

Improving automation standards via semantic modelling: Application to ISA88.

Abstract

Standardization is essential for automation. Extensibility, scalability, and reusability are important features for automation software that rely in the efficient modelling of the addressed systems. The work presented here is from the ongoing development of a methodology for semi-automatic ontology construction methodology from technical documents. The main aim of this work is to systematically check the consistency of technical documents and support the improvement of technical document consistency. The formalization of conceptual models and the subsequent writing of technical standards are simultaneously analyzed, and guidelines proposed for application to future technical standards. Three paradigms are discussed for the development of domain ontologies from technical documents, starting from the current state of the art, continuing with the intermediate method presented and used in this paper, and ending with the suggested paradigm for the future. The ISA88 Standard is taken as a representative case study. Linguistic techniques from the semi-automatic ontology construction methodology is applied to the ISA88 Standard and different modelling and standardization aspects that are worth sharing with the automation community is addressed. This study discusses different paradigms for developing and sharing conceptual models for the subsequent development of automation software, along with presenting the systematic consistency checking method.

1. Introduction

In order to manage production processes efficiently, industrial plants need to be automated and continuously improved [1]. From Purdue reference model [2] to enterprise-batch control standards [3–5], industrial plants have undergone standardization and implemented sophisticated automation systems. ISA Standards address batch control systems and especially ISA88 [3], which is used in batch facilities and affects the plants design and processes [6–8]. Additionally, different systems need to be integrated not only for data management but also sharing the

functionality [9] and this can be conducted different systematic integration techniques such as agent based techniques [10], hence conceptual models in standards need to be consistent.

The use of standards is one of the modern accomplishments of industrial activities [11]. In the era of research and mass production, a huge number of products are being manufactured by multiple devices performing the same operation [12]. One of the strengths of standardization is the provision of a context where devices can be interchanged for performing the same task, as all of them will understand instructions in the same way. That is, they share the same view of reality: the one that is established by the standard [13–15]. Furthermore, there are various guidelines for writing standards [16–18], aiming to define a way of explaining things that strive for precision and avoid ambiguity; yet, they do not discuss on conceptual models and modelling.

It is generally accepted that semantic modelling is a mature technology in automation. It is broadly used as a unifying tool when managing sequential function charts [19], for ensuring systems integration and operability [20], to provide connection between models with standards [21], or to integrate metamodels in control loops [22]. Moreover, it is also studied to improve the event-driven control systems since the interpretations from different team members in batch industry exist for creating specifications [23]. However, the capacity to ensure a very high level of conceptual consistency throughout the system plays an essential role in the implementation of the different concepts suggested in a standard [24]. This is also true for the concepts defined by the ISA88 Standard [6], which is the representative case addressed in this study. Towards this end, this work provides a first approach to the construction of conceptual models from technical standards, as well as an innovative reverse approach, which is the systematic analysis and improvement of the text of the standard using the semantic domain knowledge.

Domain ontology development is an active research area nowadays. There are works applied to other domains to create ontologies [25–27], but it is hard to find applications in a same domain allowing rigorous comparative studies. Furthermore, to the best knowledge of the authors, no investigation reports the use of an ontology as an aid for improving the formalization

and writing of automation standards. Next, some paradigms are presented in this article in order to contextualize, discuss and explain the formalization of conceptual models.

1.1. Paradigms

The effort of writing a standard for automation is overwhelming. It requires the team effort of the best experts in a domain to agree and to describe a model of how systems and processes work in the most clear and precise way. In order to explain the current situation and propose improvements, three paradigms are next discussed (Figure 1).

Paradigm 1 illustrates the current state of the art in technical standard development. Currently, domain experts have to make agreements for writing technical documents (standards). A conceptual model needs to be fit to their particular views of the world, and outlying views need to be identified and discarded, in the same way equations are fit to data. This conceptual model has to be flexible enough to allow all the different visions of the problem, but precise enough to prevent misinterpretations. Once such a technical document is written, approved, and available, area experts read it and strive to produce standard-complying developments. Thus, automation experts develop automation code according to their own interpretation of the standard and to a specific purpose to solve an automation problem.

Paradigm 1 poses three main problems (Figure 1). The first one is the inherent ambiguity of natural language (NL) in which the document defining the standard is founded. Even the most careful technical writing suffers a certain degree of imprecision. The second problem is the interpretation of the text, which is done by the reader based on her/his own understanding and her/his pre-conceptions, as well as on the purpose behind the reading effort. The third problem is that the reader (i.e. the area expert) should also be the code developer. Having these two proficient technical profiles is unlikely, and separating them is desirable.

In order to avoid the problems of Paradigm 1, Paradigm 2 is suggested and depicted in Figure 1. The method presented in this paper is depicted as the second paradigm, which supports conceptual consistency of already established technical standards. Hence, a semi-automatic

methodology enables developing an ontology from the text of the standard and the aid of a team of area experts understanding the standard. The aim of this procedure is to establish a sole interpretation of the world, so that different codes can be developed for different purposes, just like in Paradigm 1, using the ontology as the underlying knowledge source. Accordingly, the area experts, who interpret the standard to create the ontology, and the code creators, who only need the ontology as input for their automation code, became separate roles. It is worth mentioning at this point other efforts for creating conceptual representations of standard systems such as B2MML [28] and SysML [29].

Regarding further development, this paper suggests a further evolution step into Paradigm 3 (Figure 1). Once a team of domain experts meets and commits to compile a text for establishing a standardized view of a domain, the effort of developing an ontology in parallel is easily affordable, while the benefit would be significantly increased. The ontology establishes without the inherent ambiguity of NL one concrete view of a domain. In a single step, two standard contents could be established for a consistent interpretation of reality: one for humans to read, in NL, and another readily usable for automation without ambiguities.

Three main advantages can be envisaged for the application of the proposed methodology depicted in Paradigm 2. First, a normative document can easily lead to an ontology of the domain helping programmers to develop standard-complying software. Secondly, an ontology can help domain experts to interpret a normative document in a precise and simpler way. Finally, a semantic analysis based on the ontology may produce useful feedback for producing an improved version of the normative document written in NL. This feedback from the ontology to the written document is a novel and an original approach to enhance standardization in the field of automation.

1.2. Why ontologies?

Any text written in NL is intrinsically ambiguous and imprecise. NL is not the best tool for establishing a view of a domain that is the result of an agreement among the experts in technical

domain, but it has been the best available tool for the latest millennia. If the intention is to transmit the knowledge to other persons, NL is the best medium available [30]. If NL is considered as a way of representing events or entities of the real world in terms of human language, the mapping is not bi-univocal: a linguistic unit (a word, a phrase, a sentence) can refer to different real world units (polysemy) while a real world unit can be uttered in NL using several variants (synonymy). Table 1 summarizes the main differences between textual documents and ontologies. Given NL inherent ambiguity, there is no mechanism to validate and ensure that the mental models built by different readers from the same text match. Thus, texts seem not enough for automation systems; they are intended and satisfactory for transmitting knowledge to humans.

Ontologies, on the other hand, are formal representations of a domain [31]. They do not carry the problems of synonym, polysemy, ellipsis or implicit knowledge. Ontologies represent concepts in a unique way by means of their relations to other concepts. Although concepts may be identified by words of NL, ontologies allow differentiating only one sense among all the senses a word may have: synonymy has to be represented as a relation between concepts. Ellipsis or implicit knowledge does not occur in an ontology, all knowledge must be explicitly represented, due to the closed world assumption usually accepted.

Researchers are using ontologies for coordinating multiple standards and models [32]. In the process of writing a standard, the use of ontologies from the very beginning removes the effects of inconsistencies between models, the confusion resulting from synonymy, and eventual terminological conflicts not only between different hierarchical levels, but also between different standards. In the field of Process Systems Engineering, several works [33–36] have recently used ontologies to (i) build intelligent software systems; (ii) use them as a support tool for systematic analysis of data; (iii) use them as a knowledge management tool; and (iv) address an emerging technology to look beyond the traditional modelling and solution methods.

