An Ontological Core for Conformance Checking in the Engineering Life-cycle

Andreas JORDAN, Matt SELWAY, Wolfgang MAYER Georg GROSSMANN and Markus STUMPTNER
Advanced Computing Research Centre, School of Information Technology & Mathematical Sciences, University of South Australia, Australia

Abstract. Effective exchange of information about processes and industrial plants, their design, construction, operation, and maintenance requires sophisticated information modelling and exchange mechanisms that enable the transfer of semantically meaningful information between a vast pool of heterogeneous information systems. In order to represent entities relevant to the engineering life-cycle, social concepts, descriptions, roles, artefacts, functions, and information objects must be integrated in a coherent whole. Forming the basis of this integration in our framework is the DOLCE foundational ontology. In this paper we propose an ontologically well-founded approach to modelling artefacts, their requirement specifications and functional roles, such that consistency of their relationships in the data model can be verified. Specifically, we discuss the modelling of engineering artefacts, roles and role-filling capacity in the context of data modelling for industrial information exchange.

Keywords. Foundational Ontology, Artefacts, Roles, Information Objects

1. Introduction

Effective exchange of information about processes and industrial plants, their design, construction, operation, and maintenance requires sophisticated information modelling and exchange mechanisms that enable the transfer of semantically meaningful information between a vast pool of heterogeneous information systems. This need increases with the growing tendency for direct interaction of information systems from the sensor level to corporate boardroom level. One way to address this challenge is to provide a more powerful means of information handling, including the definition of ontology-based industry standards and their use in semantic information management.

A sample case for such ambitions is the Oil & Gas industry. The ISO15926 [1] and MIMOSA [2] standards are long-running projects aimed at defining shared concepts and relationships that can be used to express and communicate the information held in proprietary information systems. Both standards are interchange

The final publication is available from IOS Press.
A. Jordan, M. Selway, W. Mayer, G. Grossmann, and M. Stumptner. An Ontological Core for Conformance Checking in the Engineering Life-cycle. In *Formal Ontology in Information Systems*, Frontiers in Artificial Intelligence and Applications, volume 267, pages 358–371, 2014. DOI: 10.3233/978-1-61499-438-1-358

standards that are being developed to facilitate the integration of data in support of the life-cycle activities and processes of process plants. The main current use of ISO15926 is for design data documentation in EPC (Engineering, Procurement and Construction) companies, and O/O (Owner/Operator) companies for MIMOSA.

In this paper we propose an ontologically well-founded approach to modelling artefacts, their requirement specifications and functional roles, such that consistency of their relationships in the data model can be verified. We build upon the DOLCE foundational ontology [3] and refine ideas introduced in [4] and [5]. Specifically, we discuss the modelling of technical artefacts, roles and role-filling capacity in the context of data modelling for industrial information exchange.

We show that explicit models of roles and role placeholders can yield an ontologically principled approach to relating semantic aspects attached to concrete and abstract artefacts throughout their life-cycle. We discuss how the comparison of nominal values in the specification of a role to the actual values of the artefacts filling those roles allow us to implement semantic validation far beyond the capabilities of current type-based information models. Moreover, our model avoids the difficulty of modelling time-varying qualities of artefacts that may be found in static information models [6].

For example in Figure 1, the entity *CO_P-101-rev.1* in "Detailed Engineering" can be represented as an artefactual role, and the function place in "Plant Design" as a conventional system component acting as a placeholder for an artefactual object that will play the role at some time. Moreover, we make explicit the way in which the constraints stated in the specification attached to the role shall be met by the nominated or actual capabilities of the corresponding artefactual object. This allows us not only to verify conformance of the engineering specification and its design, but also to establish if the operational behaviour of the plant conforms to the specified requirements associated with its role.

This focus means that we take a somewhat different tack from the work of Borgo [7] who also examines how to deal with the definitions of ISO15926-2 (Part 2 of the standard) in an ontologically sound fashion. However, where Borgo envisions his work as a general methodology for the embellishment of ISO15926-2, that model has been standardised and in use now for more than a decade, and so there is some experience on which parts of the model are actually used for practical modelling. From our perspective of accessing and transforming ISO15926 data, we do not actually need to capture all wrinkles of abstract concepts that may be more effectively handled by linking to a sound upper ontology. Instead our focus is to mirror the content as elegantly and minimally as possible, with the ability to actually check relevant properties based on restrictions that should be expressible in the ontology. We also focus our effort specifically on the parts of the model that have been found to be required in actual use of the standard. All these factors make it possible to strive for a more natural solution that mirrors the information captured in ISO15926 in an ontologically sound form rather than attempting to capture it in the core of a new and even more complex structure that retains the idiosyncrasies of ISO15926.

2. ISO15926 in Practice

We are currently engaged in an engineering pilot (known as the "Oil and Gas Interoperability Pilot" or simply "OGI Pilot" among the participants) that aims at the automated, model-driven exchange of data between the ISO15926 and

MIMOSA/CCOM ecosystems. Specifically, one of the tasks is the provisioning of operational-side systems (based on MIMOSA) from the information contained in design-side systems (based on IS15926). This is referred to as digital handover from EPC to O&M (Operations & Maintenance) companies [8]. One use case of the OGI Pilot is to demonstrate digital handover from the design files produced by the CAD suites of participant companies (Bentley, AVEVA, Intergraph - using ISO15926) to the IBM Integrated Information Core (IIC) intended as the central O&M system, which is based on the MIMOSA/CCOM standard.

