Knowledge-Aware Operational Support

Deliverable D1.1
DOLCE Extension for Business Processes

Project Acronym	KAOS	
Project Title	Knowledge-Aware Operational Support	
Project Number	Interregional Project Network IPN12	
Workpackage	WP1 – Foundations of Organizational Modeling	
Lead Beneficiary	FBK	
Editor(s)	Emilio M. Sanfilippo	CNR-ISTC-LOA
	Stefano Borgo	CNR-ISTC-LOA
	Nicola Guarino	CNR-ISTC-LOA
Contributor(s)	Chiara Ghidini	FBK
	Chiara Di Francescomarino	FBK
	Adamo Greta	FBK
Dissemination Level	PU	
Contractual Delivery Date	30/07/2017	
Actual Delivery Date	10/11/2017	
Version	2.0	

This work is supported by the *Euregio Science Fund* under *Interregional Project Network IPN12*, which is funded by the *European Region Tyrol–South Tyrol–Trentino* (EGTC) under the first call for basic research projects.

1 Introduction

Organisations are complex entities that can be described from different perspectives: social, structural, functional, procedural etc. From the social perspective an organisation is often seen as a multi-agent system with a coordinated and supervised plan which depends on the organisation's goals. Since an organisation interacts with the world via the actions of physical agents (e.g., an employee), who act on its behalf, it is natural to describe the structure of the organisation in terms of the roles it defines.[1] The agents participate to the processes of the organisation in force of their roles which, in turn, define the goals to be achieved, tasks to be performed, and resources to be used or provided. The dependencies across roles (at the modelling level) as well as across physical agents (at the implementation level) are in large part directed by the set of goals to be achieved, and influence the processes within the organisation.

These connections need to be analysed and modelled if we aim to make explicit the levels of trust, delegation, and responsibility in the organisation.

Regarding the functional perspective, every process definition is aimed to implement a desired input-output relationship via a behaviour. It is this behaviour that the process model casts in formal terms. Usually, the process in an organisation has to consider a variety of entities: the involved physical objects, the information objects, their qualities and how they change during the process, and the people that act at the different phases of the process. Standard modelling languages lack this integrated perspective loosing the holistic view of the organisation's processes [Adamo et al., 2017]. This makes it harder to explain the motivations for some business processes' existence and structure.

This deliverable aims to develop a foundational framework of concepts focusing on organisations, processes, participants and information. To exploit the interplay among the business process executions, the participants (agents and material resources), the manipulated data, and the organisational (role) structure we need to develop a suitable ontology fragment for this domain. Starting from the foundational perspective of the Dolce ontology [Borgo and Masolo, 2013], we take the BPMN standard[2] as a paradigmatic process modelling language to show which further elements should be considered to properly model the relevant structure of the organisation, the process, the physical domain (the agents and physical objects that participate in the process), and the information domain (data, metadata and their dependences). This conceptual structure will make possible to coherently link the business process executions to domain knowledge, such as the structure of the organisation, the strategic goals, the role of the participants and resources, as well as the rules according to which the processes can be explained or motivated. It will also enable to analyse (and possibly discover) activities and/or data that escape or are only partially supported by the implemented IT systems, and to set the framework to compare business process executions and models at different levels of granularity.

The deliverable is organised in two parts. The first, corresponding to Task 1.1 of the project proposal, introduces the key notions for the business process environment. The second, relative to Task 1.2, further expands this material into an extension of DOLCE.

In the following sections we use the procedural language BPMN, which we introduce next, for exemplification. Our analysis is driven by general ontological considerations and can be applied to other procedural languages like UML-Activity Diagram[3] (UML-AD) and the Event-driven Process Chain [Scheer, 2002] (EPC). Similar considerations hold also for declarative languages like CMMN.

2 BPMN

The Business Process Modelling Notation (BPMN) is a standard business language, proposed by the Object Management Group (OMG)[4] to design business processes (BP). There are five categories of graphical elements in BPMN: (*i*) flow objects, (*ii*) data objects, (*iii*) connecting objects, (*iv*) swimlanes and (*v*) artifacts.

Flow objects are the core elements used to model the units of work to be performed in order to reach a desired goal. They are divided into *activities*, *events* and *gateways*. An activity represents "a work that is performed within a business process" [(OMG), 2011, p.29] and can be either atomic, in which is called *task*, or non-atomic, called *sub-process*, depending on whether it comprises other activities. An event "is something that "happens" during the course of a process [...] and usually has a cause or an impact [i.e., result]" (ibid., p.233). Depending on their position in the process flow, events divide into *start events*, *end events*, or *intermediate events*. Additionally, they are classified into *throw* and *catch* events, although BPMN lacks a precise definition

[1] The notion of social role has been studied from a foundational viewpoint in [Masolo et al., 2004].

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

[3] `http://www.uml-diagrams.org/activity-diagrams.html`

[4] `http://www.omg.org/`

for this distinction. The intuition is that the former are used to model events in which that process flow "delivers something", whereas the latter stand for events in which the process flow "receives something". For example, a message-throw-event is an event in which a message delivery is done, whereas a message-catch-event is an event in which a message reception is done. Finally, gateways are graphical elements used to mark forking, merging and joining of work flows.

The various flow objects are linked to each other through *connecting objects* (e.g., sequence flows). *Swimlanes* are used to represent who is responsible to execute the depicted process, e.g., a person, an organization or a specific department within an organisation. More precisely, BPMN distinguishes between *pools* and *lanes*; the former are containers for flow objects taken on by a certain entity, whereas lanes are used to subdivide pools.

Data objects are elements used to represent information that is (generally speaking) manipulated during the execution of a process. Finally, BPMN provides *artifacts* to describe groups and text annotations. Groups are useful to graphically cluster elements belonging to the same category, whereas text annotations are used to specify additional text that can be valuable to the user of the diagram.

Fig. 1 shows an example of BPMN business process model with four participants depicted by means of different pools: *Purchaser*, *Service provider A*, *Service provider B* and *Service provider C*. The process starts when the event *None* in the *Purchaser* pool happens. This is followed by the execution of task *Request Quotes* which ends with the sending of a message to each service provider participant in the process. Once a service provider receives the message, it starts its own process consisting in sending a quote to the *Purchaser* (the task *Send Quote*). After this, the service provider process ends. When the *Purchaser* receives at least two quotes (the gateway marked with an asterisk), it executes the *Assess the Quotes* task after which the process ends for the *Purchaser* provided the condition *Sufficient reserve amount?* of the last gateway is satisfied. Otherwise, the process is back to the *Request Quotes* and flows again as described above.