This paper presents results circumscribed to Paradigm 2 in Figure 1 **Error! Reference source not found.**, obtained from processing the text of ISA88 and extracting an ontology representing the knowledge in it. This work is aimed at demonstrating that ISA88 Standard consistency could be assessed from the point of view and formalisms of ontology development. Undefined concepts, extended use of synonymy, and misleading use of adjectives have been detected in the document, which have been identified as problems, not only for the implication of automatic processes, but also for human readers. Thus, this article contributes an analysis and suggestions for developing technical standards. On the other hand, the ISA88 Standard should be considered as a first case study for a research line, and the learning outcomes from this work are expected to be of practical interest to further ontology developments from other technical documents.

2. Methodology

The main objective of the presented method is to systematically check the consistency of technical documents and provide hints for consistency improvement. This novel methodology is based on the use of semantic analysis and semi-automatic ontology construction procedures, and it has been developed by extending previous partial developments [37,38]. This method is applicable to a large set of engineering problems such as the standards ISA88 and ISA95, and other technical documents and situations where a model of an application exists. In order to show the capabilities and efficiency of this method, this paper presents an application to a case of interest for the automation community (ISA88 Standard).

Figure 2 briefly depicts the idea of this methodology. When an informal model seems to exist and appears to be of practical interest, standardization is developed with the consensus of domain experts to produce a technical document attempting to define a unified version of it (Model A). The presented methodology starts executing an automatic procedure with the technical document. With this automatic procedure (Section 2.2.1) an ontological model (Model B₁) is produced. Afterwards, a semi-automatic procedure is followed to remove mistakes from the automatic procedure and to clear the ambiguity of NL (Section 2.2.2). In this step, Model B₂

is also investigated to detect inconsistencies directly derived from the technical document. As the model formalized as ontology has a clear view of concepts and relations identified from the technical document, issues can be detected straightforwardly as given in the application of the methodology in Section 3. Since the purpose of this methodology is to distinguish these issues and to suggest improvements, in the final step, text and real-world model improvements are done according to the detected inconsistencies.

The fully-automatic creation of ontologies by mining a large set of assorted documents may retrieve a lot of irrelevant and spurious data and generate a lot of noise. For that reason the methodology has the assumption that an intelligent selection of texts will reduce noise and allow fast and straight identification of concepts and relations by pattern matching methods¹. The meaningful and large amount of relations obtained from ISA88 indicate that a technical standard is a proper resource to extract ontological information, to minimize noise and capture significant information. With the presented methodology, the semantic applications are implemented both ways: not only to create a robust formalized model, but also to detect the inconsistencies and to correct the main model existing behind this model.

Certainly, automatically building general ontologies from assorted textual sources (i.e. the Internet) has been achieved by computer science and has been reported [27,39,40]. However, this is the first time that it is used to derive a conceptual model from a normative text and to correct this model. This idea, which may be regarded as a scientific concept or an engineering application, or both, can be exploited to create more robust models, as well as to improve the internal and external consistency of the normative text. Next ontological models, automatic and semi-automatic procedures are explained.

2.1. Ontological Models

After the semi-automatic procedure, Model B₂ is formed in an ontological model. Formally, an ontology is described by:

¹ To check this hypothesis, the same methodology is applied to a play by Shakespeare and the result was the outcome of zero taxonomical relations [38].

$$O = \{C, I, P, D, A\} \quad (1)$$

where the ontology model, O , consists of a set of concepts, C ; a set of individuals I , which are instances of concepts; a set of object properties P , which describe features of a concept or an individual, or relations between them; a set of data properties, D , which describe the data associated to a concept or an individual; and a set of axioms, A , defined over $\{C \cup I \cup P \cup D\}$ that represent rules and restrictions of the model [41]. The present work is based on an ongoing methodology development, and an ontology that is not finished. The development concentrates on the set of concepts, C , and a property: the is-a² relation that generates a taxonomy from the text by connecting a concept and its super-concept. is-a relation and it's extraction from the ontology is further explained in the following section.

2.2. Automatic and Semi-Automatic Procedures

The general procedure is depicted in Figure 3 where the different steps are explained along with particular examples of each step. Then, automatic and semi-automatic steps are respectively implemented.

2.2.1. Automatic Procedure

The automatic part involves the selection of phrases, the parsing and the pattern matching.

Selection of phrases: First, the main patterns are determined by pre-processing the text in order to cluster the verbal relations and to eventually determine other important patterns occurring in the text. For instance, clustering is done to identify the verb 'to be' by gathering instances of related verbal forms used in the text such as 'is', 'are', 'be', 'been'. After the pre-processing, approximately one fourth of verbal relations in ISA88 are found to belong to the relation pattern, followed by approximately a 5% of verbal forms belonging to the part-of relation pattern. The rest of the relations are still an open issue to be clustered and used to improve the ontological model.

² Relations are given without a tilde and a dash is used to separate the words such as is-a, part-of.

The is-a relation is the most important relation and it is the only relation that all the concepts have. This relation creates the taxonomical structure in an ontology and builds its skeleton (as explained in semi-automatic procedure section). All concepts in the ontology are related via this relation, which is also called taxonomic relation and builds the main structure of the ontology [42]. The part-of relation, also called meronymy [43], is the second relation. It is not a complete relation, as is-a is, since many concepts are not composites. These is-a and part-of relations construct the generic relations [44]. For specific domains, many generic relations are of interest, but additional specific relations also exist, which can be obtained using specific extraction techniques. For instance, within the medical domain a relation ‘has adverse effect’ can be defined between the concepts ‘drug’ and ‘clinical sign’. In the batch control domain a Process can consist of a sequence of (sub)-processes (meronymic relation), an Equipment is-used for a Procedure, or a Recipe is-described a Procedure (both domain-specific relations) [45].

Moreover, Hearst [42] proposes a way of detecting the taxonomic relation is-a by means of a small set of highly precise patterns including two place holders that should be mapped to two entities in the domain. This involves the lexical and grammatical analysis of the text in order not only to identify syntagmas and verbs, but also to relate them in sentences, from which they can be extracted to form single or multiword concepts. For instance the pattern ‘<X> is-a <Y>’, contains the placeholders <X> and <Y> and matches the text ‘A general recipe is a type of an equipment independent recipe’, from which the relation is-a between GeneralRecipe and EquipmentIndependentRecipe’ could be extracted. For this part of the methodology the most important generic relation, is-a, is considered and taxonomical model is created.

Parsing: The text is prepared for pattern matching. The tool Pyparsing [46] is used to parse the text. As a result, the parsed text is ready for matching the patterns decided in the selection of phrases step.

Pattern Matching: Since the is-a relation is selected to continue in order to build the taxonomy, some patterns are determined to extract these relations. The is-a relation establishing

the taxonomy is found also expressed by further textual resources, such as the colon (:) or the same ‘such as’. Pattern matching is extended to include them.

2.2.1. Semi-Automatic Procedure

The detected concepts and relations undergo the semi-automatic procedure. First, the skeleton of the ontology is build; this includes changing the format of concepts and relations to the formal way required by the ontology. Then, necessary commonsense knowledge (taken from the text) is added, and the ontology is accordingly pruned to remove the confusing relations. Finally, detecting inconsistencies by inspection, the last and crucial procedure in this work is conducted.

Building the skeleton: As it is stated before, taxonomical relation (is-a) creates the skeleton of an ontology and reflects almost all concepts in the source. In this step, concept-relation-concept pairs detected from the automatic procedure are investigated and purified from the errors of the automatic procedure, such as textual reading mistakes, conversion errors, etc. Afterwards, all the concepts and their relations are introduced to Protégé to formalize the skeleton of the ontology.

Introducing commonsense knowledge: In the detection of relations and concepts, a complex lexical and grammatical analysis is required in order to detect the syntagmas and verbs involved. Concepts having more than one noun and adjectives whose combination create a new meaning are moved to proper places in the ontology. For example, GeneralRecipe and Recipe concepts are different concepts and have the is-a relation. This information is not explicitly written in the technical document since commonsense leads to this conclusion. Moreover, more complicated concepts like ‘and’ or ‘or’ connections can be detected as ‘<X>’ where a separation is needed. In this case, concepts are separated and new relations are created.

