Computers and Chemical Engineering 42 (2012) 217–234

Contents lists available at SciVerse ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Ontological framework for enterprise-wide integrated decision-making at operational level

Edrisi Muñoz[a], Elisabet Capón-García[b], Antonio Espuña[b], Luis Puigjaner[b,*]

[a] Centro de Investigación en Matemáticas, Jalisco S/N, 36240 Guanajuato, Mexico
[b] Department of Chemical Engineering, Universitat Politècnica de Catalunya ETSEIB, Avda. Diagonal, 647, E-08028 Barcelona, Spain

ARTICLE INFO

Article history:
Received 18 September 2011
Received in revised form 27 January 2012
Accepted 2 February 2012
Available online 11 February 2012

Keywords:
Process scheduling
Multiproduct plant
Multipurpose plant
Ontology
Decision-levels integration

ABSTRACT

In the domain of chemical process engineering, there is an increased interest in the integration of the enterprise hierarchical levels for decision-making purposes. At the scheduling level, decisions on the allocation of tasks to resources, sequencing and timing of tasks must be managed. However, such decisions are directly related to other enterprise actions, such as control and planning, but they are difficult to coordinate because they are modeled at different time and space scales, and their goals are not the same. In order to achieve integrated decisions supported by high quality information, there is a need to improve and develop robust computational tools and consistent models. In general, scheduling optimization approaches for decision-making differ depending on problem features, such as physical layout or time representation. Therefore, this work focuses on providing a framework based on a semantic model that captures the diversity in scheduling problem representation. Such semantic model uses the master recipe concept from the ANSI/ISA-88 standard perspective and encapsulates the scheduling decision task features. As a result, by the use of a single representation approach, any scheduling problem can be modeled and solved by its adequate optimization tool. The potential of a general model representation is presented by means of several case studies related to the scheduling function. Such case studies shed light to the model capabilities to represent different kinds and particular scheduling problems, achieving integration at the different decision support levels.

1. Introduction

Decision Support Systems (DSS) are related to manufacturing indicators, such as economic efficiency, product quality, flexibility or reliability. Global competition has made essential for the viability of the enterprise the use of DSS for assisting in the decision-making process. In this sense, new tools for integrating the enterprise hierarchical decision levels, which differ in time and space scales, should be developed in order to ensure the enterprise competitiveness (Harjunkoski, Nystrom, & Horch, 2009).

The first requirement to achieve such integration is to define standardized information structures and more sophisticated information tools to exploit them, in order to improve the availability and communication of data between different decision levels and also achieve consistent models behind the corresponding decision support tools. In this framework, the role of infrastructures that continuously and coherently support fast and reliable decision-making activities related to the production process is now of paramount importance (Venkatasubramanian et al., 2006).

Ontologies stand for a formal specification of the domain, that is, a body of formally represented knowledge based on conceptualizations, which are abstract, simplified views of the physical or procedural elements. On the one hand, the ontology elements should be part of the model intended to represent a system for some purpose, managing the relationships that hold among the elements of the model, thus allowing the ontology to be usable (Gruber, 2008). On the other hand, reusing ontologies requires not only consideration of the ontologies, but also of the tasks for which they are intended.

This work presents the use of an ontology as an improved modeling and communication tool in the process domain, specifically the scheduling and control functions in process industry.

Scheduling usually involves deciding on the amount of products, the allocation of resources to the needed tasks, the order in which the different batches are to be performed and at what time these tasks start. This problem is stated as an optimization problem that seeks for minimizing the production makespan, lateness, earliness, or any performance function that could be adopted. The production process itself is defined in the production recipe, which contains the stages that are to be performed in different units, and the operations that are to be carried out at each stage. In addition, the constraints regarding operational timing such as

* Corresponding author. Fax: +34 934 010 979.
E-mail address: luis.puigjaner@upc.edu (L. Puigjaner).

0098-1354/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2012.02.001

simultaneity or sequence must be described. Other issues, such as material balances, resource consumption and processing times are also described in the production recipes.