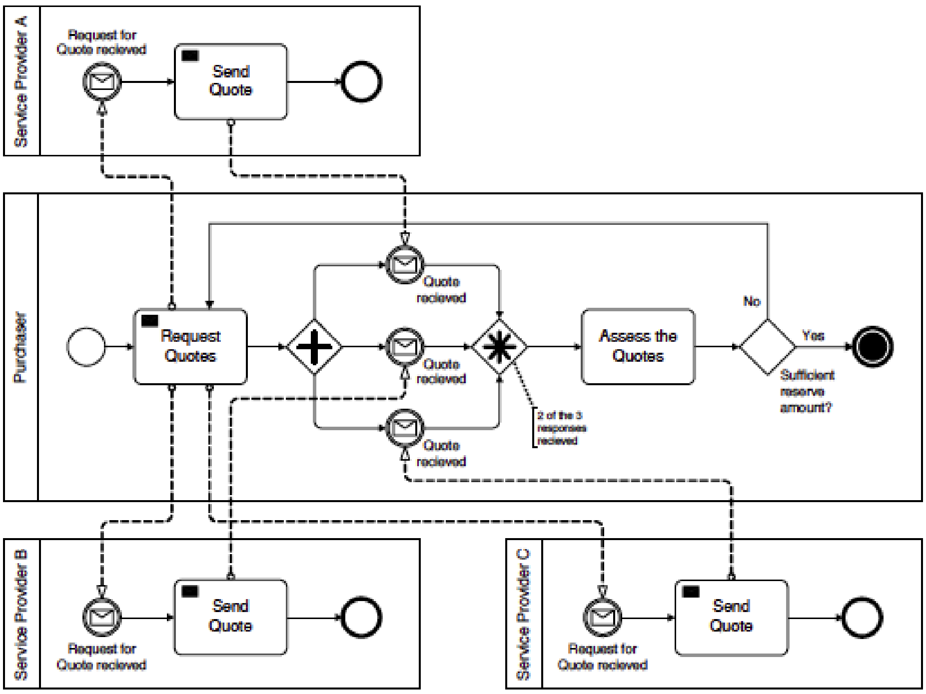

Figure 1 – Example of BPMN process model (from [(OMG), 2011]). Circles are events; rectangles are tasks and diamonds are gateways. Solid arrows indicate the process flow within a pool (identified by a rectangle labeled on the left); dashed arrows indicate interactions across pools.

Note that the model in Fig. 1 does not explicitly represent the data objects handled by the process. For instance, the *Request quotes* task is meant to send a message to the service providers, but the message is only implicitly represented within the task type. A different modelling approach is depicted in Fig. 2, which represents the process of booking a flight ticket, where data objects are explicitly encoded.

Despite its large application, the main focus of BPMN is on graphical constructs rather than on formal semantics. This choice has some well-known drawbacks: BPMN presents conceptual ambiguities regarding the

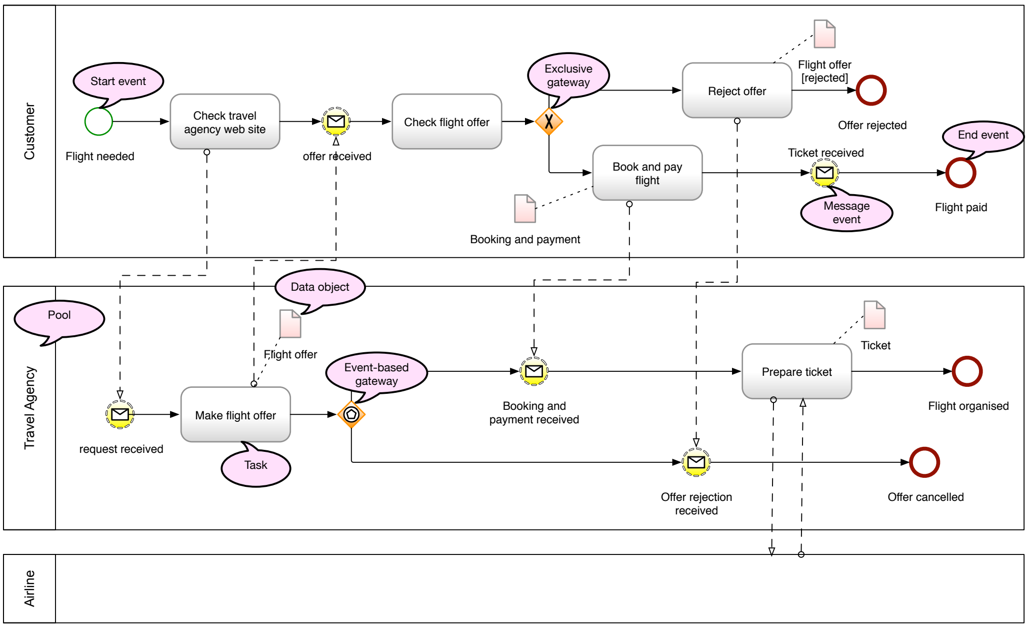

Figure 2 – Example of BPMN process model

interpretation of its metamodel;[5] the supporting software tools are not guarantee to interoperate and there is no guideline about how to use the modelling language in ontology-based systems. Consider, e.g., the differences in the intended meanings of the pools depicted in Fig. 2, where *travel agency* and *airline* [company] stand for organisations, and *customer* for a social role. It is evident that pools can be interpreted in different manners, whereas BPMN provides no guideline on how pools should be understood and coherently used.

Different communities, in particular within the Semantic Web and knowledge management domains, have been working on BPMN-like ontologies for disparate application purposes [Bertolazzi et al., 2001, Rospocher et al., 2014, Wong and Gibbons, 2011, Kossak et al., 2014]. However, due to BPMN conceptual ambiguities and the differences across the communities, no credible candidate has yet emerged as the ontological counterpart for BPMN. In particular the analysis of participants, among which agents, organisations and information is missing.