Pruning: Some pruning is required after the previous steps. Some relations can be mutually inferred with the appropriate axioms, and subsumed relations can be removed from the ontology.

Detecting inconsistencies: Detecting inconsistencies is the last step for completing the ontology. Yet, a lot of work needs to be done in order to enrich the ontology with more kinds of relations, enhancing its representation capacity. However, this is out of the scope of this paper

and does not affect the presented conclusions. Topological analysis allows detecting cases for which a concept has several super-concepts; these cases have to be analyzed individually in order to decide whether the case is correct or not. The procedure for detecting inconsistencies is based on analyzing the super-concepts of the concepts. From an ontological view, one concept can belong to several super-concepts, but the most frequent case is that a concept has only one super-concept. Analysis of the methodology and examples from inconsistent cases are given in Section 3.

3. Application of the Methodology and analysis of ISA88 Batch Control Standard

This section presents the analysis of inconsistent cases performed after the development of the ontological model as described in the previous section. The study has been limited to the normative concepts (i.e. the concepts defined in the standard) and it allows suggesting ways to enhance the procedure for creating standards by means of additional guidelines of good practices. For comparative purposes, this work uses the 2006 and 2010 versions of ISA88 Part 1 (ISA88R2006 [47] and ISA882010 [3]), and the latest version of Part 2 [48], Part 3 [49] and Part 4 [50]. For the sake of clarity, abbreviations and definitions are given in the glossary section.

3.1. Ontological Model

The methodology applied to the four parts of ISA88 resulted in a set C of 465 concepts interrelated via 544 relations. 290 of these relations are type of is-a and 254 relations are object properties P, which leads to meronym relation (part-of). All the concepts from the ISA88 Standard are introduced into the ontology using Protégé. In this work, only the taxonomy of the ontology is considered and misused concepts and definitions are detected by using the methodology. Detecting inconsistencies results in 42 issues (concepts with more than one parent) that are detected from the taxonomy (ontology skeleton). 27 of them are excluded from the list of concepts not relevant to batch control. 10 concepts are reported in this paper, as they are normative concepts and lead to different discussions.

The rest of the examples are also solved with the subsumed relation approach, which removes the unnecessary relations that can be inferred through other relations (e.g. ‘QualityInformation is-a Information’, ‘QualityInformation is-a ProductionInformation’ and ‘ProductionInformation is-a Information’ at the same time: this is solved by removing the relation from QualityInformation to Information since it can be inferred through the ProductionInformation concept). Next, the ProcessParameter concept is shown as a specific example presenting no problems in Figure 4. In ProcessParameter example, all the relations between the concepts are clear and do not lead to any confusion or ambiguity.

3.2. Analysis of detected inconsistencies from the ontological point of view

Detected cases are identified and solutions are given in this section. All cases are selected from normative concepts with two or more super-concepts where an inconsistency can be detected and an outline for explanation of these cases is given as follows:

- (i) The pattern matching has located a set of phrases in the text of ISA88 Standard that define or add information to the concept classification. They are shown in Table 2. In the table, bold words represent concepts and underlined words represent connectors in the patterns except definition pattern, which is depicted with ‘:’.
- (ii) Concept names are distinguished from normal text using CamelCase in this article.
- (iii) Issues related to the selected normative concept from ISA88R2006 are presented in the following subsections, see also the corresponding fragments in Table 2.
- (iv) Discussion on the solution after identifying the problem is given and the revised ontology view is depicted in a figure.

Finally, cases are discussed considering ISA882010 if an action has taken for the issues.

3.2.1. Pruning the BatchControl Concept

Figure 5 shows how the BatchControl concept concludes with two parents in the ontology constructed from the text. These relations are accepted because this is the real idea taken from the text of the standard.

3.2.2. Pruning the RecipeElement Concept

Figure 6a shows how RecipeElement pertains to two super-concepts: Representation and StructuralEntity. The decision made here is to remove the Representation concept, which is detected from Part 4, and to keep the relation with StructuralEntity being used as a synonym of Entity concept in ISA88 (shown in Figure 6b).

3.2.3. Pruning the ControlModule Concept

Figure 7a shows how the ControlModule concept has two super-concepts: GroupingOfEquipment and EquipmentEntity. Other super-concepts of GroupingOfEquipment and EquipmentEntity are shown in the graphic in order to have enough information for making a decision. Upon inspecting this segment of the ontology, it is clear that many concepts are being used as synonyms: PieceOfEquipment, EquipmentEntity, GroupingOfEquipment give the impression of different ways of naming Equipment. Instead of just removing these concepts, they have been kept, but marked as synonyms in Figure 7b.

3.2.4. Pruning the Procedure Concept

Figure 8a shows the ontology segment for the Procedure concept and how this concept pertains to two different super-concepts: ProceduralElement and Strategy. The decision here was to remove the Strategy concept since control strategy is defined as strategy in the standard taxonomy, and ProceduralElement is not at the same level of control strategy. Figure 8b shows the results after the pruning phase of the Procedure concept. Additionally, this decision is consistent with ISA882010, where the definition has been reformulated removing strategy in favor of ProceduralElement.

3.2.5. Pruning the ControlRecipe Concept

Figure 9a shows how ControlRecipe pertains to two super-concepts: Data and Recipe. The decision here was to remove Data, which comes from Part 4, considering the recipe model is modelled in the batch control systems. Figure 9b shows the results after the pruning phase.

3.2.6. Pruning the EquipmentModule Concept

Figure 10a shows the section of the ontology where EquipmentModule is revealed to pertain to three super-concepts: FunctionalGroupOfEquipment, LowerLevelEntity and, EquipmentEntity. Decisions from the pruning procedure are that EquipmentEntity and FunctionalGroupOfEquipment are synonyms of Equipment. LowerLevelEntity is removed since it gives ambiguous information. Figure 10b shows the results after the pruning.

3.2.7. Pruning the UnitProcedure Concept

Figure 11a shows that UnitProcedure pertains to five super-concepts: ProceduralElement, RecipeProceduralElement, ElementsOfBatchProduction, EquipmentProceduralElement, and Strategy. In the pruning step, the relation between UnitProcedure and Strategy concepts are removed by sticking to the decision made for the ControlStrategy case, and the relation between UnitProcedure and ElementsOfBatchProduction is removed since all the concepts in the ISA88 Standard are considered as ElementsOfBatchProduction. ProceduralElement is a super-concept of RecipeProceduralElement and EquipmentProceduralElement concepts and UnitProcedure is in the same level of these RecipeProcedural element and EquipmentProceduralElement concepts. Finally, the relation between UnitProcedure and ProceduralElement is kept depending on the pattern from Part 4 and Procedural Control Model figure (Figure 7 in Part 1) in the standard. The revised ontology is depicted in Figure 11b. Furthermore, this is again shown to be consistent, since the RecipeProceduralElement and EquipmentProceduralElement concepts are no longer included in ISA882010.

3.2.8. Pruning the Phase Concept

Figure 12a shows that Phase pertains to three super-concepts: ProceduralControl, EquipmentProceduralElement and RecipeProceduralElement. Since the RecipeProceduralElement and EquipmentProceduralElement concepts are not part of the procedural model in the standard, these relations are removed and a relation to ProceduralElement concept is added accordingly. In addition, the

SmallestElementOfProceduralControl concept is removed since it gives part-of relation between those concepts. As a result, the final decision on the concept is depicted in Figure 12b.

3.2.9. Pruning the MasterRecipe Concept

Figure 13a shows how MasterRecipe is simultaneously related to four super-concepts: Data, Recipe, Source, and TemplateRecipe. Relations coming from Part 4 (Data and TemplateRecipe) are removed from the taxonomy because Part 4 of the standard focuses on batch production database records. In addition, the relation to the Source concept (SourceConcept) is removed because of the consistency of recipe model. Recipe concept is allowed. Final decision is depicted in Figure 13b.