Process control performs several functionalities in batch processes management. On the one hand, the batch operation model should be implemented in a feed-forward manner. This model is defined in the named control recipes, obtained from the optimization solution of the scheduling system and the master recipes which have been optimized in a previous process design step. Control recipes are composed by a serial of process operations, phases and the transition logics to be implemented in a specific equipment piece. Additionally, processing conditions, actuation variables and set points for each stage are given in the control recipes. On the other hand, basic control is carried out. Firstly, automatic feedback control is used to reduce the influence of uncertainty and reject intra-batch disturbances, since in batch processing it is crucial to satisfy quality specifications in each batch. In addition, supervision may be performed to improve processing conditions from run to run and this constitutes a recipe adjustment. Process monitoring and fault diagnosis are also employed to handle faults or exceptions and perform manual actions, if required. Finally, interlock is used to provide safety or to avoid mistakes in processing the batch. As for the performance objectives in process control level, they can be either economic (for example, to maximize profit or profitability) or processing in type (for example, to accomplish customer specifications, to minimize product variability, to meet safety or environmental regulations or to maximize the plant flexibility with reliable control).

2. Integration of decision-making in enterprise structure

Recent trends in process industries are shifting the focus from controlling the process plant as a stand-alone entity toward managing it as an integral part of a larger system (Klatt & Marquardt, 2009). Such approach aims at exploiting the process and environment dynamics in order to maximize the plant economic indicators. Obviously, such understanding of process management entails the integration of information and decisions across the different hierarchical levels. Therefore, a current important challenge lies on the coordination of the decision-making and the optimization of different decision levels, both vertically across a single process plant, and horizontally along the different geographically distributed subsystems of the supply chain in a given time horizon.

One first step toward such integration consists of the sharing of information, which is nowadays being achieved with modern IT tools, such as SAP and Oracle, that allow the instantaneous flow of information along the various organizations in a company (Grossmann, Erdirik-Dogan, & Karuppiah, 2008). However, a better understanding, structuring and even modeling of the whole process is necessary for an effective transformation of the information into knowledge. In this line, several standards are used in enterprises in order to improve their efficiency and flow of information, such as CAPE-OPEN, the ANSI/ISA-88 or the ANSI/ISA-95. Thus, semantic technologies seem to offer an appealing way to capture knowledge and integrate information, for supporting a smooth integration of information and mathematical modeling in a single modeling framework (Klatt & Marquardt, 2009).

Another major issue consists of the modeling and optimization approaches for integration among decision levels. In fact, decision-making levels of the enterprise structure are forced delimitations of the actual reality which aim to tackle the problem complexity. Hence, coordinated procedures are needed, which can maintain a certain degree of independence of subsystems, while at the same time aiming at the objectives of integrating optimization of the overall system (Grossmann et al., 2008). In addition, the development of procedures that can effectively work across large spatial and temporal scales, as well as global model-based optimization techniques (Grossmann, 2005) are crucial for attaining global solutions.

From a process scheduling perspective, recent reviews highlight the actual importance of its integration both upward (Grossmann et al., 2008; Kallrath, 2005; Maravelias & Sung, 2009), mid- and long-term planning, and downward (Grossmann et al., 2008; Harjunkoski et al., 2009), process and control, within the decision level pyramid. Therefore, the objective of plant operation is moving from controlling the plant at its set-point toward optimizing its performance in real-time subject to process, environmental, and others (such as quality and services) constraints (Klatt & Marquardt, 2009).

2.1. Scheduling and control integration

Recently, Harjunkoski et al. (2009) present the issue of scheduling and control integration from an industrial and academic perspective. The authors state that the scheduling and control functions aim at filling the gap between enterprise resources planning (ERP) and operations by ensuring that business targets are correctly transferred to the production level. On the one hand, the scheduling function is directly related to the planning level, since production objectives and time horizon are compulsory data for scheduling. On the other hand, the control level is directly related to product quality, equipment monitoring and other equipment related activities.

In order to manage the aforementioned functions and apply optimization theory, the use of standards can be adequate, since they allow re-usability of the solutions and components, improve the connectivity with vendors and provide with a unified data structure. In the scheduling and control scenario, the standard ANSI/ISA 88 (IEC61512) (International Society for Measurement and Control, 2001) was approved in 1995 by the International Society of Automation, a consortium of academics and industrialists; their purpose was to overcome the existing difficulties in batch automation. This standard provides a framework that an engineer can use to specify automation requirements in a modular fashion and can be used for integrating batch-related information and formalizing the description of the scheduling and control decision levels and the whole production plant, including data, information and knowledge required for the decision-making. However, the two main drawbacks in the standards application consist of (i) the lack of the specification of its implementation, and (ii) the great difficulty in its understanding.