The demonstration design being produced in the pilot is that of a bitumen refinery (refining bitumen from oil sands into diesel fuel) and is patterned after an ongoing real world project. While the models are being extended to capture a whole refinery by 2016, the examples currently used in the pilot are based on a debutaniser tower, a specific major part of the plant that takes part in the fractionation process (unit operation) of the refinery (specifically removing butane from the hydrocarbon mix).

Although the ISO15926 standard provides a flexible framework for defining globally unique names for shared concepts and relationships between them, the framework remains insufficient for many information exchange use cases that are relevant in practice. To achieve our goal of model-driven information transformation between the different standards' "ecosystems", a conceptual reference model has to be developed that will capture the domain information and relationships that are actually encoded in practical use of the respective standards, while being sufficiently sound to permit effective modelling and testing of domain relationships and transformations. In the spirit of [9], we consider such a conceptual model to be "an actual implementation of an ontology that has to satisfy the requirements of a running application", while at the same time attempting to minimize the "impedance mismatch" that inevitably results from divergence between the two levels.

While the conceptualisation developed for ISO15926 has been suggested as a universal upper ontology, it has been shown that it suffers from significant shortcomings such as terminological confusions that make it difficult to understand and apply (see [10]). In addition, it includes poor definitions that hamper consistent adoption and shared understanding and, from an applied ontology perspective, the data model does not qualify for the status of ontology for a number of reasons such as its lack of formalisation of concepts that ensure a strong philosophical underpinning. The generic modelling primitives of ISO15926 allow the data modeller to classify entities, concepts, and relationships according to various categorisations, and define type restrictions based on these categorisations. Although this approach facilitates type checking of assertions, it lacks sufficient semantic information to enable systems to validate essential semantic constraints about the models.

An example model is displayed in Figure 1, which shows the ISO15926 representation of the specification of a single activity in the process design for the bitumen refinery as well as some related elements from the detailed engineering, plant design, and operations stages of the life-cycle. The upper section of the model states that the activity (identified by 'Process Design Tag CO_P24') requires a 'Performer' participant that can perform the 'Pump Function (related to CO_P24)'. Furthermore, the activity requires a second participant, i.e. the 'Pumped' thing, to perform the 'Stream Function (related to CO_P24)'. The "Detailed Engineering" box illustrates the specification of the abstract placeholder ('CO_P-101') that will perform the pump function, while the "Plant Design" box shows the concrete placeholder for a specific plant. Finally, the "Operations" box represents the actual

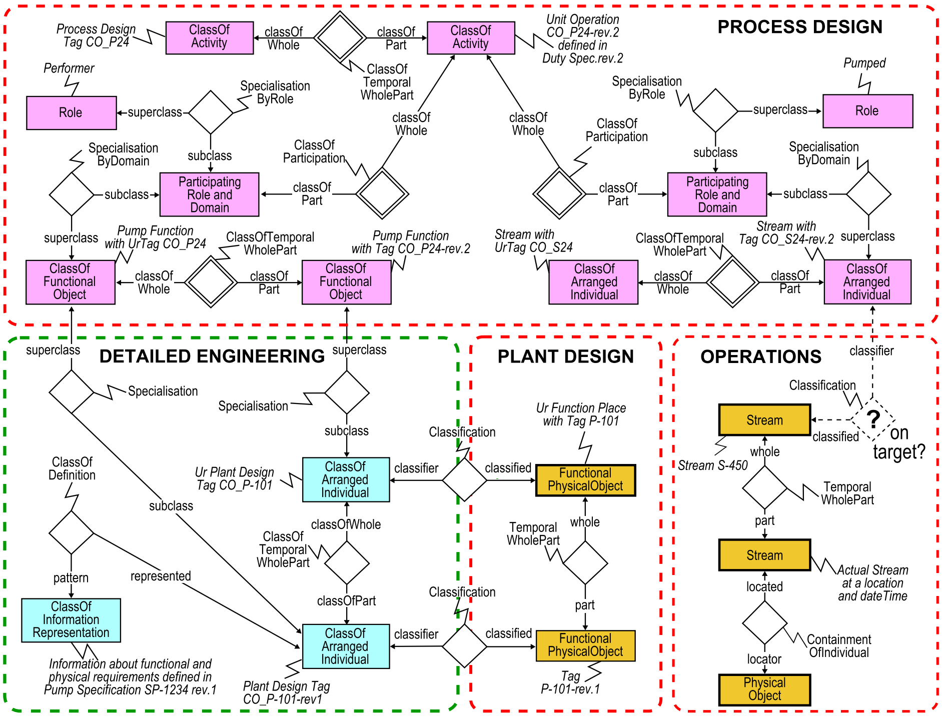

Figure 1. Process Modelling in ISO 15926

stream that is (supposed to be) fulfilling the 'Pumped' role.

This model demonstrates a number of the semantic validation issues of ISO15926. For example, the *Classification* relationships between the "Detailed Engineering" and "Plant Design" boxes indicate that the requirements specified by the abstract placeholder (*CO_P-101-rev1*) should (or may) be realised by the classified entity, i.e. the function place (*P-101-rev.1*) of the specific plant. For example, the specification may require that a certain flow rate, direction of flow and/or pressure is maintained. However, there is insufficient information to automatically verify whether or not these constraints are actually being adhered to. This is due to the lack of semantics inherent in the *(ClassOf)Definition* of the abstract placeholder (which is only a representation, not a formal definition, of the requirements), the *Classification* relationship of ISO15926, and the *Temporal whole-part* relationship that would link the actual pump to the concrete function place (not shown in the diagram).