3 DOLCE

The *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) [Masolo et al., 2003] is an upper-level ontology explicitly designed to capture the ontological categories underlying natural language and commonsense thinking. DOLCE is part of our methodology for the analysis of business process knowledge, because it is based on a worldview that meets modelling requirements in the BP domain, e.g., the distinction between events and the objects that take part, execute or are created in events. Additionally, it has been already employed for the analysis of BP notations [Sanfilippo et al., 2014, Adamo et al., 2017]. Currently, three versions of DOLCE are available:

1. DOLCE [Masolo et al., 2003], the entire axiomatisation of the ontology in first-order modal logic;
2. DOLCE Core [Borgo and Masolo, 2013], which comprises only the core categories of DOLCE in first-order logic;
3. DOLCE Light[6], which consists in the partial axiomatisation of DOLCE in the Web Ontology Language (OWL) [Baader et al., 2003].

For the purposes of our work, we rely on DOLCE-Core (taxonomy shown in Fig. 3)[7] since it provides the basic modelling elements we need. As we will explain, we extend DOLCE-Core to meet the modelling requirements of

[5] With 'metamodel' we refer to the basic elements used in BPMN.
[6] `http://www.loa.istc.cnr.it/old/DOLCE.html`
[7] The taxonomy does not consider the class ARBITRARY SUM, which is not relevant for our purposes. Additionally, it does not show the relations between the classes.

the KAOS project. Foundational ontologies like DOLCE are large and complex systems; hence, we just provide here a minimal introduction. The reader can refer to [Borgo and Masolo, 2013] for a throughout discussion of technical aspects.

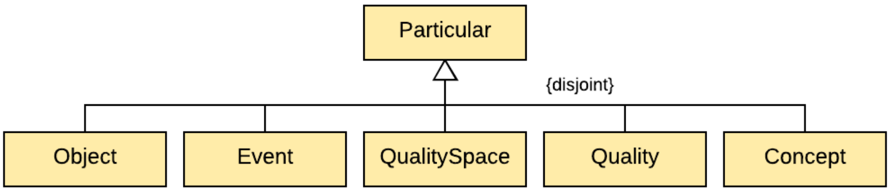

Figure 3 – DOLCE-Core

DC:PARTICULAR[8] is the most general class covering any entity in the quantification domain. Roughly speaking, particulars are entities that exist in time, like the Pisa tower, the event of climbing Cerro Torre on a specific day performed by certain alpinists, or the weight of John's car.

Objects are entities that are primarily present in space,[9] differently from events, which primarily exist in time.[10] Objects and events are linked via the relationship of DC:PARTICIPATION: object o participates in the event e at time t when the event e happens during t and o is one of the elements involved in the happening of e.

Instances of both DC:OBJECT and DC:EVENT can be characterised with a bunch of *individual qualities*, which are partitioned into *quality kinds*, i.e., classes of maximal comparable qualities, e.g., the colour-kind, the shape-kind or the weight-kind. In DOLCE-Core qualities satisfy the so-called *non-migration principle*, i.e., they are associated with *one and only one* entity, therefore cannot characterise (*inhere in*) different entities. For example, the object car_1 has its own individual quality of the weight-kind, which we can measure by means of some measurement device according to a reference measurement system. Following this reasoning, the individual car car_2, different from car_1, bears its own weight-quality, which is different from car_1's weight no matter whether the two are associated to the same value, say, $500kg$. Indeed, DOLCE-Core distinguishes a quality from its value, that is, the weight-quality of car_1 is distinct from its value $500kg$. Values can be provided according to different scales; e.g., weight qualities by means of the Imperial System or the International System of Units. To make sense of this, DOLCE-Core refers to *quality spaces*, which provide formal means, possibly embodied with geometric or topological structures, to organise (quality) values according to measurement devices or cognitive systems. The relationship of DC:LOCATION links a quality to its value (i.e., a region in a quality space) at a certain time. For example, if cars car_1 and car_2 are both $500kg$ heavy, in DOLCE-Core this means that their different weight-qualities, wq_1 and wq_2, respectively, are located in the same $500kg$ region in the corresponding weight-space, whose structure is provided according to the International System of Units (see Fig. 4).

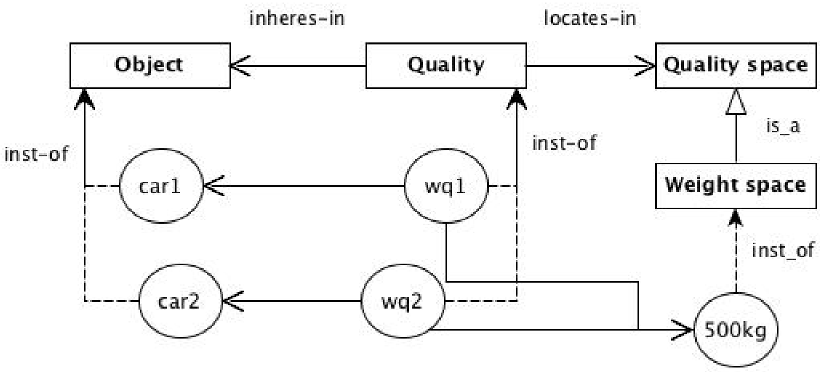

Figure 4 – Example of quality representation according to DOLCE-Core

Finally, DOLCE-Core includes *concepts* amongst particulars, which are fundamental to represent (social) *roles*, as we will see in Sect. 4.2. The relationship of DC:CLASSIFICATION is used to talk about the instances of

[8]DOLCE-Core classes and relationships are prefixed with DC.

[9]'To exist' and 'to be present' will be interchangeably used across the deliverable.

[10]The DOLCE Core notion of event does not have to be confused with the notion of event in the business process literature. We shall provide some clarification about their relationship throughout the deliverable.

a concept; e.g., CF(*student*, *John*, t) represents John as an instance of the concept-role *student* at time t, where the temporal parameter in CF refers to the time at which John is present.

4 Ontological analysis

In the following sections, we dig into the ontological analysis of some entities commonly represented in BPMN models. The purpose is to develop a general conceptual framework that is tuned on BPMN and can therefore support its analysis and coherent usage. In Sect. 5 we show the extension of DOLCE-Core on the basis of the notions hereby introduced, while in Sect. 6 we analyse the BPMN constructs relevant for our purposes.

4.1 Activities and Events

We saw in Sect. 2 that BPMN distinguishes between activities and events. The distinction is not however plain clear in all cases. Consider, for instance, *send tasks* and *throw events*. According to their intended semantics, both are to be used to model the delivering of some entity e.g., a message. There is not however a rationale for preferring the use of one construct over the other. The same consideration is true for *receive tasks* and *catch events*, whose intended meaning is to capture an entity in the context of the executed process. These ambiguities lead to different modelling approaches. On the one hand, e.g., Kossak and colleague [Kossak et al., 2014] suggests to avoid the use of send/receive tasks altogether because "their semantics, as described in the BPMN standard, does not significantly differ from the semantics of [...] throwing and catching [...] events". On the other hand, we saw that in both Fig. 1 and Fig. 2 send tasks are used instead of catch events.