The integration between control and scheduling should emerge from the functionality point of view (Harjunkoski et al., 2009), and should fulfill the workflow tasks within the production planning environment; in which, scheduling contains the production schedule, the actual production information and the production capacity, whereas the control function is related to the tasks regulated by the process control system.

Three main problems have been identified when aiming at the integration of the control and scheduling levels according to Harjunkoski et al. (2009):

- *Methodological aspects.* The amount of details and problem complexities increase toward the control direction, and most of the methods used in the lower level can be applied to handle upper level complexities. Therefore, the time granularity and level of detail are the two main issues that must be tackled when facing integration.
- *Information transfer.* The information flow between the control and scheduling systems must be standardized, and it can be either

unidirectional or bidirectional, which is the most common and challenging case.
- *Modeling approaches.* The way the scheduling and control problems are formulated and solved differs widely. In fact the modeling approaches are not straightforward compatible with each other, and it is difficult to capture efficiently all the problem aspects. At lower level, an important question is how to couple scheduling models with process models, and particularly with dynamic models that can rigorously predict the optimal control (Grossmann et al., 2008). With today's knowledge and tools, both levels cannot be fully merged, but need to jointly find better and more natural ways of collaborating. Therefore, it is crucial to focus on the actual needs for optimization, and on the understanding of the system, in order to obtain the trade-offs between shorter solution times and the solution quality.

In general terms, the integration of scheduling and control requires intensive involvement of other areas, and their collaboration for improving modeling and optimization approaches and software architectures. Thus, the integration can be tackled by using adaptation through parameters, improving the information flow, the modularity, the data availability, the use of standards, and adopting general objective functions (Harjunkoski et al., 2009). The amount of academic contributions to this specific field of integration has increased along the last years, although it is still very scarce.

Furthermore, scheduling optimization approaches for decision-making differ depending on the problem features such as physical layout or time representation. For this reason, in order to unify different conceptual scheduling models, this work provides a general semantic model which encompasses the most important features of the existing models. In addition, such model also includes the control decision level allowing the potential integration of both decision levels.

This work focuses on providing a semantic model, namely an ontology, which deals with diversity in scheduling problem representation and allows effective data sharing and information flow. Specifically, this model aims to adequately provide the necessary data which will be used by the optimization tool and the decision maker. Thus, the decision maker has the opportunity of creating high information quality and reaching better solutions, in terms of shorter time of response and choices. As a result, an effective data-information cycle is performed.

Such semantic model uses the master recipe concept from the ANSI/ISA-88 standard perspective capturing the scheduling and control decision tasks features. Furthermore, the detail of the master recipe has been specified in the semantic model in order to consider mass balances as well as sequencing and timing constraints.

Moreover, by the use of a single representation approach consisting of the ontological model, any scheduling problem can be modeled and solved by the adequate optimization tool as will be shown in the case studies.

3. Ontological model

Ontologies are hierarchical domain structures that provide a domain theory, have a syntactically and semantically rich language, and a shared and consensual terminology (Klein, Fensel, Kiryakov, & Ognyanov, 2002). As it is well known, the design, construction and operation of chemical plants are considered the major engineering activities (Morbach, Wiesner, & Marquardt, 2009) and, in this way, the ontological framework model should represent not only the terminology, but also the entire domain of chemical engineering processes, with a particular attention to the integration of control activities at different levels. Previous work toward an integrated environment (Muñoz, Capon-Garcia, Moreno-Benito, Espuna, & Puigjaner, 2011; Muñoz, Espuña, & Puigjaner, 2010) presented a batch process ontology and a design strategy. However, only multi product plants were tackled and the integration with any optimization framework was not studied. In this paper special emphasis is given to the development of the master recipe in order to represent any kind of batch plant and to provide the necessary inputs for subsequent optimization.