Another issue illustrated in Figure 1 is the mixing of classification levels in which the *Participating Role and Domain* specialises a class from a higher level of classification (distinguished by diamonds with a double border being relationships from a higher level). This is contrary to the strict meta-modelling of UML[1] as well as significant research efforts into the area of multilevel modelling (e.g. [11,12]) that allow specialisation only within a single level of classification. In addition, the use of *Specialisation* means there is no ability to check that the object playing

[1] http://www.uml.org/

the role (indicated through a *Classification* relationship) can actually perform the pump function as it implies that the classified object has the function.

The "Operations" box of Figure 1 illustrates the desire to determine whether or not the plant is achieving the target output of the (current stage of the) process. Similar to before, ISO15926 utilises the *Classification* relationship to indicate that the actual *Stream* is on target with the requirements specified in the process design. Again, due to the inherent lack of semantics, the existence or not of this *Classification* relationship cannot be automatically determined. Moreover, there is no provision for determining to what *degree* the operation is on target; i.e. if the relationship holds it is *exactly* on target.

Similar issues arise when considering complex subsystems constructed from individual artefacts, where properties of the whole subsystem depend on the properties of its constituent parts. For example, checking the consistency of interconnections in a piping network demands matching diameters and tolerances of nozzles and attached pipes. Moreover, pressure tolerances must be checked throughout the entire network in order to obtain the minimum and maximum tolerable pressure within the entire subsystem. Such scenarios are essential if semantic consistency between plant requirements, design, implementation and operations are to be maintained across systems [13].

Effective validation however requires that the (combined) properties of the subsystems be related to that of the whole system. Similarly, requirements imposed on design placeholders, such as functional locations in a plant design, must be expressed in a form that is suitable for automatic semantic checking. Although basic classification and mereology are captured in the (4D) spatio-temporal extent of objects, the precise relationship between the properties of wholes and those of its parts remains implicit in the ISO15926 model.

Moreover, simple classification alone is insufficient to express variation of behaviour over time. The 4D approach of ISO15926 further complicates the representation of identity and functional roles and its role-holders over time [6]. For example, if one was to interpret the *Classification* relationship in Figure 1 as an assertion that the plant design indeed meets the requirements posed in the engineering specification, then the model becomes inconsistent once the physical object/part fails or deteriorates to the point where it can no longer perform its function, unless the classification relationship is terminated (and re-established once the equipment has been repaired). Similar issues arise in subsequent life-cycle stages, where the operational behaviour of a plant and its subsystems must be monitored and possible breaches of specifications or safety conditions, for example, are to be checked.

In the current ISO15926 standard, semantic validation is thus left implicit in the modelling tools, and various modelling and operational systems hence must agree on the specific yet implicit semantic meaning of these relationships. Given the generality of the underlying data model, different interpretations can easily arise. As a result, comprehensive semantic validation of non-trivial semantic integrity constraints is considered impossible to date. We address the inability of performing automatic validation of semantic constraints through the use of artefacts, roles and conventional system components.

3. Ontological Foundations

In this section we describe the ontological foundations used to represent artefacts (including information artefacts), roles, and functions in such a way that enables the checking of conformance between an entity and its specification. We use DOLCE as the basis for our representation, combined with a number of extensions to DOLCE in terms of social concepts, roles, and descriptions proposed in [14,15]. Furthermore, we incorporate and extend theories of artefacts and functions into the DOLCE foundational ontology in order to complete our framework.

3.1. Social Concepts and Their Descriptions

Social concepts are reified concepts that are defined by (communities of) intentional or rational agents. A *concept* x is mutually dependent on the *description* y that *defines* it ($DF(x, y)$). Descriptions represent the content or meaning of a concept and are encoded in a formal or informal language. Both social concepts and their descriptions are *non-agentive social objects* in DOLCE as they are dependent on the (community of) agents that define/encode them. Furthermore, a concept x can be *used by* a description y ($US(x, y)$) that defines other concepts, which introduces a primitive dependence on x.

The primary aspect of concepts is their ability to *classify* things. However, this classification is dynamic and temporary in nature (although it could extend for the duration of an object's life). Since endurants in DOLCE exist in time, Masolo et al. [14] define a *temporaralised classification* relationship $CF(x, y, t)$ that indicates the endurant x is classified by the concept y during time t.

A special type of social concept is *role*, which are dynamic, contextually dependent, anti-rigid, and founded properties [14]. Since the notion of roles defined in [14] and that of [16] are largely overlapping, either could be used for our purposes. As we are grounding our domain ontology on DOLCE, it makes sense to adopt the role theory of [14]; however, we retain the use of the role-holder terminology to refer to qua-individuals that arise from the role playing relationship.

In the DOLCE formalisation of roles [14], the dynamic, contextually dependent, and property aspects of roles are accounted for by their status as social concepts, their definition by descriptions, and the classification relation. For a concept to be considered *anti-rigid*, there must exist a time that an entity is present but not classified by the concept if there exists a time that the entity is classified by the concept[2]. A concept x is considered *founded* (i.e. has a context) if its definition depends on another concept y such that there exists an entity classified by y for each entity classified by x and the entity classified by y is external to that of the concept x. Since the definition of anti-rigidity is based on temporalised classification, it follows that roles can only classify endurants; or rather, only endurants can play a role. When an entity plays a role, a *role holder* is created that aggregates the qualities that the entity gains from playing that role. Moreover, the role holder *inheres* (in a similar way to how qualities inhere in entities) in the entity and is dependent on the role and the entities classified by the dependent concepts of the role's definition [15]. Since the role holder is an endurant, it can participate in perdurants and be classified by other concepts.