The BPMN Quick Guide[11] suggests to prefer the use of events, instead of tasks, when the performed work is meant to be *instantaneous*. The Guide adds the following comment: "From a temporal perspective, an Event maps to a time point on a time line and a Task maps to a time interval". This consideration deserves some attention to understand whether it can coherently support the interpretation of the modelling constructs at hand, and therefore the definition of some guideline for the users of BPMN. As we saw in Sect. 2, indeed, BPMN activities distinguish into tasks and subprocesses depending on their structure, i.e., whether they comprise further activities.[12] The Quick Guide now tells us that activities map to time intervals. How do the notions of **atomicity** and **temporal extension** relate? BPMN does not include a theory of time; therefore the Quick Guide reference to time intervals has to be taken with a pinch of salt. Apart from atomicity and temporal extension, some authors [Sanfilippo et al., 2014, Kossak et al., 2014] propose to look at the distinction between *receive tasks* and *catch events* in terms of endogenous and exogenous entities, respectively, a distinction we shall consider as well.

Atomicity. We propose to distinguish between two meanings of atomicity, namely, what we call **ontological** and **modelling** atomicity. The former regards domain entities as they exist in the application domain at stake, whereas the latter is a matter of modelling abstraction.[13] From the latter perspective, for an entity e to be atomic means that, given a modelling context c, e is represented as not comprising further entities in c, whereas it may be even compound at the ontological level.

Consider, e.g., the activity *Request quotes* in the *Purchaser* Pool of Fig. 1. It is specified as a BPMN task. Reasonably, however, in order to request a quote the purchaser has to execute other activities, e.g., *Selecting providers* or *Write request*. At the level of the model one may assume that these activities are not relevant and therefore are not represented. From this consideration, a BPMN **send task** may be ontologically compound even though it is atomic in a model. In our understanding, a similar consideration can be done for **receiving tasks**; e.g., *Receiving an email* may mean—from an ontological stance—*Entering email account*, *Checking incoming emails* and so on.

Looking now at **events**, differently from activities, their internal structure is never depicted in BPMN models. The intended meaning is that they are always atomic. However, similarly to activities, one may assume that events atomicity is only a matter of modelling abstraction, whereas their occurrence in reality is compound. For example, a *Requesting quote* throw event may comprise further events which are not represented in a certain model just for application reasons.

[11] Available at `http://www.bpmn.org`, last access in June 2017.

[12] Recall that, differently from subprocesses, tasks are atomic.

[13] Atomicity concerns the structure of an entity with respect to its parts; hence, talking of entities being atomic or non-atomic (i.e., *compound*), we sometimes refer to their *mereological structure*.

Temporality. In order to make sense of the temporal dimension of BPMN activities and events, we need to take into account a clear cut-off distinction between the mereological structure of events ('event' in the sense of DOLCE Core) and the structure of time regions. In this sense, parthood relations are inter-categorial, i.e., they are between either events or time regions, but not both. Additionally, we have to assume that the structure of the former does not coincide with the structure of the latter, therefore atomic events may unfold through extended (non-instantaneous) temporal regions.

Looking at Fig. 5, e_1 is an atomic and instantaneous event; e_2 is atomic but temporally extended (the time region spans from t_{2_s} to t_{2_e}); e_6 is compound and occurs over a distributed (non-instantaneous) time region; e_5 is also compound but instantaneous.

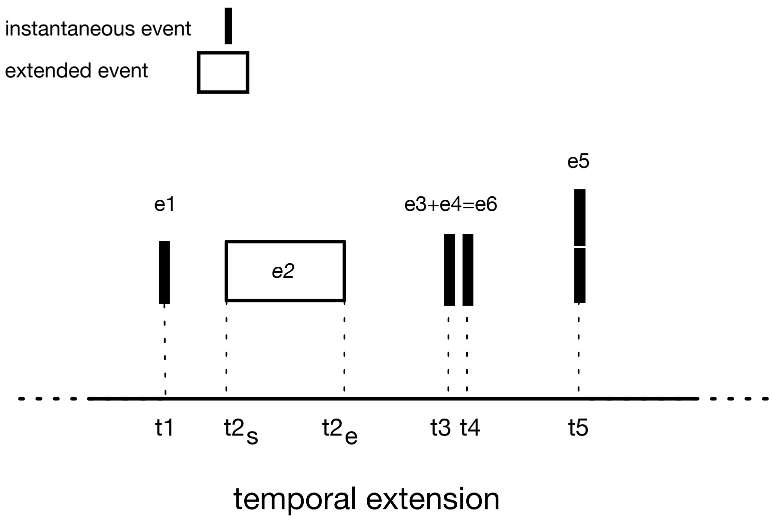

Figure 5 – Temporality of events

Endogenous and exogenous events. An event is said to be endogenous, rather than exogenous, when it occurs under the control of a certain agent. For example, the event of John writing an email is endogenous from John's perspective, because it happens under John's control; differently, the event of someone knocking on John's office door is exogenous from John's perspective, because it happens outside John's control. It should be clear that the endogenous *vs* exogenous dichotomy can be assessed only within a certain context. We will see in Sect. 6 how this distinction applies to BPMN.

4.2 Agents, social roles and organisations

We saw from the previous section that BPMN allows for the specification of business entities like *purchaser* or *service provider* (see Fig. 1). At the same time, however, the modelling language leaves open to interpretations whether such entities refer to individuals or *roles* that individuals bear within certain contexts. Also, there is no mean in BPMN to distinguish between individual persons and *organisations*, e.g., John from Apple Inc., although this difference is of fundamental relevance in business modelling scenarios. In order to cover these gaps, we now explore the ontological characterisation of social entities. For this purpose, we rely on the extension of DOLCE to some aspects of the social reality presented in Bottazzi and Ferrario [Bottazzi and Ferrario, 2009] and further developed in [Porello et al., 2014].

First, we introduce a primitive notion of agent. Agency is largely debated across AI, cognitive sciences and philosophy where different but not well-integrated theories have been proposed. Following [Russell and Norvig, 1995], we assume that an agent is an entity with sensors, actuators and the capability to act on itself or on the environment. From this perspective, an agent bears a decision module upon which it acts.[14] Human beings are examples of agents to which intentionality is ascribed. In a manufacturing scenario, which can be partially represented by BPMN diagrams, a lathe machine is an example of non-intentional (artificial) agent when, e.g., it has sensors by which it acquires data from the objects to be manufactured and acts upon them by elaborating the data.