3.2.10. Pruning the GeneralRecipe Concept

Figure 14a shows that GeneralRecipe pertains to six super-concepts: Container, Recipe, EnterpriseLevelRecipe, CorporateRecipe, EnterpriseWideRecipe, and EquipmentIndependentRecipe. Relations extracted from Part 3 contain the GeneralRecipe segment and this generates misperception by giving additional descriptions with synonym concepts to GeneralRecipe. Other concepts are created as different concepts such as EquipmentIndependentRecipe, EnterpriseWideRecipe, EnterpriseLevelRecipe, and CorporateRecipe. Since all they represent the same concept, the is-a relations are changed to the is-synonym relation. The relation between GeneralRecipe and Container is removed since containment leads to part-of relation. Finally, the is-a relation between GeneralRecipe and Recipe concepts is kept as shown in Figure 14b.

4. Guidelines

Cases in Section 3 show how a topological analysis of the ontological model allows detecting inconsistent lexical issues that may undermine the precision of the model to be standardized. As another result of this work, this section gives some guidelines to avoid these problems in a technical document defining a model. Hence, recommendations can be suggested to improve the

development and writing of technical standards. Issues are explained with following logic: (i) problem, (ii) guideline, and (iii) example from ISA88.

Figures: The methodology presented relies on the automatic matching of textual patterns in the document. However, figures in a document cannot be processed in this way. The development of an automated process to infer semantic models from unprecedented, non-normative figures are out of the scope of this work, as well as currently unlikely. Thus, an expert has to inspect figures manually to extract information from them. In the case of ISA88, the lack of a legend in several figures made it difficult to extract information properly. This issue is more affecting when arrows of different kind occur in the same figure and no legend explains their different meaning (e.g. Figure 11 in [4] is updated in the new version [3] of standards).

- **Guideline:** *Add textual information to figures (legends, notes) when ambiguity exists.*

Synonymy: Although synonymy is a powerful rhetorical resource of the language, having multiple names for the same concept goes against precision and clarity. Terms and particularly normative definitions should be strictly followed in standard. For example, the case in Section 3.2.6 shows how several names are used for referring to the same concept in ISA88: Equipment, GroupingOfEquipment, EquipmentEntity, and PieceOfEquipment. This diversity of concept names damages both human comprehension and computer automation.

- **Guideline:** *All references to the same concept should stick to the chosen name. Synonymy should be avoided or explicitly declared especially for the normative concepts.*

Polysemy: Polysemy is the usage of similar names for different concepts. Although it can be understood by humans, it is a challenging task for computers. For instance, the concept Phase is a procedural element in the procedural model but also RecipePhase exists in the standard, which is part of a recipe procedure.

- **Guideline:** *Use different names for different concepts.*

Adjectives: ISA88 Standard is using adjectives with great care. Qualifier adjectives are likely good indicators of an is-a relation between concepts such as ‘ProceduralControl is-a Control’.

Procedural is used in 38 different concepts thus creating new concepts for instance, ProceduralElement, ProceduralControl, ProceduralHierarchy, ProceduralElementReference. More confusing are grade adjectives and comparative adjectives such as used in LowerLevelEquipment, since they suggest the existence of possible superclasses (LevelEquipment or Equipment), and further relations.

- **Guideline:** *Avoid using comparative and unnecessary adjectives.*

Adjectives with antonym: Another adjective usage is adjectives creating antonym as used in the CommonResource concept, which is also discussed by Fisher and Emerson [51]. These adjectives should be used with great care since they create the antonym of the concept and the ambiguity arises owing to searching the antonym of the concept.

Guideline: In order to avoid the ambiguity, adjective usage should be limited to determining adjectives.

5. Discussion

Prior works have documented different ways of producing ontologies or conceptual models. They are based on either the exhaustive search of a great amount of noisy data. On the other hand, writing technical documents requires a great effort because the intrinsic ambiguity of NL, which tends to increase the length of these documents.

This work is part of the continuing development of methodology for semi-automatic ontology construction, aimed at creating domain ontologies in a systematic way using technical standards as resource. It contributes an original use of semantic modelling for improving the development of automation standards and the subsequent development of automation software, according to the standard. Consequently, the main objective of the method is to systematically check the consistency of technical documents and provide suggestions for consistency improvement.

ISA88 Batch Control Standard is used as the case study in this methodology and the issues regarding to conceptual modelling that have appeared are now discussed and shared with the automation community. Quantitative assessment of the performance of the methodology

presented is difficult due to the lack of convenient metrics for quantitative comparison. Furthermore, the most significant drawback is the lack of repetitive cases for sampling and comparing. However, the paper addressed a comparison between ISA 2006 and 2010 in regard of the issues automatically detected by the proposed approach and those issues that were detected and improved by a team of experts after a revision procedure.

As a result, improvement suggestions arisen from the analysis performed in this work are compared with the newer version of the standard when it is applicable. On the other hand, technical standards appear in many communities, engineering fields and study areas, and this methodology may have a significant impact in applications with large sets of models.

An evolution of different paradigms for standards development was explained in Section 1. ISA88 is shown to be an example of the first paradigm, where a team of domain experts meets in order to agree a model for a domain model. A mid step of this evolution could be exemplified by B2MML [28], a markup language that is based in ISA88 and ISA95. In this case, a team of area experts has examined ISA88 and has built an object diagram of concepts and relations together with a language for information interchange based in XML or specifications. A first step in this evolution towards Paradigm 3 would be those automatic pattern matching processes [37,38] that help area experts to build an ontology. This ontology would be more powerful than an object model since it allows performing inferences through axioms.

In third step proposed, an ontology would be constructed in parallel and in coordination with a NL text document. The result of this would be twofold: one document for humans to be read with a high degree of precision and clearness, and an information resource directly usable by computers. There would be no need for a two-step process as in Paradigm 2 where two teams meet, first to write a standard, and later to understand it and give an interpretation of it. In Paradigm 3, the text and its interpretation would be coordinated and simultaneous.

6. Conclusions

This work has presented a practical and successful technique for building semantic models from texts and for detecting inconsistencies in technical documents. Hence, this work has described how an ontological analysis of the text of a standard is a powerful tool that can help enhancing the precision of the text and speed up the implementation of automation software complying the standard, as well as finding a practical implementation to a theoretical study.

This paper has also presented the ontological analysis of a conceptual model coming from a technical standard. The semi-automatic analysis of the ISA88 technical documents has revealed modelling questions worth to be considered by the domain experts, such as the interaction between written standards and ontological models. Therefore, after explaining the ontological analysis in Paradigm 2, Paradigm 3 is suggested to prevent inconsistencies in automated applications. Furthermore, some useful guidelines for writing technical standards are given as a result of detected issues.

Acknowledgement

Financial support from the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund, both funding the Project ECOCIS (DPI2013-48243-C2-1-R), SKATER (TIN2012-38584-C06-01), and from the Generalitat de Catalunya (2014-SGR-1092-CEPEiMA, AGAUR) is fully appreciated. This work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

References

- [1] Vogel-Heuser B, Fay A, Schaefer I, Tichy M. Evolution of software in automated production systems: Challenges and research directions. *J Syst Softw* 2015;110:54–84. doi:10.1016/j.jss.2015.08.026.
- [2] Williams TJ. A Reference Model for Computer Integrated Manufacturing (CIM), A Description from the Viewpoint of Industrial Automation. ISA - Instrument Society of America; 1989.
- [3] ISA. Batch Control, Part 1: Models and Terminology, ANSI/ISA-88.01-2010. ISA Committee; 2010.
- [4] ISA. Enterprise-Control System Integration, Part 1: Models and Terminology, ANSI/ISA-95.00.01-2000. ISA Committee; 2000.
- [5] ISO British Standards Institution. Enterprise-control system integration, Part 1 : Models and terminology (BS EN 62264-1:2013). BSI Standards Publication; 2013.
- [6] Schaefer JJ. ISA-S88.01-1995 impact on design and implementation of batch control systems. *ISA Trans* 1996;35:351–6.
- [7] Nortcliffe A., Thompson M, Shaw K., Love J, Fleming P. A framework for modelling in S88 constructs for scheduling purposes. *ISA Trans* 2001;40:295–305. doi:10.1016/S0019-0578(00)00052-5.
- [8] Nelson PR, Shull RS. Organizing for an initial implementation of S88. *ISA Trans* 1997;36:189–95. doi:10.1016/S0019-0578(97)00013-X.
- [9] Harjunkoski I, Nyström R, Horch A. Integration of scheduling and control - Theory or practice? *Comput Chem Eng* 2009;33:1909–18. doi:10.1016/j.compchemeng.2009.06.016.