[2] Masolo et al. [14] note that this definition, provided in first-order logic without modality, is too strong and can be improved by incorporating a possibility operator in the definition.

3.2. Artefacts

A core aspect of our ontological framework is the representation of artefacts. Due to our target application domain of engineering model transformations, all the artefacts we consider are generally related to the space of what are called *technical artefacts* in [17]. Therefore, we eschew any arguments on whether or not artefacts constitute a natural kind and ascribe artefacts an ontological status in our framework based on the work of Houkes and Vermaas [18], and Guarino [4]. In these works, two aspects of artefacts are considered: their definition based on *item descriptions* or *design specifications*, and an object's ability to be *used for* some purpose. The former relates primarily to the physical features of an artefact and forms an *artefactual species* that an object can conform to (e.g. a particular pump type in a catalogue), while the latter primarily relates to the performing (or filling) of some *artefactual role* (e.g. a particular functional place of a plant design). This is in contrast to the work of [19] who argue that objects (natural or artificial) *constitute* an artefact based on the intentional selection by some agent and attributing it certain *capacities*.

In the following, we assume that the "technical" in "technical artefact" is understood and will omit it. However, an important departure from [18,4,17,19] is that these all focus specifically on physical artefacts. An important extension for our domain is that we need to model both physical and information artefacts occurring within engineering systems, and we will comment on the differences where necessary.

Both artefactual roles and artefactual species are dependent on (communities of) intentional agents to define and describe them. In particular, artefactual roles (as the name implies) are a type of role that are typically defined in terms of the function that is to be performed, possibly with additional constraints on the behaviour of the entity that is to fill that role. For example, the P-101 functional place (represented as an artefactual role) can only be played by an entity that can fulfil the pumping function within some threshold for the desired pressure and/or flow-rate. While a natural object can fill an artefactual role (e.g. a branch playing the role of chair) we do not consider this explicitly in our ontological framework as it is not required for our application domain.

Artefactual species, in contrast to artefactual roles, are described primarily in terms of their intrinsic properties relating to how a particular functionality may be realised [4]. These properties are specifically chosen based on a design rationale and encoded in a design specification. In engineering, these design specifications typically consist of blueprints and other technical documentation. While Houkes and Vermaas [18] admit only technical documentation in the description of artefactual species (or the 'Product classification system' in their terminology), Guarino [4] incorporates the notion of *artefactual kind* that is defined on the basis of the functionality that it is intended to provide. This is an important link between the artefactual species (which will typically specialise an artefactual kind) and the artefactual roles that entities classified by the species may be able to play.

Moreover, Guarino [4] introduces the notion of *conventional system components*, which allow the replacement of parts in an entity, while allowing the entity to retain its identity. The conventional system components in [4] are physical endurants in that they are dependent on a *host* artefactual object (defined as a physical object) and, hence, have a spatial location. Furthermore, when an entity is installed in the functional place of a conventional system component, it is said to *physically constitute* it. Finally, conventional system components hold the *nominal* *values* from the specification, allowing the properties of the entity that physically constitutes the conventional system component to be compared to the desired properties.

3.3. Function-Roles

A key aspect when dealing with artefacts is the concept of function. There are a number of ontological approaches to functions. Borgo et al. in [20] formalise the concepts of function based on two main archetypal approaches to modelling functions, the *Functional Representation* approach proposed in [21] and the *Functional Basis* approach by Stone and Wood in [22]. They ground their formalisations in DOLCE in order to facilitate ontological analyses. While their main goal is enabling the development of tools for automated reasoning between different models of functions, they state their work is still preliminary.

The approach we adopt is that of [5] and [23], in which *functions* are *roles* played by *behaviour*, which is a process. The basis for this is that the behaviour is independent of context and the same behaviour can perform different functions, while the function is dependent on a context. This *functional context* can be a *design context* or a *use context* (both specialisations of *systemic context*) given rise to *design functions* and *use functions*. We focus here on design context and functions due to their importance to our application area.

A design context is based on an entity, identifies some sub-components of the entity, behaviours of the entity and/or of the identified sub-components, and a function (role) or *goal* intended by a designer (the design function). For example, a particular heat exchanger can perform the design function of 'heating' given a design context focusing on a heat exchanger, its behaviour of heat transfer, and the intended function of 'heating' the target fluid, as opposed to 'cooling' it.

3.4. Information Objects

Another important aspect our ontological framework must deal with is *information objects*, since the design specifications of artefacts are all represented by information objects in our application domain. The DOLCE Lite Plus[3] suite of modules for DOLCE includes a basic theory of information objects. In this theory, information objects are non-agentive social objects that are *realised* by some entity. They are related to social concepts in that, e.g., information object x *expresses* description y ($EX(x,y)$). Furthermore, they can be *interpreted by* agents. In addition, the information objects of DOLCE Light Plus are linked to a communication theory.