[14]We are intentionally general on the notion of 'decision module'; it may be a software in the case of a software-agent, or some piece of knowledge in the case of human-agent.

Second, along with [Bottazzi and Ferrario, 2009, Porello et al., 2014], we distinguish between *social role* and *organisation*. Social roles are useful to model properties that are only contingently satisfied by objects at a certain time. In this sense, a social role is a DOLCE-Core concept. When an individual satisfies (*is classified by*) a role, it necessarily satisfies a non-role property, too, which guarantees its identity over time.[15] An example is the role r of *being president* (of a company or government). Take the individual object John. At time t and only at t John satisfies r, e.g., he is the president of Apple Inc. When John stops being Apple's president, he continues to exist as a person. We allow for one and the same individual to play more roles at the same time and, similarly, for one role to be simultaneously played by several individuals (e.g., *being employee*).

Organisations are non-physical objects that are created and sustained by groups of people, and are regulated by some norms, which in some cases comply with the laws of governments. Amongst the relationships presented in [Bottazzi and Ferrario, 2009], the most relevant for our purposes is that of *affiliation* linking an agent to an organisation via the role(s) it plays within the organisation. "For instance, an individual [agent] who plays the role of researcher is affiliated to a University [i.e., an organisation]" [Bottazzi and Ferrario, 2009, p.7], where the role is said to be *institutionalised*, i.e., regulated and embedded within the structure of the University. According to [Bottazzi and Ferrario, 2009], a fundamental distinction that characterises organisations in opposition to general aggregations, groups, of people regards indeed memberships conditions. More precisely, "[t]he agent who decides to become member of [...] an organi[s]ation agrees to undertake all the rights and duties connected to the role that (s)he will play within the organi[s]ation" (ibid.). Even more, the members of an organisation acknowledge "a decision procedure, that is to say when there exists a mechanism to unify the heterogeneous attitudes of the individuals into a single attitude" [Porello et al., 2014, p.11].

4.3 Business objects

The representation of what we call *business objects* is common in business process; as we saw in Sect. 2, e.g., BPMN includes *data objects* to refer to, generally speaking, data that is manipulated during processes executions. Our idea is to establish a clear cut-off distinction between some piece of data encoded in a certain language and displayed on specific, e.g., computer files or paper sheets, and the "content" of the data. For example, we want to say that two different files, e.g., pdf and word files, display the same content but in two different formats.

In applied ontology, few systems [Smith and Ceusters, 2015, Bekiari et al., 2015, Mizoguchi, 2010] have attempted an ontological treatment of what is called *information object*. These ontologies agree in distinguishing between information objects and their physical carriers like paper sheets or computer files; also, the same information object may be displayed in multiple carriers while retaining its identity. For example, John's copy of his flight ticket to New York and the copy of the ticket owned by the travel agency are different carriers of the same information object. There are clearly some overlaps between the notions of information and business objects.

In computer science one usually distinguishes between data, information and knowledge (see, e.g., [Halpin and Morgan, 2010]). Roughly, the first refers to uninterpreted signs, the second to their (contextual) interpretations and the third to the overall system by which interpretations are provided. For example, a red light on a traffic signal is a sign-data, its interpretation as a stop is the information, whereas the rules of the social system behind the traffic signal is part of the knowledge owned by social agents. Differently from information objects, however, in the case of business objects, we do not want to account for 'interpretations' or the 'meaning' of data. As said, the intuition is to distinguish between, e.g., a written document and its content, where the latter is hereby assumed to be independent from any interpretation.

With these considerations at hand, we hold that business objects can change over time while keeping their identity, are necessarily displayed in physical carriers although they do not depend on specific carriers, and can therefore participate in events only indirectly, namely, by means of their carriers. In the next section we shall see how the notion of business object relates to DOLCE-Core.

5 DOLCE Core extended

Having discussed these ontological aspects as requested in KAOS, we provide an integration of DOLCE Core and DOLCE with an extension that adds new categories as deemed relevant in our analysis. In Fig. 6 the three top categories form a fragment of DOLCE Core, compare to Fig. 3. We extend this with a few categories from DOLCE, namely, `DC:Non-physical Object` with its subcategory `DC:Social Object` and `DC:Physical Object` with two subcategories `DC:Agentive Physical Object` and `DC:Non-Agentive Physical Object`.

[15] See the Appendix section for some comment on the notion of identity.

This organization is further extended with four new categories. The category of `Social Role` has been long discussed and forms a subcategory of the `Concept` category. Some social objects are now specialised in `Business Object`, which we have seen to be crucial for KAOS, and `Organization`, which collects agentive social entities. As specialization of the `Non-Agentive Physical Object` category, we add the category `Information carrier`, which collects all the physical objects in which some content is (intentionally) codified.

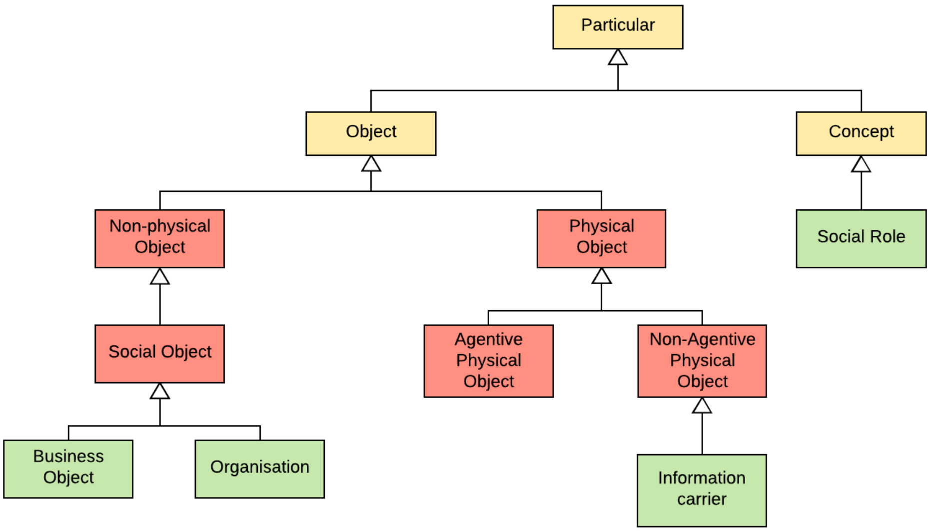

Figure 6 – DOLCE-Core extended. Yellow classes belong to DOLCE-Core; red classes to DOLCE; green classes are hereby added.