- [10] Yu L, Schüller A, Epple U. On the engineering design for systematic integration of agent-orientation in industrial automation. *ISA Trans* 2014;53:1404–9. doi:10.1016/j.isatra.2013.12.029.
- [11] Brayton R, Carloni LP, Sangiovanni-Vincentelli AL, Villa T. Design Automation of Electronic Systems: Past Accomplishments and Challenges Ahead [Scanning the Issue]. *Proc IEEE* 2015;103:1952–7. doi:10.1109/JPROC.2015.2487798.
- [12] Jung K, Morris K, Lyons KW, Leong S, Cho H. Performance Challenges Identification Method for Smart Manufacturing Systems. Gaithersburg, MD: 2016. doi:10.6028/NIST.IR.8108.
- [13] ETSI, European Telecommunications Standards Institute. Why we need standards 2015. <http://www.etsi.org/standards/why-we-need-standards> (last access April 15, 2016).
- [14] ISO, International Organization for Standardization. Benefits of International Standards 2015. <http://www.iso.org/iso/home/standards/benefitsofstandards.htm> (last access April 15, 2016).
- [15] IEC, International Electrotechnical Commission. Why do we need standards? 2015. http://www.iec.ch/perspectives/general_public/why_do_we_need_standards_1.htm (last access April 15, 2016).
- [16] ISO, International Organization for Standardization. How to write standard: Tips for standards writers 2014. <http://www.iso.org/iso/how-to-write-standards.pdf> (last access April 15, 2016).
- [17] IEEE, Institute of Electrical and Electronics Engineers. 2014 IEEE-SA Standard Style Manual 2014. <https://development.standards.ieee.org/myprojct/Public/mytools/draft/styleman.pdf> (last access April 15, 2016).
- [18] ISO/IEC GUIDE 17:2016. Guide for writing standards taking into account the needs of micro, small and medium-sized enterprises. 2016.
- [19] Bauer N, Huuck R, Lukoschus B, Engell S. A unifying semantics for sequential function charts 2004;3147:400–18.
- [20] Pauwels P, Terkaj W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Autom Constr* 2016;63:100–33. doi:10.1016/j.autcon.2015.12.003.
- [21] BS-ISO/IEC 25010:2011. BSI Standards Publication Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. 2011.
- [22] OMG. Software & Systems Process Engineering Meta-Model Specification. Object Manag Gr Inc 2008. doi:formal/2008-04-01.
- [23] Sanchez A, Rotstein G, Alsop N, Bromberg JP, Gollain C, Sorensen S, et al. Improving the development of event-driven control systems in the batch processing industry . A case study. *ISA Trans* 2002;41:343–63. doi:10.1016/S0019-0578(07)60093-7.
- [24] Godena G. A new proposal for the behaviour model of batch phases. *ISA Trans* 2009;48:3–9. doi:10.1016/j.isatra.2008.08.002.
- [25] Henderson-Sellers B, Gonzalez-Perez C, McBride T, Low G. An ontology for ISO software engineering standards: 1) Creating the infrastructure. *Comput Stand Interfaces* 2014;36:563–76. doi:10.1016/j.csi.2013.11.001.
- [26] Gonzalez-Perez C, Henderson-Sellers B, McBride T, Low GC, Larrucea X. An Ontology for ISO software engineering standards: 2) Proof of concept and application. *Comput Stand Interfaces* 2016;48:112–23. doi:10.1016/j.csi.2016.04.007.
- [27] Vegetti M, Roldán L, Gonnet S, Leone H, Henning G. A framework to represent, capture, and trace ontology development processes. *Eng Appl Artif Intell* 2016;56:230–49. doi:10.1016/j.engappai.2016.10.001.
- [28] MESA. (Manufacturing Enterprise Solutions Association) Business To Manufacturing Markup Language (B2MML) V0600 2015. <http://www.mesa.org/en/B2MML.asp> (last access April 15, 2016).
- [29] Object Management Group. OMG Systems Modeling Language (OMG SysML™) n.d. <http://www.omg.org/spec/SysML/> (last access June 15, 2016).
- [30] Noy NF, McGuinness D. Ontology development 101: a guide to creating your first ontology 2001. http://protege.stanford.edu/publications/ontology_development/ontology101.pdf (last access April 15, 2016).
- [31] Gruber TR. A translation approach to portable ontology specifications. *Knowl Acquis* 1993;5:199–220. doi:10.1006/knac.1993.1008.
- [32] Pardo C, Pino FJ, García F, Piattini M, Baldassarre MT. An ontology for the harmonization of multiple standards and models. *Comput Stand Interfaces* 2012;34:48–59. doi:10.1016/j.csi.2011.05.005.
- [33] Muñoz E, Capón-García E, Laínez-Aguirre JM, Espuña A, Puigjaner L. Supply chain planning and scheduling integration using Lagrangian decomposition in a knowledge management environment. *Comput Chem Eng* 2015;72:52–67. doi:10.1016/j.compchemeng.2014.06.002.
- [34] Cecelja F, Trokanas N, Raafat T, Yu M. Semantic algorithm for Industrial Symbiosis network synthesis. *Comput Chem Eng* 2015;83:248–66. doi:10.1016/j.compchemeng.2015.04.031.
- [35] Morbach J, Wiesner A, Marquardt W. OntoCAPE—A (re)usable ontology for computer-aided process engineering. *Comput Chem Eng* 2009;33:1546–56. doi:10.1016/j.compchemeng.2009.01.019.
- [36] Venkatasubramanian V. DROWNING IN DATA: Informatics and modeling challenges in a data-rich

- networked world. *AIChE J* 2009;55:2–8. doi:10.1002/aic.11756.
- [37] Farreres J, Graells M, Rodríguez H, Espuña A. Towards Automatic Construction of Domain Ontologies : Application to ISA88. *Comput Chem Eng* 2014;871–6.
- [38] Dombayci C, Farreres J, Rodríguez H, Muñoz E, Capón-García E, Espuña A, et al. On the Process of Building a Process Systems Engineering Ontology Using a Semi-Automatic Construction Approach. *Comput. Aided Chem. Eng.*, vol. 37, 2015, p. 941–6. doi:10.1016/B978-0-444-63577-8.50002-4.
- [39] Benaissa B, Bouchiha D, Zouaoui A, Doumi N. Building Ontology from Texts. *Procedia Comput Sci* 2015;73:7–15. doi:10.1016/j.procs.2015.12.042.
- [40] Dahab M, Hassan H, Rarea A. TextOntoEx: Automatic ontology construction from natural English text. *Expert Syst Appl* 2008;34:1474–80. doi:10.1016/j.eswa.2007.01.043.
- [41] Baader F, Calcanese D, McGuinness DL, D. N. F. P-SP, editors. *The Description Logic Handbook*. Cambridge; 2007.
- [42] Hearst MA. Automatic acquisition of hyponyms from large text corpora. *Proc. 14th Conf. Comput. Linguist. -*, vol. 2, Morristown, NJ, USA: Association for Computational Linguistics; 1992, p. 539. doi:10.3115/992133.992154.
- [43] Girju R, Badulescu A, Moldovan D. Automatic Discovery of Part-Whole Relations. *Comput Linguist* 2006;32:83–135. doi:10.1162/coli.2006.32.1.83.
- [44] Girju R, Nakov P, Nastase V, Szpakowicz S, Turney P, Yuret D. Classification of semantic relations between nominals. *Lang Resour Eval* 2009;43:105–21. doi:10.1007/s10579-009-9083-2.
- [45] Aljamel A, Osman T, Acampora G. Domain-specific relation extraction: Using distant supervision machine learning. vol. 1, SciTePress; 2015, p. 92–103.
- [46] McGuire P. *Getting started with pyparsing*. O’Reilly, California, US 2007.
- [47] ISA. *Batch Control, Part 1: Models and Terminology, ISA-88.01-1995 (R2006)*. ISA Committee; 2006.
- [48] ISA. *Batch Control Part 2: Data Structures and Guidelines for Languages, ANSI/ISA-88.00.02-2001*. ISA Committee; 2001.
- [49] ISA. *Batch Control Part 3: General and Site Recipe Models and Representation, ANSI/ISA-88.00.03-2003*. ISA Committee; 2003.
- [50] ISA. *Batch Control Part 4: Batch Production Records, ANSI/ISA-88.00.04-2006*. ISA Committee; 2006.
- [51] Fischer Bandy J, Emerson D. Common resources in SP88 style applications presented at the World Batch Forum. *ISA Trans* 1996;35:267–73. doi:10.1016/S0019-0578(96)00036-5.