4. Bringing it all together

In order to represent entities relevant to the engineering life-cycle, social concepts, descriptions, roles, artefacts, functions, and information objects must be integrated in a coherent whole. Forming the basis of this integration is the DOLCE foundational ontology. Therefore, to ease the integration process, we make use of a number of theories that have already been formulated in DOLCE; such as those for social concepts and descriptions [14], and information objects. However, the notion of artefacts from [18,4] and that of functions from [5,23] must still be integrated in

[3] `http://www.loa.istc.cnr.it/old/DOLCE.html`

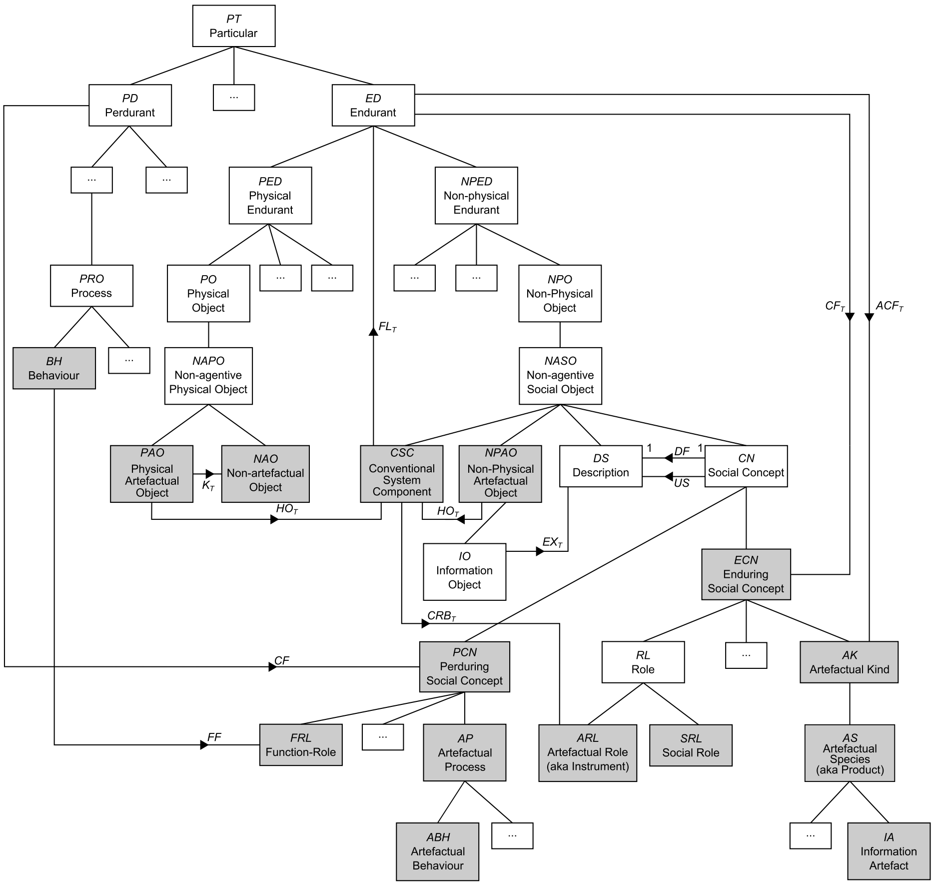

Figure 2. Overview of the types incorporated into the DOLCE foundational ontology. The shaded rectangles represent our additions.

a consistent fashion. Moreover, the 4D approach of ISO15926 must be reconciled with the 3D approach of our framework. Fortunately in practice ISO15926 does not make use of a pure 4D approach, but rather makes use of additional objects (*whole life individuals* and *Ur classes*) that identify the temporal extent of an entity. Therefore, property values of an endurant e at time t in our framework are translated into property values of a *temporal part* of the whole life individual (or Ur class) e at time t in the ISO15926 representation, and vice versa. The remainder of this section outlines our integrated ontological framework, which is summarised in Figure 2.

4.1. Artefacts

As artefactual kinds, species and roles are dependent on agents that define them, they are (non-agentive) social objects. Moreover, they are *social concepts*, as defined by [14], as they are mutually dependent on the *description* that defines them. This is straightforward for artefactual roles, since they are a subtype of role, which is already defined as a social concept [14]. By considering artefactual kinds and species to be reified concepts as well, it allows us to make explicit their definitions, handle changing specifications, and manage their relationship to the information objects that express them.

An artefactual object can then be *classified by* an artefactual species, however, we distinguish *artefactual classification* from the generic classification relation introduced for social concepts. This is due to the need for an object to be classified as an artefact species, even if it does not '... satisfy all of the constraints stated in the description', which is the definition of the *classified by* relation given by Masolo et al. in [14]. This relates to conventional system components, their nominal values, and the need to measure the conformance of an entity to what it is 'supposed to be', rather than the classification holding only when, for example, a piece of equipment is functioning perfectly as is the case in ISO15926. Therefore, we define the artefactual classification relation to allow the nominal values defined in the specification of an artefactual species to be accessible to the entity classified by the species and the conventional system components for which it is the *host object*. This allows semantic checking, as the objects that *fill* the conventional system components can be compared for *conformance* to the nominal values. If the classification were to disappear when the entity no longer met the specification, it would be impossible to compare the artefactual object to the constraints specified by the description and, hence, determine the level of conformance.

We also distinguish another type of classification, a *role playing* classification, that is between an entity and a role. Similarly to artefactual classification, this more specific relation allows properties to flow to the entity filling the role (through the role holder that inheres in the entity). It is necessary to specify these different types of relation to allow for the possibility of different types of properties to flow from the definitions of the different types of concepts to their classified entities.

