in traditional information systems, e.g., [6], but the quality of big data (which includes sensor-generated data) is still to be precisely defined [12].

- *Alternative Goal*: as the process resilience implies to mitigate the effect of a failure, a possible mitigation includes revising the initial expectations of the process to achieve a given goal. The designer defines, for each party, a new goal that represents a status that can terminate the execution of the process in an acceptable way. If the initial goal corresponds to the optimal goal, the alternative goal could be considered as a best-effort goal.

It is worth noting that the business process models at this level do not prescribe any specific actions to cope with the failures at run-time. For this reason, a model at this level only supports who is in charge of executing the process, to select, in case of failures, new data sources as well as to decide to consider satisfactory the result of the execution even if the initial goal is not possible to be fulfilled, accepting a weaker goal.

In [26], we propose an extension to CMMN to represent *alternative data*, represented with document shaped icons with dashed borders as shown in Fig. 3.
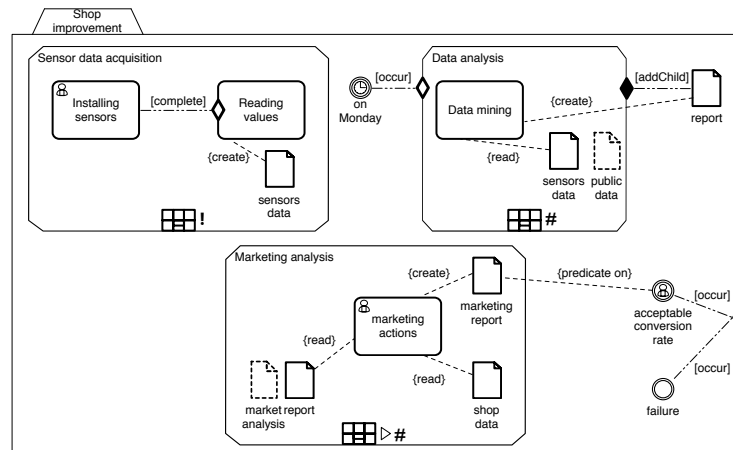

Fig. 3. CMMN model extended with alternative data [26].

Using the data model introduced in Section 4.2, we can represent also the relationships between entities in the process. As shown in Fig. 4, the Report to be produced for a given interval is based on the Sensor Data for a period, but can also use as an alternative Public data available for the period of interest of the report. The source of data that will be actually selected (either Sensor Data or the alternative Public Data) will become permanently liked to a given report using it.
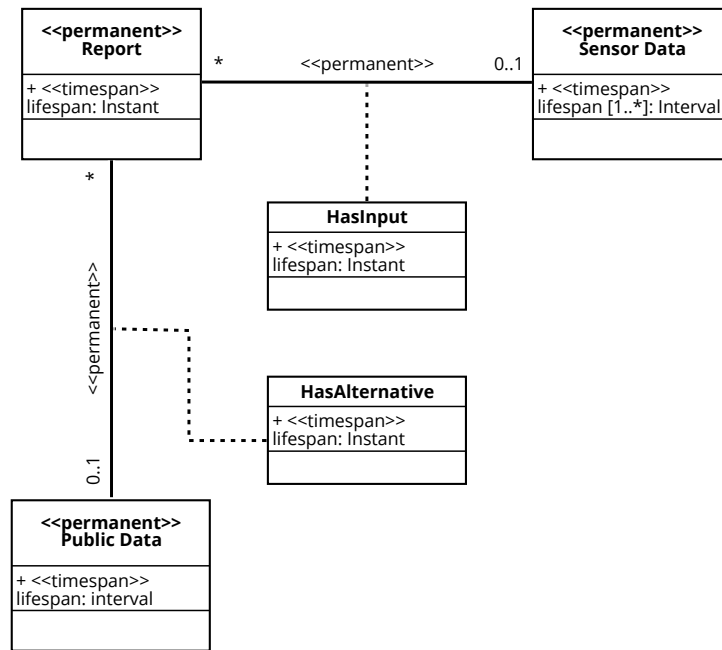
Fig. 4. Alternative data sources representation.

– **Level 3 – Defining alternative actions.** At this level, processes have been designed by considering also 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 goal), the designer can embed in the business process how these alternatives could be effectively managed. New 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 equivalent to the original service.

With these levels of resilience, we aim at supporting the process designer in understanding if the resilience is modeled, and if 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.

6 Related work

Research on *resilient systems* encompasses several disciplines, such as psychology [34], ecology [14], sociology [3] and engineering [17]. In information systems, *resilience engineering* has its roots in the study of safety-critical systems [17], i.e., systems aimed to ensure that organizations operating in turbulent and interconnected settings achieve 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 existing system's design [23]. In the BPM field, cf. [23] and [28], 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* [27] and *risk-aware BPM* [31,30].