Glossary

ISA88:	ISA88 Standard
ISA88R2006:	ISA88 Standard Part 1 (Batch Control Part 1: Models and Terminology) published in 2006 as a revision of first version [47]
ISA882010:	ISA88 Standard Part 1 (Batch Control Part 1: Models and Terminology) published in 2010 [3]
NL:	Natural Language
Part 2:	ISA88 Standard Part 2 (Batch Control Part 2: Data Structures and Guidelines for Languages) [48]
Part 3:	ISA88 Standard Part 3 (Batch Control Part 3: General and Site Recipe Models and Representation) [49]
Part 4:	ISA88 Standard Part 4 (Batch Control Part 4: Batch Production Records) [50]

Table 1. Text and ontology comparison

Text	Ontology
ambiguity of NL: polysemy, synonymy, ellipsis, implicit knowledge	formal and strict definition of concepts, meanings and relations
need for interpretation (while reading).	no need for interpretation
different interpretation for each human	one interpretation always and for everyone
written for humans, unusable by computers	structured for computers, complex for humans

Table 2. Selected case studies with which part of the standard that they occur

Case Name	Sentence	Part I R2006	Part I 2010	Other parts
Batch Control	‘ batch control: Control activities and control functions that...’	Y	Y	-
Recipe Element	‘ recipe element: a structural entity that ...’	Y	N	-
	‘ A recipe element is a representation of ...’	N	N	P4
Control Module	‘... equipment entities such as units, equipment modules, and control modules. ’	Y	N	-
	‘...the lower level entities, such as equipment modules and control modules.’	Y	Y	-
	‘ control module: The lowest level grouping of equipment ...’	Y	Y	-
Procedure	‘... the procedural element, such as procedure, ...’	N	N	P4
	‘ procedure: The strategy ...’	Y	N	-

Control Recipe	'...include data such as control recipes , ...'	N	N	P4
	' control recipe : A type of recipe which ...'	Y	Y	-
Equipment Module	'... equipment entities such as units , equipment modules ...'	Y	N	-
	' equipment module : A functional group of equipment that ...'	Y	Y	-
	'... lower level entities , such as equipment modules ...'	Y	Y	-
Unit Procedure	'... elements of batch production such as campaigns , unit procedures ,...'	Y	N	-
	'... recipe or equipment procedural element smaller than a complete batch, such as a unit procedure , ...'	N	N	P4
	'... procedural element , such as procedure , unit procedure , ...'	N	N	P4
	' unit procedure : A strategy for ...'	Y	Y	-
Phase	' Phase The smallest element of procedural control that ...'	Y	N	-
	'... recipe or equipment procedural element smaller than a complete batch, such as a unit procedure, operation, or phase .'	N	N	P4
Master Recipe	'... data such as control recipes , master recipes , ...'	N	N	P4
	' master recipe : A type of recipe that ...'	Y	Y	-
	'... sources such as other types of schedules, master recipes , ...'	Y	Y	-
	'A master recipe is a template recipe that ...'	N	N	P4
General Recipe	' general recipe : A type of recipe that ...'	Y	Y	-
	'The general recipe is an enterprise level recipe that ...'	Y	N	-
	'A general recipe is a container of ...'	N	N	P3
	'A general recipe is a corporate recipe that ...'	N	N	P3
	'A general recipe is an enterprise-wide recipe that ...'	N	N	P3
	'A general recipe is a type of an equipment-independent recipe .'	N	N	P3