Following the methodology proposed by Muñoz et al. (2010), new functionalities have been added to previous model versions. Such methodology is based on a Plan, Do, Check/Study and Act Cycle (PDSA), which results in an ordered sequence of steps, which are easy to understand and track (Fig. 1). In this figure, as a first step, the Plan phase is found whose aim is to guide the process development and establish the main targets. Secondly, the Do phase is in charge of implementing the work toward ontology building (knowledge management and software implementation) with help of the previously planned. The following phase is the Study where a full checking of domain standards conformity, ontological reasoning and the revision of the implementation are done in order to check the success in the application of the proposed methodology. Next, the Act phase ensures the correct implementation, and carries out the operations related to the ontology maintenance. Finally, the re-planning consists of a SWOT analysis (Strengths, Weaknesses, Opportunities and Threats) over the life of the ontological structure (the ontological model + computer application).

Next the different phases of the methodology are described.

3.1. Plan phase

3.1.1. Project general information

As a first step for the project development, the information related to the project must be documented for future records. The following data must be provided:

- Project name (ontology name): *Integrated process system (master recipe model).*
- Date: December 2010 to February 2011.
- Creators: Edrisi M. and Elisabet C.
- Other information: UPC CEPIMA group.

3.1.2. Domain definition

The scope of this ontology lies in the planning and programming functions considered in the structure of the company as belonging to the area of the process systems engineering. Such decision levels have been thoroughly described in Sections 1 and 2. As a whole, the scheduling problem is stated as an optimization problem that seeks for minimizing the production makespan, lateness, earliness, or any performance function that could be adopted. The production process itself is defined in the production recipes, which contain the stages to be performed in different units, and the operations to be carried out at each stage. In addition, the constraints regarding operational timing such as simultaneity or sequential must be described. Other issues, such as the material balance, the resource consumption and processing times are also described in the production recipes.

In general, the different parts of a company may require different types of information about the product or the process to manufacture it. Particularly, recipes hold the necessary information for the different companies functions and hierarchical levels. The ANSI/ISA-88 defines the master recipe, which is adopted in this work, in order to (i) guide the scheduling problem instantiation, (ii) facilitate the integration with other decision levels (control level), and (iii) provide the necessary information for the optimization tools.

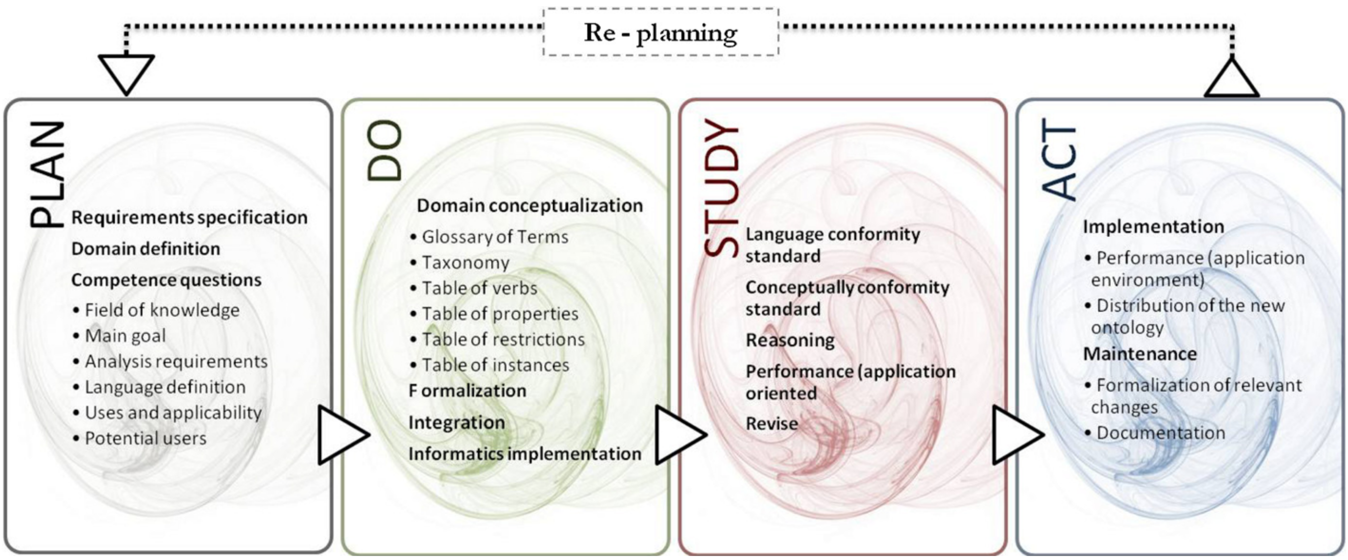

Fig. 1. Methodology for developing enterprise ontology.