As indicated by Guarino [4], we introduce the concepts of *physical artefactual object* and *non-artefactual object* as subtypes of *non-agentive physical object*. However, unlike Guarino, we admit the constitution of artefactual objects by non-artefactual objects. Although this is not our main concern, because in our domain non-artefactual objects are typically not substituted for artefactual objects, it allows us to clearly differentiate between the three cases of: (1) a natural object, such as a pebble, playing an artefactual role but not constituting an artefactual object as it does not conform to the specification of any artefactual species; (2) an artefactual object conforming to the specification of an artefactual species (and possibly playing an artefactual role); and (3) a natural object, such as rock, happening to conform to a design specification of an artefactual species and, therefore, constituting an artefactual object rather than 'becoming' an artefactual object itself.

4.2. Information Artefacts

A special type of artefactual kind is *information artefact*, which is to *information objects* as artefactual species is to (physical) artefactual objects. That is, information artefacts are social concepts that define a "type" of information object, e.g. an engineering blueprint or data sheet, that particular information objects can be classified by. Since an information artefact is an artefactual kind, the artefactual classification relation also holds for information artefacts. This allows for the semantic checking of an information object, similar to the semantic checking of a piece of a equipment.

An important relationship between information objects and artefacts (or social concepts in general) is that they are used to *express* ($EX(x, y)$) the descriptions that define concepts. Typically the description is considered to be the actual meaning or content, while the information object only expresses it. However, in our application domain, we are primarily dealing with information objects themselves. Therefore, we see information objects as a means of creating and manipulating the descriptions that they express. How this is performed can be defined in the description of the information artefact that an information object is classified by.

Since we admit conventional system components as parts of information objects, which are non-physical endurants, we must amend Guarino's [4] definition of conventional system components, which restricts them to be *physical* endurants. Instead, we consider them as a subtype of *non-physical artefactual object* (mirroring the "standard" artefactual object), which in turn is a subtype of non-agentive social object. Furthermore, we allow conventional system components to be filled by either physical or non-physical endurants as determined by their host object, which can be either physical artefactual objects or non-physical artefactual objects.

4.3. Function-Roles

The final aspect that we integrate into our framework is the notion of function as described by Mizoguchi et al. [5,23]. As such, *behaviour* is considered a process (a type of perdurant) and functions are roles played by behaviour, which we term *function-roles*[4]. In order to complete the picture we need to relate artefactual kinds and species to behaviour, and artefactual roles to function-roles. However, keeping in mind that artefactual kinds/species are reified concepts in the domain of discourse, we cannot directly associate them to behaviours. Since the concept itself does not have the behaviour, rather it includes a specification of the behaviour that an entity classified by it should have. Therefore, we separate social concepts into *enduring social concepts* and *perduring social concepts*, which classify endurants and perdurants respectively (amending the original definition of classification). We then introduce two additional concepts as subtypes of perduring social concept: *artefactual process* and a subtype *artefactual behaviour*, which are specifications of processes in general and behaviour in particular, respectively. Using these concepts, the formal descriptions of artefactual species and roles can be associated with specifications of behaviour with the *used-by* relationship (or some more specialised form thereof) through which semantic checking can be performed.

With this distinction, artefactual roles can be considered analogous to a design context in that it defines the desired/required function(s), identifies the component (i.e. its founding context), and selects the necessary behaviour specification(s) required to fill it (which may be different from the behaviour specification of any specific artefactual species). A system (i.e. an artefactual object) can then only fill that artefactual role if its behaviour specification, from its artefactual classification, is compatible with the behaviour specification of the role (among other constraints). Due to the explicit description of the role, we can perform a semantic check between the artefactual role and the entity filling it. Moreover, whether or not the system fulfils the desired function is determined by the actual behaviour of the system, i.e. it is dependent on an execution of the behaviour, which can also be semantically checked against the specifications of their descriptions.

[4]We prefer the term 'function-role' to 'functional role', used by Mizoguchi et al., since the artefactual roles, which are defined in terms of functions that need to be fulfilled can be referred to as 'functional roles'.