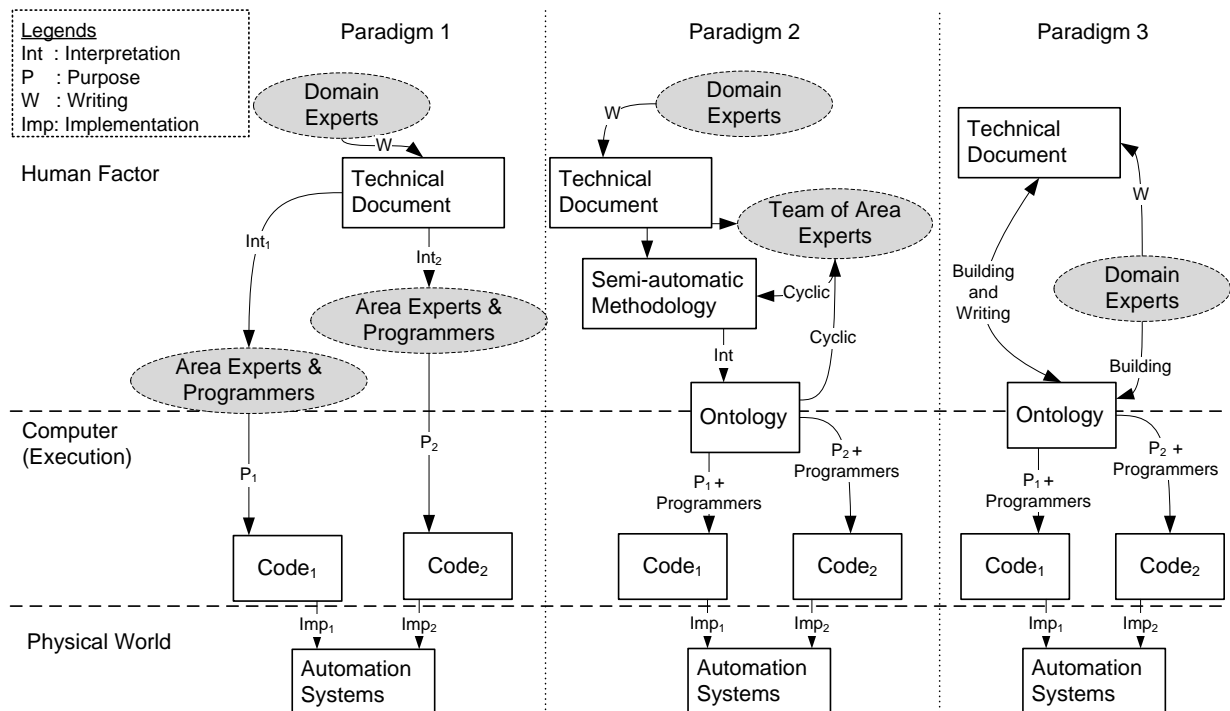


Figure 1. Demonstration of explained paradigms

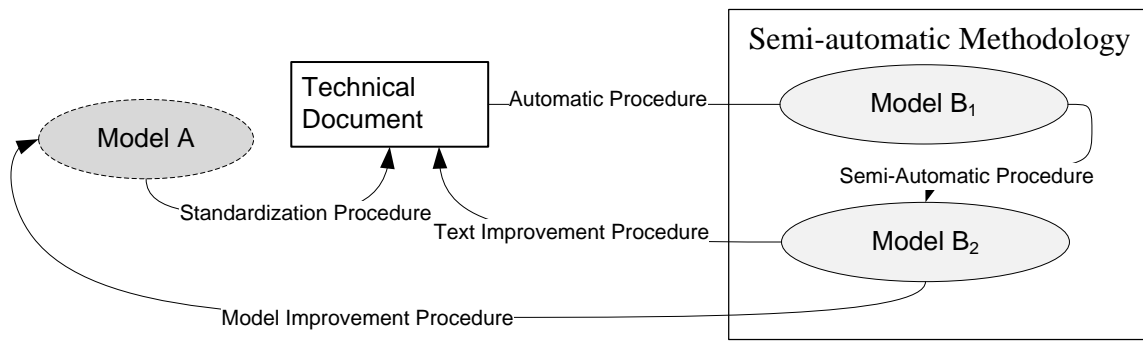


Figure 2. General view of the methodology

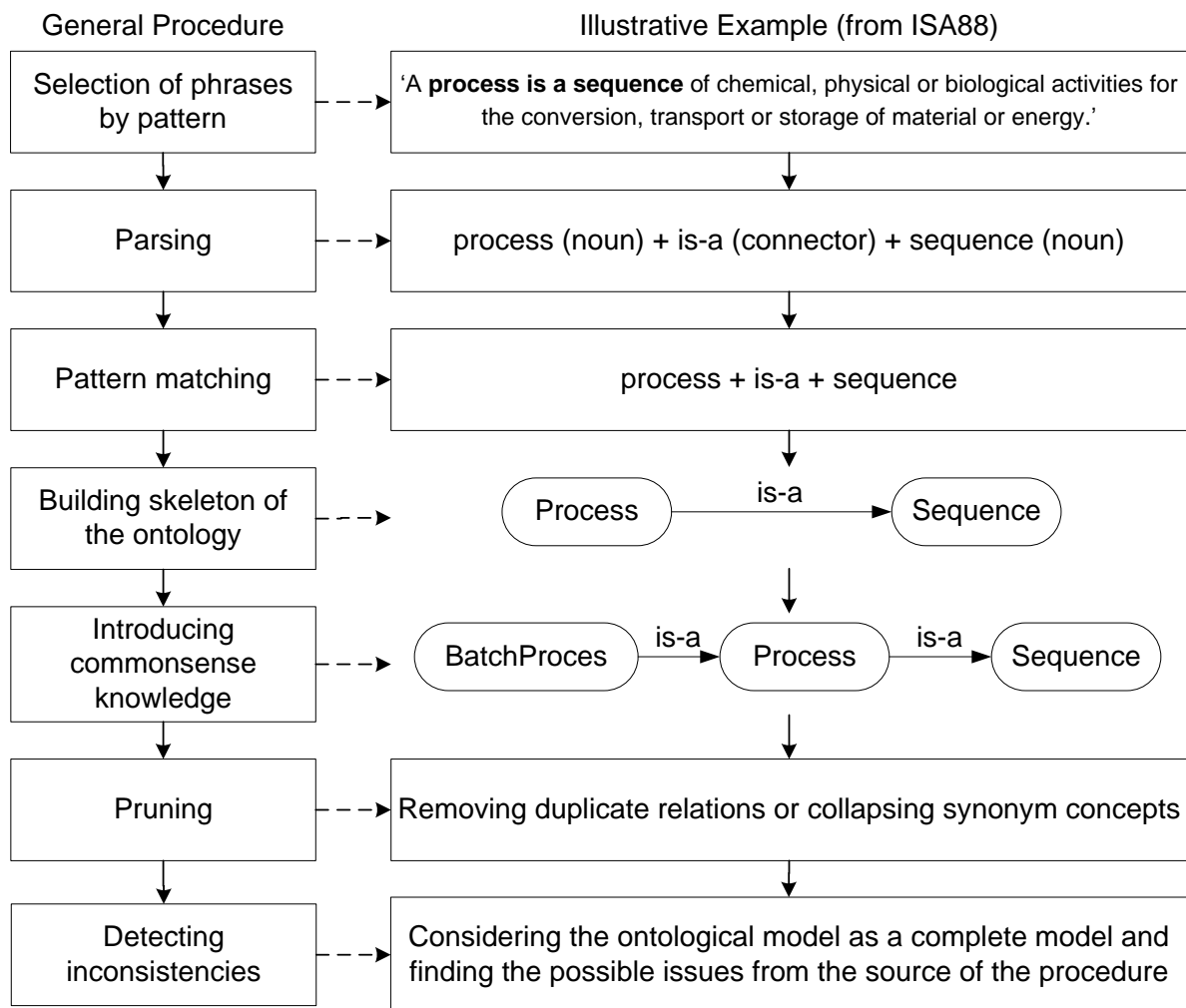


Figure 3. General Produce Overview

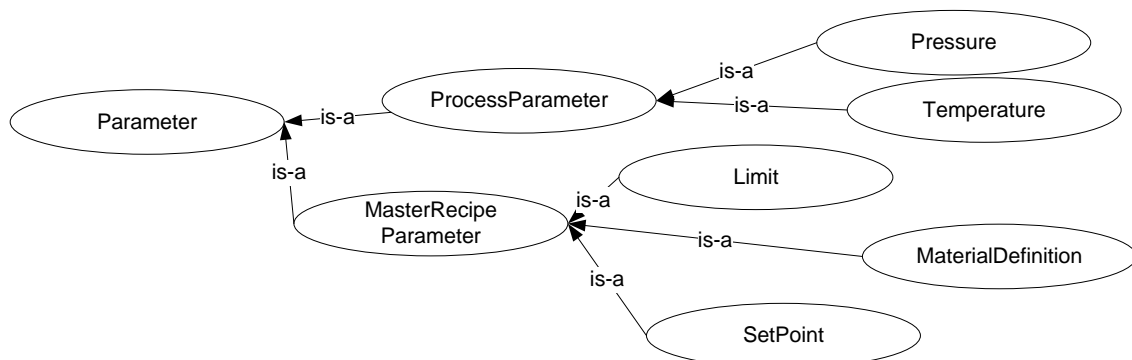


Figure 4 ProcessParameter Concept and its is-a relations

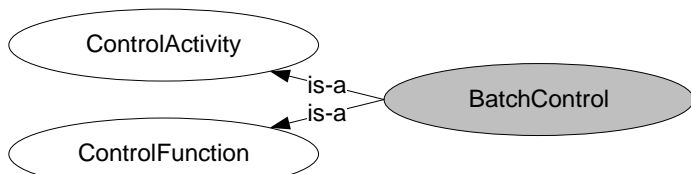
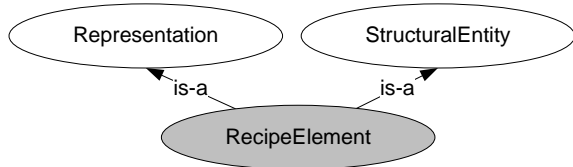
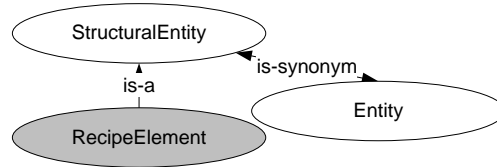


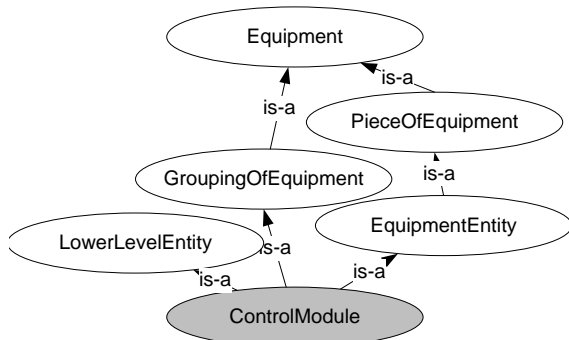
Figure 5. Batch Control



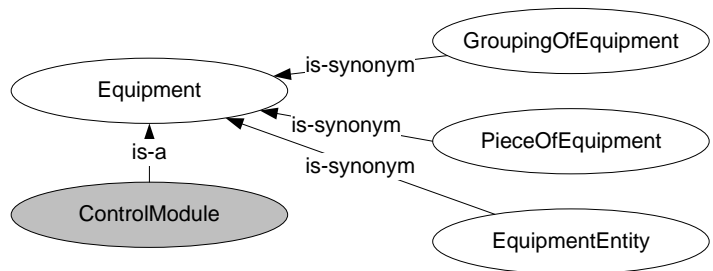
(a) Original Ontology
Figure 6. Recipe Element