The master recipe is targeted to a processing area and is derived from either a general or site recipe. Master recipes depend on equipment types or classes. These recipes can contain product-specific information required for detailed scheduling, such as equipment requirements. Furthermore, the ANSI/ISA-88 batch control requires a master recipe, which is the template for recipes used to create individual batches. Master recipes take into consideration the equipment requirements within a given process cell. They include the following information categories: header, formula, equipment requirements and procedure. Therefore, the master recipe contains the information required for scheduling and is the keystone for scheduling and control integration.

Knowledge sources: As a source for the domain knowledge for the planning and scheduling function the following documents are considered.

ANSI/ISA-88 standard (International Society for Measurement and Control, 1995, 2006, 2007; Shirasuna, 2007), allows to create a general infrastructure to be applied to any process system. The ANSI/ISA-88 representation provides an advantage of establishing a more general conceptualization in the batch process domain. Such a generalization is behind years of joint work by recognized batch manufacturing experts who met to define a perceptive view of batch plants organization and its corresponding hierarchy of control functions. As a consequence, following the ANSI/ISA-88, virtually all activities concerning batch processes can be properly represented.

The ANSI/ISA-88 standard is composed of five parts, three of which have been used as knowledge source for the domain definition:

- ANSI/ISA-88.01-1995 Batch Control Part 1: Models and terminology
- ANSI/ISA-88.00.02-2001 Batch Control Part 2: Data structures and guidelines for languages
- ANSI/ISA-88.00.03-2003 Batch Control Part 3: General and site recipe models and representation

The basic relations can be taken from the assertions of the models that ANSI/ISA-88 describes. These assertions show the basic relations between physical model, recipe model and procedural model and are described by entity-relationship diagrams (E-R diagrams). For ANSI/ISA-88, recipes are needed because of the set of information that uniquely identifies the production requirements for a specific product.

Thus, the recipe specifications provided by the World Batch Forum (Brandl & Emerson, 2003) are also considered. Such information are schemes based on the ANSI/ISA standards.

3.1.3. *Ontology motivation*

The ontology motivation refers to the definition of the aim of the ontology, which could answer the following questions: (i) what is going to be modeled?; (ii) which are the targets of the model?; (iii) which is its scope?; (iv) how will be the model implemented?; (v) who, where and how will use the model? In order to accomplish this step, the following information must be fulfilled:

(1) Field of knowledge: Chemical process industry planning and scheduling function.
(2) Main goal: The creation of an ontological framework capable of representing any kind of scheduling problem within the enterprise structure.
(3) Analysis requirements: A thorough review of the planning and scheduling requirements is given in previous work (Muñoz et al., 2010). At such level, the process system involves multiple and interrelated activities that are performed at single or multiple sites, with different durations and amounts of information. Generally, information flows from the marketing department to the manufacturing department, which determines the production schedule that is needed to meet the sales strategies. In its most general form, the scheduling problem requires information that is related to the configuration of the plant (the available equipment units and resources), the product recipes (the set of processing tasks and resources required to manufacture a given product), precedence relationships between materials and final product requirements (demands and related due dates).
(4) Language definition: OWL has been selected to meet the need for a Web Ontology Language, which is part of the growing stack of World Wide Web Consortium (W3C) recommendations that are related to the Semantic Web. This language was used previously, and is maintained for this extension.
(5) Uses and applicability: The ontological framework should be used for helping in the scheduling decision support-task, as a tool for extracting information quality as required at

the scheduling level. The framework will also be used as a quantitative support tool improving the use of optimization models at scheduling level, since it will provide the necessary data for the optimizers.

(6) Potential users: Any enterprise environment in which automation has been implemented or automation is desired, can benefit from the formal specification, which means that it can be read by a computer, of the proposed ontology because it is based on the ANSI/ISA standards. Moreover, the schedulers can directly use the proposed framework because it is an informal specification of the domain, which means that knowledge is available in a human-readable form.