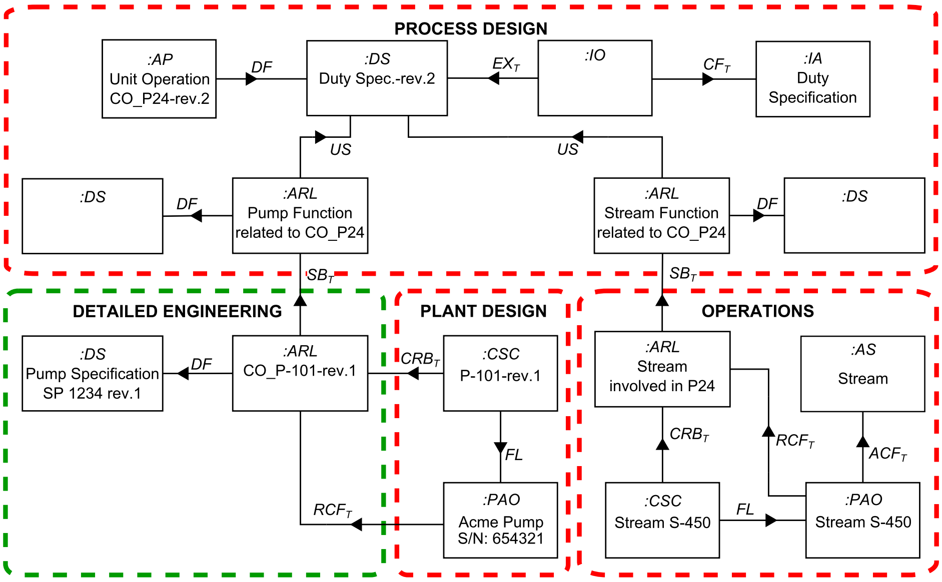

Figure 3. Process Modelling in Dolce with Artefactual Extensions

Finally, there is a link between conventional system components, artefactual roles, and function-roles, in that a conventional system component forming part of an artefactual object is created by a particular artefactual role used by the description of the artefactual kind that classifies the artefactual object. This forms a link between the object filling a conventional system component and the role, regardless of whether or not the object is currently successfully playing the role.

5. Application to the Engineering Life-cycle

Having described our ontological framework, we turn our attention to how it can be used to improve the representation of the different stages of the engineering life-cycle, including: (conceptual) process design, detailed design, plant design, and operations and maintenance. A formalisation of the concepts and relations introduced in the previous section are listed in Table 1; however, for brevity, we focus on the axioms relating to the endurant aspect of our framework.

Figure 3 demonstrates how the ISO15926 model shown in Figure 1 would be represented using our framework. In contrast to the ISO15926 representation, our framework models the activity in the process design as an *artefactual process* with a formal description (created by the duty specification information object) that defines it. This description requires there to be a participant in the process that plays an *artefactual role*, whose definition in turn requires the pump function, along with any other constraints or requirements of the role.

The specification of a role is represented by an *information object* that *expresses* a formal *description* (thus causing its creation), which *defines* an *artefactual role* for the function place. The formal description includes constraints on the possible players of the role, the *context* (not shown in the diagram for brevity) in which the role is played (e.g. location in the plant, etc.), and the formal expressions of the qualities that must be achieved for the requirements to be met (e.g. the flow

Table 1. Formalisation. See Figure 2 for the meaning of the acronyms.

Predicate	Meaning
$ACF(x, y, t)$	*x is artefactually classified by y during t*
$CF(x, y)$, $CF(x, y, t)$	*x is classified by y (during t)*
$CRB(x, y, t)$	*x is created by y during t*
$FF(x, y)$	*x fulfils the function of y*
$FL(x, y, t)$	*x is filled by y during t*
$HO(x, y, t)$	*x is the host object of y during t*
$RCF(x, y, t)$	*x plays the role of y during t*
$VPRE(x, t)$	*x is* virtually *present during t*
$context(x, y)$	*x is the context of the role y*
$exp(\phi)$	*the formal expression(s) of the description* ϕ
$\phi(x)$	*x satisfies the description* ϕ
Argument Restrictions	
$CF(x, y) \rightarrow PD(x) \wedge PCN(y)$	
$CF(x, y, t) \rightarrow ED(x) \wedge ECN(y) \wedge TR(t)$	(TR = Temporal Region [3])
$ACF(x, y, t) \rightarrow ((PAO(x) \vee NPAO(x)) \wedge AS(y))$	
$ACF(x, y, t) \rightarrow (IO(x) \wedge IA(y))$	
$RCF(x, y, t) \rightarrow ED(x) \wedge RL(y) \wedge TR(t)$	
$FF(x, y) \rightarrow (BH(x) \wedge FRL(y))$	
$HO(x, y, t) \rightarrow (PAO(x) \vee NPAO(x)) \wedge CSC(y) \wedge TR(t)$	
$FL(x, y, t) \rightarrow CSC(x) \wedge ED(y) \wedge TR(t)$	
$FL(x, y, t) \wedge HO(z, x, t) \rightarrow (PED(y) \leftrightarrow PAO(z))$	
$FL(x, y, t) \wedge HO(z, x, t) \rightarrow (NPED(y) \leftrightarrow NPAO(z))$	
$CRB(x, y, t) \rightarrow CSC(x) \wedge ARL(y) \wedge TR(t)$	
$VPRE(x, t) \rightarrow CSC(x) \wedge TR(t)$	
Ground Axioms	
$VPRE(x, t) \triangleq \exists y(PRE(y, t) \wedge HO(y, x, t)) \wedge \nexists z(FL(x, z, t))$	
$FL(x, y, t) \wedge HO(z, x, t) \wedge PRE(z, t) \rightarrow PRE(x, t)$	(Actually Present)
$HO(x, y, t) \rightarrow SD(y, x)$	(a CSC is constantly specifically dependent on its host object)
$ACF(a, c, t) \wedge DF(c, d) \wedge US(r, d) \wedge context(c, r) \rightarrow \exists! x(HO(a, x, t) \wedge CRB(x, r, t))$	
$FL(x, y, t) \wedge CRB(x, r, t) \wedge DF(r, \phi) \wedge \phi(y) \rightarrow CF(y, r, t)$	
$RCF(x, r, t) \wedge ARL(r) \wedge ACF(x, s, t) \wedge DF(r, \phi_r) \wedge DF(s, \phi_s) \rightarrow (exp(\phi_s) \rightarrow exp(\phi_r))$	

rate and pressure constraints). As a result, the conformance to those requirements can be automatically validated.