6 Discussion on BPMN

With the analysis provided in the previous sections, we can now provide an ontological interpretation of some modelling elements of BPMN; the purpose is to disambiguate their meanings and to support their coherent use.

Recalling the DOLCE-Core distinctions between events and objects, BPMN activities and events are modelling elements for the representation of `DC:event`. More specifically:

- `BPMN:Activity`. Following the BPMN Quick Guide, `BPMN:Activity`—at the token level—refers to DOLCE-Core events that occur over **extended** temporal regions independently from their atomicity. A task in BPMN can be therefore atomic (either from the modelling or ontological view) but temporally extended (as we saw in Fig. 5).

- `BPMN:Event`. BPMN events—at the token level—refer to DOLCE Core events that occur over **instantaneous** temporal regions. From the structural viewpoint, they could be (at least in principle) either atomic or compound, even though, as we saw, the general interpretation is to see them as atomic (in the ontological sense). Hence, we stick to the understanding of BPMN events as ontological atomic and instantaneous DOLCE-Core events.

Additionally, as we saw, BPMN activities and events can be either **sending** (throwing) or **receiving** (catching). In our terms, sending activities/events correspond to **endogenous** DOLCE-Core events, whereas their receiving counterparts correspond to **exogenous** DOLCE-Core events. The motivation is that only in the former case, given a process, there is an agent who is responsible for their occurrence within that process. In the latter case, an agent participates in a receiving task/event, whose happening is not however under its control.

Despite these considerations concerning the temporality, structure and endogenous/exogenous dimension of BPMN activities and events, as we saw, there is a well-documented [Kossak et al., 2014] overlap between them, especially between (i) sending activities and throwing events, (ii) receiving activities and catching events. From this perspective, BPMN suffers from what is known as *construct redundancy* [Guizzardi, 2005], namely, the presence of multiple modelling elements to represent the same entity. For example, it is plausible to see a receiving task as being ontologically atomic and instantaneous, in which case it corresponds to a catching event.

Moving now to the characterisation of the entities that take part in business processes, let us focus on `BPMN:Pool`, which is the basic graphical element to represent participants. From a high-level perspective, pools are used to depict the objects responsible for the execution of the process at stake. More specifically, a pool can stand for a:

- Agent, e.g., *Robert* in Fig. 7;[16]
- Organisation, e.g., *travel agency* in Fig. 2;
- Social role, e.g., *customer* in Fig. 2 or *purchaser* in Fig. 1.

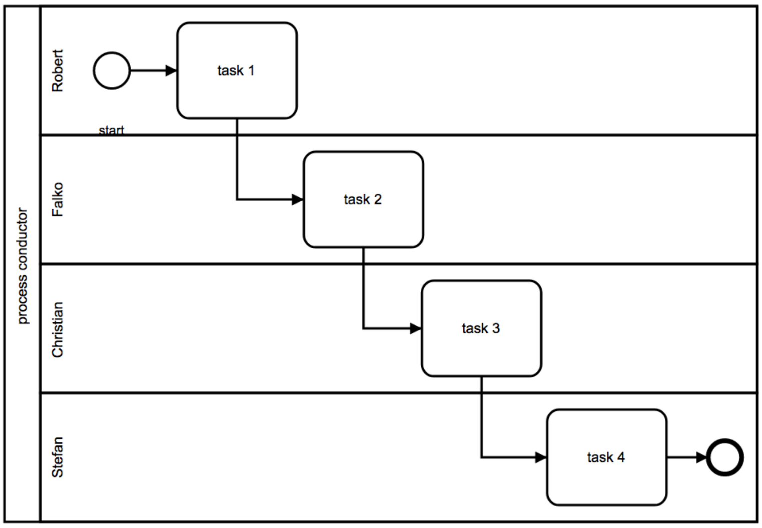

Figure 7 – Example of BPMN process model

Under this respect, `BPMN:Pool` suffers from *construct overload* [Guizzardi, 2005], since one and the same modelling element can be used to represent different entities.

Note that it makes a great difference to use a pool to represent an agent or a social role. Consider, e.g., the process model in Fig. 7, whose lanes depict individual agents, namely, Robert, Falko, Christian and Stefan. Accordingly, were one of them replaced by a different individual agent, the overall process model would change, since it is bound to very specific agents. Differently, consider *customer* in Fig. 2. Since one and the same social role can be played by different agents, the player of *customer* can change without affecting the model itself. These considerations should not discourage modellers in attaching individual agents to pools, since this choice may make perfectly sense within certain application contexts. Nevertheless, from our perspective, the use of pools for social roles makes a process model more flexible to changes since it is not restricted to specific individuals; also, it allows specifying the *social* dimension of business processes, namely, the fact that an agent is responsible for the process execution because of its role.

As regards the use of pools for depicting organisations, consider the models in Fig. 8 and Fig. 9, showing pools that are decomposed into lanes.[17] The decomposition of pools into lanes can be done in different manners, e.g., by associating each lane either to a social role (Fig. 8) or an agent (Fig. 7) or the departments within an organisation (Fig. 9). Hence, there is a relevant change in the relationships that are assumed to hold between a pool and its lanes depending on what they represent. In particular:

- **Pool-organisation** and **Lane-agent**: the agent is a member of the organisation. This may stand for the relation of *affiliation* presented in Sect. 4.2;
- **Pool-organisation** and **Lane-social role**: the social role is embedded into the organisation. This may be understood as the relation of *institutionalisation* we saw in Sect. 4.2;
- **Pool-organisation** and **Lane-department**: the department is part of the organisation so that each member of the former is member of the latter. This may be understood as a mereological notion of parthood tuned on social entities.

[16] The figure is taken from `https://camunda.org/bpmn/reference/`.
[17] Recall that lanes are graphical elements used to specify multiple participants within the same pool.