On the one hand, research on process flexibility has focused on four major flexibility needs, namely *(i) variability* [15,16,27], *(ii) looseness* [2,19], *(iii) adaptation* [29,22], and *(iv) evolution* [8,9]. The ability to deal with changes makes process flexibility approaches a required but not sufficient mean for building resilient BPM systems. In fact, there exists 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 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 on the "realized risks" and their consequences, to improve risk prevention and mitigation, and therefore aim at complementing conventional risk-aware approaches.

Surprisingly, the fact is that there exists only a limited number of research works investigating resilience of BPM systems [4,36,35], and they are all at conceptual level. For example, the work of Antunes and Mourao [4] derives a set of fundamental requirements aimed at supporting resilient BPM. The approach of Zahoransky et al. [36] investigates the use of process mining [1] to create probability distributions on 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 [35] 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.

In our previous work [26], we started to approach IS resiliency in a systematic way, with the goal of defining possible levels of resiliency, and of investigating which models, or variants of models, can support the design of resilient IS.

Considering conceptual models, there is a specific need for conceptualizing the evolution of the information. To this purpose, we considered previous work on the representation of time and events at conceptual level. The fundamental concepts for representing temporal data have been discussed and presented in [18]. In the direction of conceptual modeling, the information systems group lead by Antoni Olivé has given an important contribution, studying possible extensions for modeling temporal aspects of information and events in IS design (see e.g., [7,25,24]. Part of the work presented in this paper has been based on the classification and notations proposed by Olivé et al. for extending UML with temporal features.

If compared with the aforementioned works, our research aims to provide guidelines to model resilient-by-design business processes by focusing on the data exchanged between the activities composing the process, an aspect neglected in the existing approaches to process resilience. In this work, we have tackled the issue of modeling the temporal dimension of the data, in order to have a coherent approach both in the process and in the information model.

7 Conclusions and future research lines

Adopting design-time models to represent resilience aspects allows the IS designer to take a "preparedness" approach, to anticipate what should be done in case of possible occurring or anticipated failures, to guarantee a certain level of resilience. Following a continuous improvement approach, we propose to analyze temporal features of data over time evaluating past failures of the system. Further work is needed in this direction, with the goal of improving resilience. A focus on a preparedness phase can help improving the models by learning from previous failures. Further modeling and analysis techniques are also needed to represent possible interferences among processes, as unexpected consequences may arise over time from apparently unrelated processes [33].

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer (2016)

2. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science - R&D 23(2) (2009)
3. Adger, W.N.: Social and ecological resilience: are they related? Progress in human geography 24(3), 347–364 (2000)
4. Antunes, P., Mouro, H.: Resilient Business Process Management: Framework and services. Expert Systems with Applications 38(2), 1241 – 1254 (2011)
5. Avizienis, A., Laprie, J., Randell, B., Landwehr, C.E.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Sec. Comput. 1(1), 11–33 (2004)
6. Batini, C., Scannapieco, M.: Data and Information Quality - Dimensions, Principles and Techniques. Springer (2016)
7. Cabot, J., Olivé, A., Teniente, E.: Representing temporal information in UML. In: UML 2003 - The Unified Modeling Language, Proc. 6th Intl. Conference on Modeling Languages and Applications, San Francisco, CA, USA. pp. 44–59 (2003)
8. Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow evolution. Data & Knowledge Engineering 24(3), 211–238 (1998)
9. Dadam, P., Rinderle, S.: Workflow evolution. In: Encyclopedia of Database Systems, pp. 3540–3544. Springer (2009)
10. D'Andria, F., Field, D., Kopaneli, A., Kousiouris, G., García-Pérez, D., Pernici, B., Plebani, P.: Data movement in the internet of things domain. In: Service Oriented and Cloud Computing - 4th European Conference, ESOCC 2015, Taormina, Italy, September 15-17, 2015. Proceedings. pp. 243–252 (2015)
11. Fahland, D., Lübke, D., Mendling, J., Reijers, H.A., Weber, B., Weidlich, M., Zugal, S.: Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In: 10th Int. Workshop on Business Process Modeling, Development, and Support (BPMDS). pp. 353–366 (2009)
12. Firmani, D., Mecella, M., Scannapieco, M., Batini, C.: On the Meaningfulness of "Big Data Quality". Data Science and Engineering 1(1), 6–20 (2016)
13. Friedrich, G., Fugini, M., Mussi, E., Pernici, B., Tagni, G.: Exception handling for repair in service-based processes. IEEE Trans. Software Eng. 36(2), 198–215 (2010)
14. Gunderson, L.H.: Ecological Resilience–in Theory and Application. Annual review of ecology and systematics pp. 425–439 (2000)
15. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. Journal of Software Maintenance and Evolution: Research and Practice 22(6-7) (2009)
16. Hallerbach, A., Bauer, T., 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)
17. Hollnagel, E., Woods, D.D., Leveson, N.: Resilience Engineering: Concepts and Precepts. Ashgate Publishing, Ltd. (2007)
18. Jensen, C.S., Dyreson, C.E., Böhlen, M.H., Clifford, J., Elmasri, R., Gadia, S.K., Grandi, F., Hayes, P.J., Jajodia, S., Käfer, W., Kline, N., Lorentzos, N.A., Mitsopoulos, Y.G., Montanari, A., Nonen, D.A., Peressi, E., Pernici, B., Roddick, J.F., Sarda, N.L., Scalas, M.R., Segev, A., Snodgrass, R.T., Soo, M.D., Tansel, A.U., Tiberio, P., Wiederhold, G.: The consensus glossary of temporal database concepts - february 1998 version. In: Temporal Databases, Dagstuhl. pp. 367–405 (1997)
19. Marrella, A., 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). pp. 277–291 (2013)