3.2. Do phase

As mentioned in the previous section, the modeling is based on the ANSI/ISA standards. Now that the knowledge sources and the goals of the project have been defined, it is time to formalize the model through the construction of the ontology. This means the implementation of the work based on the previous Plan phase.

3.2.1. Domain conceptualization

- Glossary of terms: Table 1 defines the terminology associated with the planning and scheduling level based on the ANSI/ISA-88 standard.
- Taxonomy: The aforementioned terms are formalized in a taxonomic manner. Specifically the whole domain is organized in nine top terms or classes, as shown in Fig. 2. Each top class unfolds its corresponding subclasses. Thus, Fig. 3 contains an extract of the whole taxonomy, including the elements directly related to the master recipe.
- Table of properties: In order to identify the object type properties an interrelationship matrix has been constructed. This is a square matrix where all the concepts are placed in both axes (X and Y) as headers. At each intersection between concepts, a relation must be defined if it exists. This process must be run twice, the first one is X/Y in top-down direction and the second is Y/X in left-right direction. As a result of the properties identification process, Tables 2 and 3 contain the object type properties and the data type properties of the enterprise ontology project, respectively, as well as their domain and their range. Specifically, a total of 113 object properties and 10 data properties are specified.

Table 1
Concepts from ANSI/ISA-88.

Name	Description
Batch process	A process that leads to the production of finite quantities of material by subjecting quantities of input materials to an ordered set of processing activities over a finite period of time using one or more pieces of equipment.
Recipe	The necessary set of information that uniquely defines the production requirements for a specific product. There are four types of recipes defined in S88.01: general, site, master, and control. A recipe may contain a header, a formula, an equipment, requirements, a procedure and other information.
Header	Information about the purpose, source and version of the recipe such as recipe and product identification, creator, and issue date.
Formula	A category of recipe information that includes process inputs, process parameters, and process outputs.
Process	A sequence of chemical, physical, or biological activities for the conversion, transport, or storage of material or energy. A process consists of an ordered set of recipe elements.
Site	A component of a batch manufacturing enterprise that is identified by physical, geographical, or logical segmentation within the enterprise. A site may contain areas, process cells, units, equipment modules, and control modules.
Area	A component of a batch manufacturing site that is identified by physical, geographical, or logical segmentation within the site. An area may contain process cells, units, equipment modules, and control modules.
Process cell	A logical grouping of equipment that includes the equipment required for production of one or more batches. It defines the span of logical control of one set of process equipment within an area. This term applies to both the physical equipment and the equipment entity. It must contain unit.
Unit	A collection of associated control modules and/or equipment modules and other process equipment in which one or more major processing activities can be conducted. Units are presumed to operate on only one batch at a time. Units operate relatively independently of one another. This term applies to both the physical equipment and the equipment entity. A unit may contain equipment modules, and control modules.
Recipe management	The control activity that includes the control functions needed to create, store, and maintain general, site, and master recipes.
General recipe	A type of recipe that expresses equipment and site independent processing requirements.
Site recipe	A type of recipe that is site specific. Site recipes may be derived from general recipes recognizing local constraints, such as language and available raw materials.
Master recipe	A type of recipe that accounts for equipment capabilities and may include process cell-specific information.
Control recipe	A type of recipe which, through its execution, defines the manufacture of a single batch of a specific product.
Batch schedule	A list of batches to be produced in a specific process cell. The batch schedule typically contains such information as what is to be produced, how much is to be produced, when or in what order the batches are to be produced, and what equipment is to be used.
Common resource	A resource that can provide services to more than one requester. Common resources are identified as either exclusive-use resources or shared-use resources.
Exclusive-use resource	A common resource that only one user can use at any given time.
ID	A unique identifier for batches, lots, operators, technicians, and raw materials.
Process parameter	Information that is needed to manufacture a material but does not fall into the classification of process input or process output.
Shared-use resource	A common resource that can be used by more than one user at a time.
State	The condition of an equipment entity or of a procedural element at a given time. The number of possible states and their names vary for equipment and for procedural elements.
Production planning and scheduling	The activity that is formed by decision algorithms for producing plan production.
Production information management	The activity that is in charge of the compilation, processing and notification of the information of production.
Requirement	A singular documented need of what a particular product or service should be or do.
Production process	The production process is concerned with transforming a range of inputs process into those outputs that are required by the market by the use of transforming resources. There are three main types of production process: job, batch and flow production.
Recipe element	The necessary set of information that uniquely defines the production requirements for a specific product.
Procedural logic	The necessary set of links that uniquely defines the production sequence of recipe elements within a given recipe.
Equipment requirement	The necessary equipment entities to perform a recipe.