These distinctions are not explicitly addressed in BPMN, and the decomposition of pools into lanes is left open to users. Also, despite the presence of pools (and lanes) for the representation of processes' participants responsible for their execution, it is worth stressing that BPMN does not allow for the explicit modelling of many other entities that take part in processes. Looking at Fig. 8, for instance, there is no mean to explicitly represent the pizza or the money; these are just represented as strings.[18]

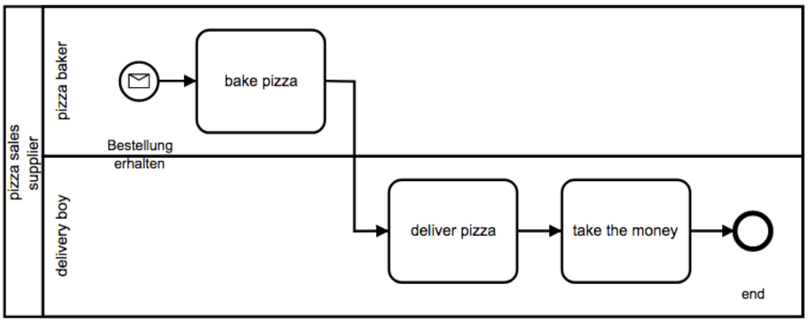

Figure 8 – Example of pool-organisation decomposed into lane-roles

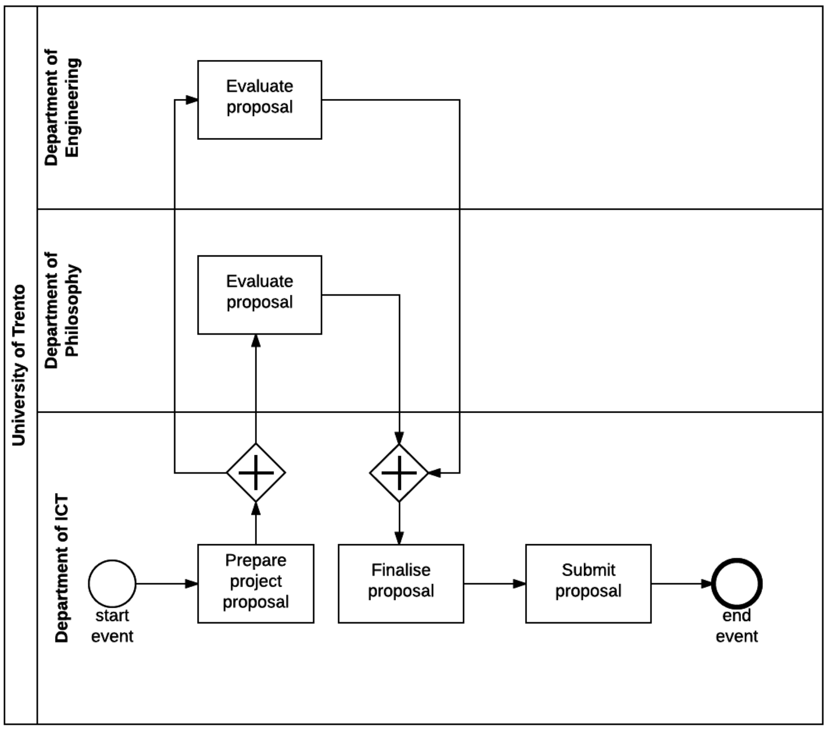

Figure 9 – Example of pool-organisation decomposed into lane-deparments

Finally, we saw in Sect. 2 that BPMN includes data objects to model information that is produced and exchanged during business processes. Recalling the analysis presented in Sect. 4.3, in order to represent information, it is necessary to distinguish between information objects and their physical carriers. Under this perspective, `BPMN:data object` suffers from construct overload, since it is meant to convey both meanings. Note that for business process modelling it is relevant to explicitly address this distinction. Looking at Fig. 9, e.g., one may want to explicitly say that the *Department of ICT* sends an information object to both the *Department of Philosophy* and the *Department of Engineering*, which means that the *same* information object is displayed in *different* supports, e.g., two different pdf files.

[18]Compare BPMN with notations like Event-Driven Chain (EPC) where participants can be explicitly encoded by means of graphical elements [Adamo et al., 2017].

7 Conclusion

We presented throughout the deliverable how DOLCE-Core can be used to analyse the BPMN modelling notation, where the focus on the latter is due to the KAOS project's requirements and goals. In particular, we took into account both the core BPMN flows elements, namely, activities and events, and BPMN Pools and Lanes; the former are fundamental for the representation of the main building blocks of processes, while the latter are relevant to talk about the entities responsible for their execution.

As a result of the analysis, on the one hand we provided a conceptual framework to ground some of the modelling choices behind BPMN, e.g., how to justify the structural atomicity and temporal extension of activities, or the use of pools for agents, organisations and social roles. On the other hand, however, we also put forward some of the deficiencies of BPMN, e.g., the fact that multiple modelling elements (tasks and events) can be used to represent the same domain entities.

Hopefully, the analysis can support modellers in the development of clear and coherent process models. Also, our results may be taken as starting point for the extension and improvement of BPMN itself, for instance, adding explicit means to distinguish between lanes for organisations, agents or roles. Also, we saw that BPMN cannot explicitly handle the representation of all participants in a process, e.g., the resources used for manufacturing. This consideration may suggest a direction to extend the notation. Further analysis is however required; e.g., by taking into account agents and organisations, one may explore their *commitments* towards the realisation of a certain process. Especially for *business* process modelling, it is indeed relevant to represent agents' (or organisations) expectations, and how a process fulfils them.

Appendix: Identity

The notion of identity is one of the most challenging in ontology. It is not a case that according to some philosophical theories, identity is one of the most primitive and non-definable relations (see [Hirsch, 1992]).

In applied ontology, identity became a pillar for analysis and ontology design with the work of Guarino and colleagues (see [Guarino and Welty, 2000, Guarino et al., 1994]). Guarino's contribution builds on the philosophical distinction between *sortal* and *non-sortal properties*. In the words of Strawson [Strawson, 1959], a sortal property provides a manner to count and classify the entities to which it applies. An example is *being a cat*, let us call C its corresponding predicate. If x and y refer to instances of (the property described by) C, then they satisfy some properties that contribute to the definition (or characterisation) of C and by which x and y can be distinguished from each other (if they refer to different individuals). The individuation of these properties is not easy and, as suggested by Lowe [Lowe, 2013], it may not be a matter to be established *a priori* by philosophical investigation. For instance, as in the case of mathematics it is up to mathematicians to establish when two sets are the same, it is up to physicists to establish when two material bodies, understood as collections of particles bounded by physicals laws, are identical. A *criterion of identity* makes explicit the properties for the identity of the instances of a certain sortal (e.g., *having the same members* for sets).