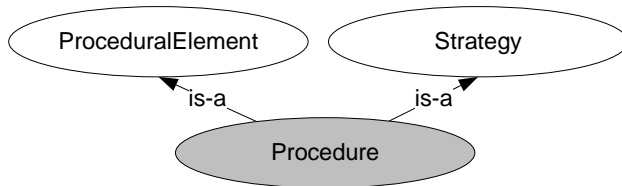
(b) Revised Ontology



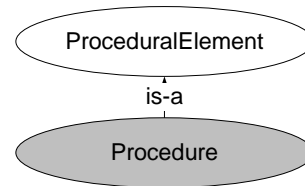
(a) Original Ontology
Figure 7. Control Module



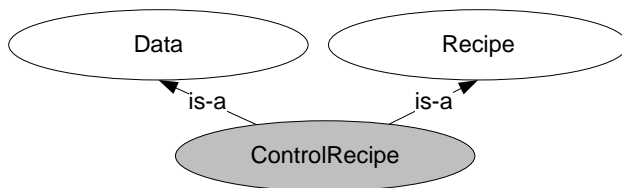
(b) Revised Ontology



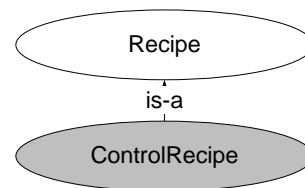
(a) Original Ontology
Figure 8. Procedure



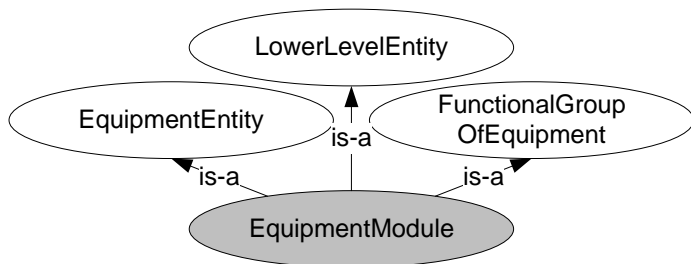
(b) Revised Ontology



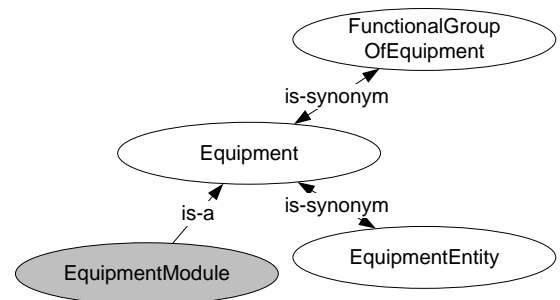
(a) Original Ontology
Figure 9. Control Recipe



(b) Revised Ontology



(a) Original Ontology
Figure 10. Equipment Module



(b) Revised Ontology

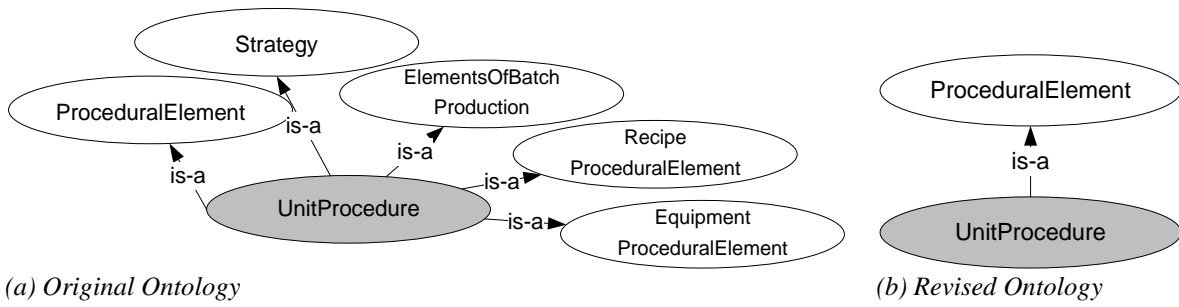


Figure 11. Unit Procedure

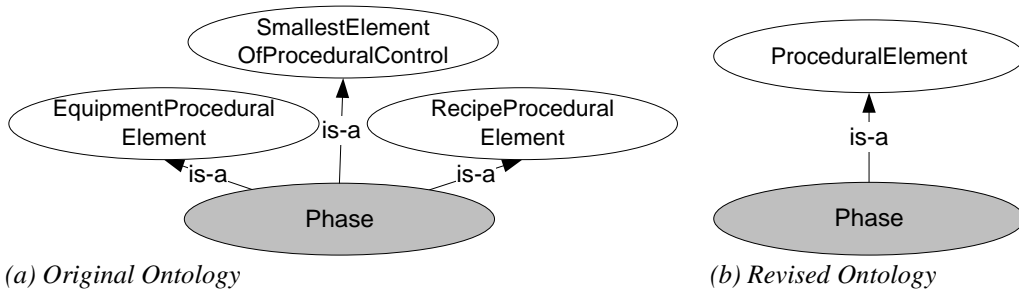


Figure 12. Phase

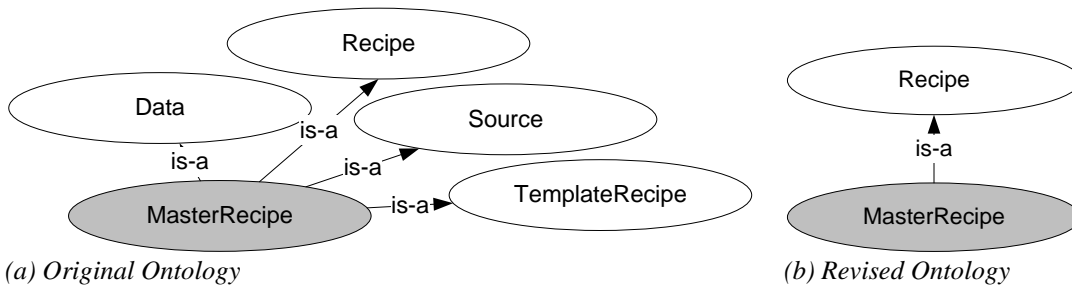


Figure 13. Master Recipe

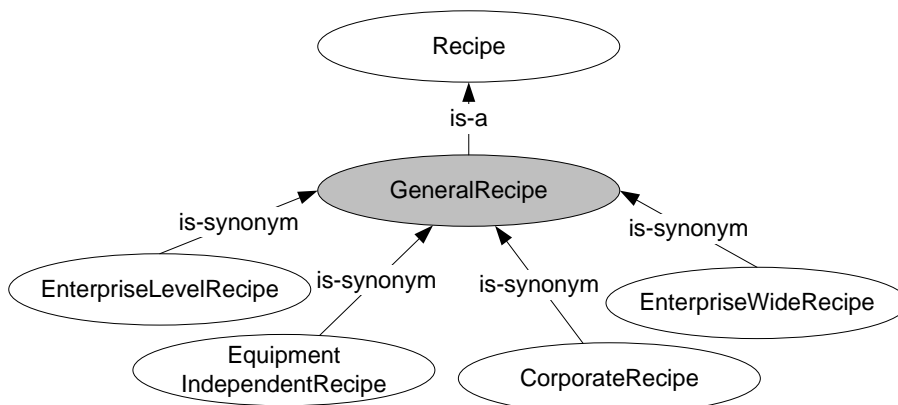
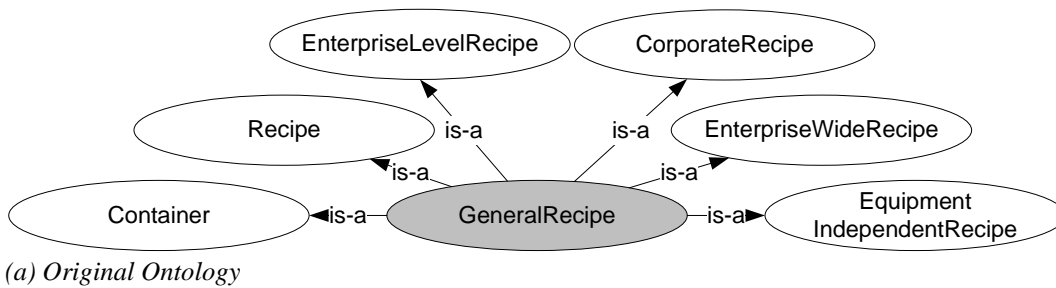


Figure 14. General Recipe