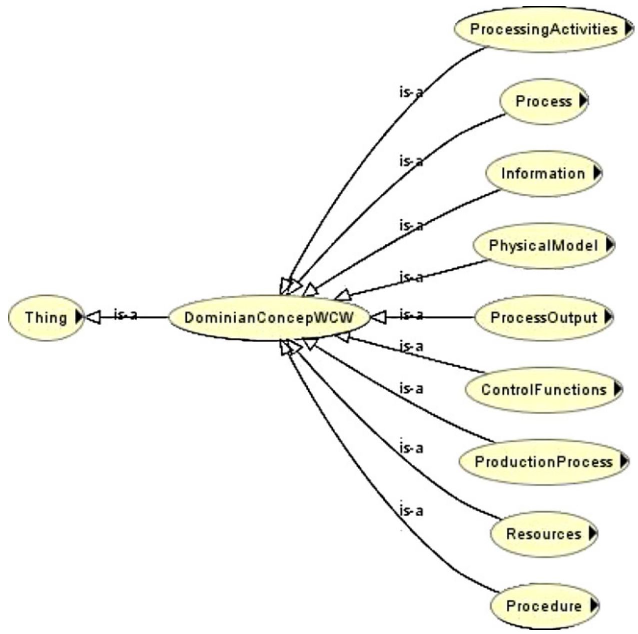

Fig. 2. Taxonomy of the top classes (domain).

- Table of restrictions (axioms): The restrictions build constraints between the properties above mentioned, are general for the domain and are related to the classes of the ontology. Some examples of restrictions are quantifier restrictions, cardinality restrictions and hasValue restrictions. A total of 73 axioms have been defined for the proposed ontological model. Fig. 4 contains the restrictions related to the master recipe class and the site class.

Fig. 5 presents the UML diagram of an extract of the resulting ontological, which contains the more relevant classes and properties for the application in this work.

- Tables of instances: They are created when a specific case study is represented. For this reason, the table of instances is given in Section 4. In this way, the property of usability is demonstrated since one general model (the ontology developed in this work) can be used by different process realities (cases studies).

3.2.2. Formalization (ontology editor implementation)

The selected editor as in previous works is Protégé (Biomedical Informatics Research, BMIR), a description logic reasoning system used as a tool for ontology editing and knowledge acquisition (Horridge, Jupp, et al., 2007). The ontological framework has been implemented in a Intel Core2 Quad CPU at 2.83 GHz.

3.2.3. Integration

There is a lack of consensus among existing ontologies, regarding issues such as the language of construction, the way of structuring knowledge, the terminology used, among others. For this reason, it has not been possible to consider other ontologies for their integration (reuse) in this work.

3.2.4. Informatics implementation and application

The main objective established for developing the informatics system is its capacity to integrate different perspectives (e.g., different hierarchical decision levels) and the mappings among them. The description of the informatics implementation and application is given in previous work (Muñoz et al., 2011). For the applications, Java has been used as a high-level programming language. Java presents a good versatility, efficiency and security. Thus, Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs), exist for most operating systems.