Differently from sortals, non-sortal properties provide only a manner to distinguish and therefore group particulars which are already categorised according to some sortal. For example, in the extension of the property *being red*, we find t-shirts, tables, chairs and books, among others. These can be grouped in as much as they are red; however, it is clear that they are instances of different sortals (e.g., *being a t-shirt*, *being a table*, etc.).

What is the place of identity in applied ontology? Guarino and colleagues' idea is that by establishing a criterion of identity for the domain entities to be represented, we can make explicit their *nature*, e.g., we can establish what distinguishes books from chairs, persons from cows. Following this reasoning, they argue that every instance in a knowledge-base has to instantiate at least one sortal property.[19] Take, for instance, *being customer*; reasonably, it is a non-sortal property that can apply to entities differently classified, e.g., persons and organisations.[20] Following the taxonomical constraints in [Guarino and Welty, 2000], each instance of *being customer* has to instantiate some sortal. It is this property that establishes *what type of entities* we have when we talk of customers in our domain and that preserves their existence when they stop being customers.

Looking back at the business process modelling literature, asking for criteria of identity for business processes means to look for the properties that (i) characterise their nature, e.g., are helpful to distinguish *buying* from *selling* processes, (ii) are fundamental to understand when two process instances, x and y, are actually the same.

[19] This principle has been inherited in the OntoUML approach as a strict constraint [Guizzardi, 2005].

[20] More precisely, one should distinguish between non-sortals that apply to entities of the same type (e.g., *being a student*) and non-sortals that apply to entities of different types. The latter are known as *mixin* in the conceptual modelling literature [Guizzardi, 2005].

References

[Adamo et al., 2017] Adamo, G., Borgo, S., Di Francescomarino, C., Ghidini, C., Guarino, N., and Sanfilippo, E. (2017). Business processes and their participants: an ontology-based perspective. In *Proceedings of the 16th Conference of the Italian Association for Artificial Intelligence (AIXIA)*.

[Baader et al., 2003] Baader, F., Calvanese, D., McGuinnes, D., Nardi, D., and Patel-Schneider, P. (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge university press.

[Bekiari et al., 2015] Bekiari, C., Doerr, M., Le Bœuf, P., and Riva, P. (2015). FRBR object-oriented definition and mapping from FRBRER, FRAD and FRSAD (version 2.4). *International Working Group on FRBR and CIDOC CRM Harmonisation*.

[Bertolazzi et al., 2001] Bertolazzi, P., Krusich, C., and Missikoff, M. (2001). An approach to the definition of a core enterprise ontology: CEO. In *OES-SEO 2001, International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations*, pages 14–15.

[Borgo and Masolo, 2013] Borgo, S. and Masolo, C. (2013). Foundational choices in DOLCE. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*. Springer Science & Business Media.

[Bottazzi and Ferrario, 2009] Bottazzi, E. and Ferrario, R. (2009). Preliminaries to a DOLCE ontology of organisations. *International Journal of Business Process Integration and Management*, 4(4):225–238.

[Guarino et al., 1994] Guarino, N., Carrara, M., and Giaretta, P. (1994). Formalizing ontological commitment. In *AAAI*, volume 94, pages 560–567.

[Guarino and Welty, 2000] Guarino, N. and Welty, C. (2000). Towards a methodology for ontology-based model engineering. In *Proceedings of the ECOOP 2000. Workshop on Model Engineering, Cannes, France*.

[Guizzardi, 2005] Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models*. PhD thesis, Centre for Telematics and Information Technology, University of Twente.

[Halpin and Morgan, 2010] Halpin, T. and Morgan, T. (2010). *Information modeling and relational databases*. Morgan Kaufmann.

[Hirsch, 1992] Hirsch, E. (1992). *The concept of identity*. Oxford University Press.

[Kossak et al., 2014] Kossak, F., Illibauer, C., Geist, V., Kubovy, J., Natschläger, C., Ziebermayr, T., Kopetzky, T., Freudenthaler, B., and Schewe, K.-D. (2014). *A Rigorous Semantics for BPMN 2.0 Process Diagrams*. Springer.

[Lowe, 2013] Lowe, E. J. (2013). *Forms of thought: A study in philosophical logic*. Cambridge University Press.

[Masolo et al., 2003] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). WonderWeb deliverable D18. Technical report, Laboratory for Applied Ontology ISTC-CNR.

[Masolo et al., 2004] Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., and Guarino, N. (2004). Social roles and their descriptions. In Dubois, D., Welty, C., and Williams, M., editors, *Principles of Knowledge Representation and Reasoning*, pages 267–277. AAAI Press.

[Mizoguchi, 2010] Mizoguchi, R. (2010). Yamato: yet another more advanced top-level ontology. In *Proceedings of the Sixth Australasian Ontology Workshop*, pages 1–16.

[(OMG), 2011] (OMG), O. M. G. (2011). Business process model and notation (BPMN), version 2.0. Technical report.

[Porello et al., 2014] Porello, D., Bottazzi, E., and Ferrario, R. (2014). The ontology of group agency. In *FOIS*, pages 183–196.

[Rospocher et al., 2014] Rospocher, M., Ghidini, C., and Serafini, L. (2014). An ontology for the business process modelling notation. In *FOIS*, pages 133–146.

[Russell and Norvig, 1995] Russell, S. and Norvig, P. (1995). *Intelligence artificial: A modern Approach*. Prentice-Hall, Egnlewood Cliffs.

[Sanfilippo et al., 2014] Sanfilippo, E., Borgo, S., and Masolo, C. (2014). Events and activities: Is there any ontology behind BPMN? In Garbacz, P. and Kutz, O., editors, *Formal Ontology in Information Systems. Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS)*, volume 267, pages 147–156. Springer Verlag Berlin Heidelberg.

[Scheer, 2002] Scheer, A. (2002). *ARIS - vom Geschäftsprozess zum Anwendungssystem*. Springer, Berlin [u.a.], 4., durchges. aufl. edition.

[Smith and Ceusters, 2015] Smith, B. and Ceusters, W. (2015). Aboutness: Towards foundations for the information artifact ontology. In *Proceedings of the International Conference on Biomedical Ontology (ICBO) 2015*.

[Strawson, 1959] Strawson, P. F. (1959). *Individuals*. Routledge.

[Wong and Gibbons, 2011] Wong, P. and Gibbons, J. (2011). Formalisations and applications of BPMN. *Science of Computer Programming*, 76(8):633–650.