Moreover, all of the necessary information resides in explicit *descriptions* of the process design.The relationship between the artefactual role for the stream function and the role holder filling that role allows the comparison of the actual stream to the explicit description of the requirements (see Figure 3). While the role-playing relationship between the behaviour and function will (automatically) come and go as the function is being fulfilled (i.e. exactly on target), the role holder of the artefactual role can be checked as to the degree that it is on target compared to the specification.

6. Conclusion and Future Work

In this work we have proposed a framework for the modelling of engineering lifecycle information as captured in applications based on several major engineering standards. The framework is based on well-founded ontological notions grounded in a formal ontology. We have shown how the use of these notions provide a richer set of modelling mechanisms that permit validation and conformance reasoning capabilities while still affording the same granularity of domain modelling capabilities defined in these standards, in particular the scope of ISO15926 as used in the Oil & Gas Interoperability Pilot. Our current goal is to incorporate the formalisation of these concepts in first-order logic in our domain modelling and transformation environment to provide seamless ontology support for design and execution of the large-scale model transformations currently being tested in the OGI Pilot.

Acknowledgement: We thank the anonymous reviewers for their extensive comments.

References

[1] *Industrial Automation Systems and Integration – Integration of Lifecycle Data for Process Plants Including Oil and Gas Production Facilities – Part 1: Overview and fundamental principles.* International Standards Organisation, Geneva, Switzerland, 2004. Ref. No. ISO15926-1:2004(E).

[2] *Open Systems Architecture for Enterprise Application Integration 3.2.3.* MIMOSA, 2012.

[3] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, and L. Schneider. WonderWeb deliverable D17: The WonderWeb library of foundational ontologies. In *TR-NRC, Institute of Cognitive Sciences and Technology*, Italy, 2003.

[4] N. Guarino. Artefactual systems, missing component and replaceability. In *Artefact Kinds: Ontology and the Human-Made World*, Synthese Library No. 365, pages 191–206. Springer, 2014.

[5] R. Mizoguchi, Y. Kitamura, and S. Borgo. Towards a unified definition of function. In *Proc. Formal Ontology in Information Systems (FOIS 2012)*, pages 103–116, 2012.

[6] G. Guizzardi and V. Zamborlini. A common foundational theory for bridging two levels in ontology-driven conceptual modeling. In *Proc. Software Language Engineering, (SLE 2012)*, pages 286–310, 2012.

[7] S. Borgo. An ontological approach for reliable data integration in the industrial domain. *Computers in Industry*, 2014.

[8] G. Mintchell. Open O&M demonstrates 'information interoperability' for oil and gas applications. *Automation World*, pages 24–25, November 2012.

[9] C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data Knowl. Eng.*, 39(1):51–74, October 2001.

[10] B. Smith. Against idiosyncrasy in ontology development. In *Proc. Formal Ontology in Information Systems (FOIS 2006)*, pages 15–26, Amsterdam, 2006. IOS Press.

[11] C. Atkinson, M. Gutheil, and B. Kennel. A Flexible Infrastructure for Multilevel Language Engineering. *IEEE Transactions on Software Engineering*, 35(6):742–755, 2009.

[12] J. de Lara and E. Guerra. Deep Meta-modelling with MetaDepth. In *Proc. of TOOLS*, LNCS 6141, pages 1–20. Springer, 2010.

[13] A. Wiesner, J. Morbach, and W. Marquardt. Information integration in chemical process engineering based on semantic technologies. *Computers & Chemical Engineering*, 35(4):692–708, 2011.

[14] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. In *Proc. Int'l. Conf. on Knowledge Representation and Reasoning (KR '04)*, pages 267–277, 2004.

[15] C. Masolo, G. Guizzardi, L. Vieu, E. Bottazzi, and R. Ferrario. Relational roles and qua-individuals. In *Proc. AAAI Fall Symposium on Roles, an Interdisciplinary Perspective*, pages 103–112, 2005.

[16] R. Mizoguchi, K. Kozaki, and Y. Kitamura. Ontological analyses of roles. In *Proc. FedCSIS 2012*, pages 489–496, 2012.

[17] S. Borgo, M. Franssen, P. Garbacz, Y. Kitamura, R. Mizoguchi, and P.E. Vermaas. Technical artifact: An integrated perspective. In *Proc. Formal Ontology Meets Industry Workshop (FOMI 2011)*. IOS Press, 2011.

[18] W. Houkes and P. E. Vermaas. On what is made: Instruments, products, and natural kinds of artefacts. In *Artefact Kinds: Ontology and the Human-Made World*, Synthese Library No. 365, pages 167–190. Springer, 2014.

[19] S. Borgo and L. Vieu. Artefacts in formal ontology. In *Philosophy of Technology and Engineering Sciences*, Handbook of the Philosophy of Science, pages 273 – 307. North-Holland, Amsterdam, 2009.

[20] S. Borgo, M. Carrara, P. Garbacz, and P. Vermaas. Formalizations of functions within the DOLCE ontology. In *Proc. TMCE*, volume 1, pages 113–126, 2010.

[21] B. Chandrasekaran and J. R. Josephson. Function in device representation. *Engineering with Computers*, 16(3-4):162–177, 2000.

[22] R. B. Stone and K. L. Wood. Development of a functional basis for design. *Engineering with Computers*, 122(4):359–370, 1999.

[23] Y. Kitamura, Y. Koji, and R. Mizoguchi. An ontological model of device function: Industrial deployment and lessons learned. *Journal of Applied Ontology*, 1:237–262, 2006.