20. Marrella, A., Mecella, M., 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., Russo, A., Steinau, S., Andrews, K., Reichert, M.: Data in business process models, A preliminary empirical study. In: 8th IEEE SOCA 2015, Rome, Italy, October 19-21, 2015. pp. 116–122 (2015)
22. Marrella, A., Mecella, M., Sardiña, S.: Intelligent Process Adaptation in the SmartPM System. ACM TIST 8(2) (2017)
23. Müller, G., Koslowski, T.G., Accorsi, R.: Resilience - A New Research Field in Business Information Systems? In: 16th Int. Conf. on Business Information Systems (BIS). pp. 3–14. Springer (2013)
24. Olivé, A.: Relationship reification: A temporal view. In: Advanced Information Systems Engineering, 11th International Conference CAiSE'99, Heidelberg, Germany, June 1999, Proceedings. pp. 396–410 (1999)
25. Olivé, A.: Definition of events and their effects in object-oriented conceptual modeling languages. In: Atzeni, P., Chu, W.W., Lu, H., Zhou, S., Ling, T.W. (eds.) Conceptual Modeling - ER 2004, 23rd International Conference on Conceptual Modeling, Shanghai, China, November 2004, Proceedings. Lecture Notes in Computer Science, vol. 3288, pp. 136–149. Springer (2004)
26. Plebani, P., Marrella, A., Mecella, M., Mizmizi, M., Pernici, B.: Multi-party business process resilience by-design: A data-centric perspective. In: Proc. Conference on Advanced Systems Engineering, CAiSE'17. Springer (June 2017)
27. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies. Springer (2012)
28. Rosemann, M., 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)
29. Sadiq, S., Orlowska, M.: On Capturing Exceptions in Workflow Process Models. In: 3rd Int. Conf. on Business Information Systems (BIS), pp. 3–19. Springer (2000)
30. 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), 933–984 (2014)
31. Tjoa, S., Jakoubi, S., Goluch, G., Kitzler, G., Goluch, S., Quirchmayr, G.: A Formal Approach Enabling Risk-Aware Business Process Modeling and Simulation. IEEE Transactions on Services Computing 4(2), 153–166 (2011)
32. Van De Walle, B., Turoff, M., Hiltz, S.R.: Information Systems for Emergency Management. M.E.Sharpe (2009)
33. Vitali, M., Pernici, B.: Interconnecting processes through IoT in a health-care scenario. In: IEEE International Smart Cities Conference, ISC2 2016, Trento, Italy, September 12-15, 2016. pp. 1–6 (2016)
34. Yates, T.M., Masten, A.S.: Fostering the Future: Resilience Theory and the Practice of Positive Psychology. John Wiley & Sons Inc (2004)
35. Zahoransky, R.M., Brenig, C., Koslowski, T.: Towards a Process-Centered Resilience Framework. In: 10th Int. Conf. on Availability, Reliability and Security (ARES). pp. 266–273. IEEE (2015)
36. Zahoransky, R.M., Koslowski, T., Accorsi, R.: Toward Resilience Assessment in Business Process Architectures. In: Computer Safety, Reliability, and Security: SAFECOMP 2014 Workshops. pp. 360–370. Springer (2014)
37. Zhou, S., Lin, K., Na, J., Chuang, C., Shih, C.: Supporting service adaptation in fault tolerant internet of things. In: 8th IEEE International Conference on Service-Oriented Computing and Applications, SOCA 2015, Rome, Italy. pp. 65–72 (2015)