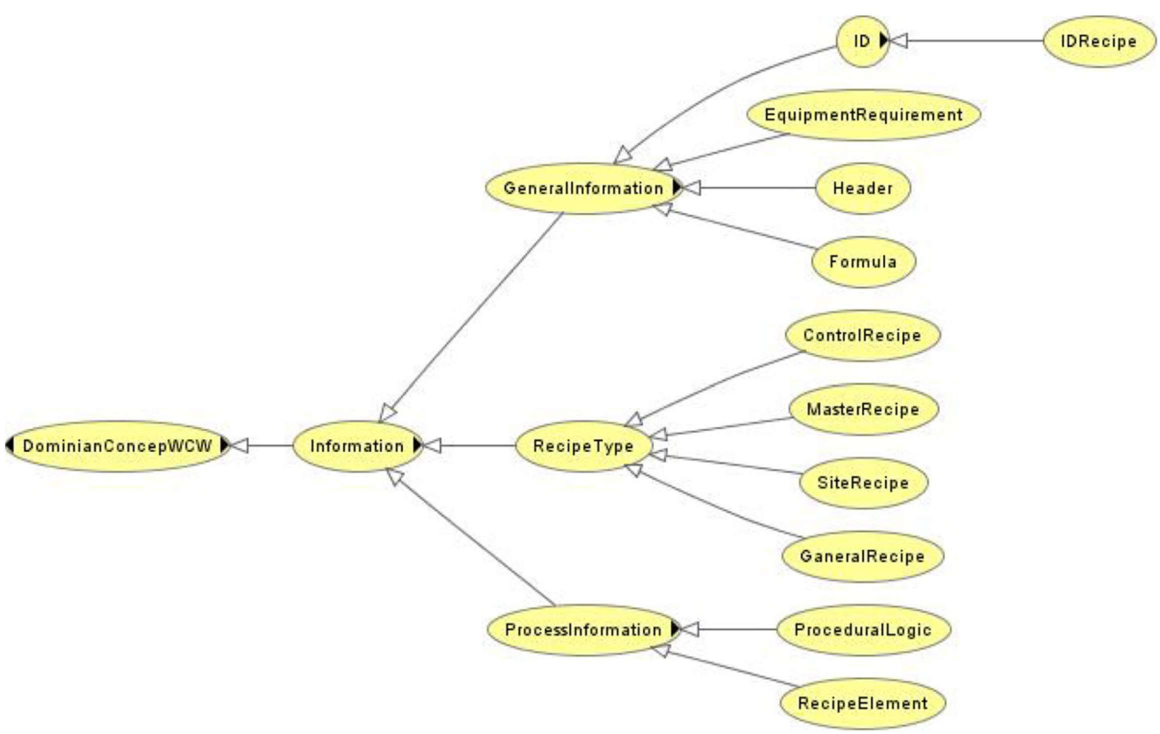

Fig. 3. Taxonomy extract containing the master recipe elements.

Table 2
Relevant object properties (domain and range).

Object property	Domain	Range
hasResource	InventoryManagement	Resources
hasID_RecipeElement	RecipeElement	IDRecipeElement
hasParameter	Formula; RecipeElement	Parameter
hasProductionPolicy	Demand	ProductionPolicy
hasEquipmentRequirement	MasterRecipe; RecipeElement	EquipmentRequirement
hasID_Equipment	Unit	IDEquipment
hasRestriction	EquipmentRequirement	Restriction
hasID_EquipmentRequirement	EquipmentRequirement	IDEquipment
hasDemand	EndProduct	Demand
hasID_Logic	ProceduralLogic	IDLogic
hasLink_from	ProceduralLink	IDProcessStage
hasIDFromRecipeElementType	RecipeElement	IDRecipeElementType
receiveInformation FromIDRecipe ElementType	IDRecipeElementType	ProcessStage; RecipePhase; RecipeOperation
hasID_EquipmentModule	EquipmentModule	IDEquipmentModule
hasParameterSource	Parameter	MaterialResources; ProcessParameter
hasHeader	RecipeType; RecipeElement	Header
refersTo	Unit	EquipmentRequirement
hasLink_to	ProceduralLink	IDProcessStage
hasProcessOperation	ProcessStage	ProcessOperation
hasFormula	RecipeType	Formula
referedFrom	EquipmentRequirement	Unit
receiveInformationFromIDLink	IDLink	ProceduralLink
provideID	ProceduralLink	ID
hasTime	ProcessStage	P_Time
hasProceduralLogic	RecipeType; RecipeElement	ProceduralLogic
relatesToProduct	Demand	EndProduct
hasID_Material	Resources	IDMaterial
hasID_ProcessStage	ProcessStage	IDProcessStage
hasBatchSize	Header	BatchSize
hasProcessInputParameter	ProcessStage	P_Materials
hasLink	ProceduralLogic	ProceduralLink
hasProcessOutputParameter	ProcessStage	P_Materials
hasID_RecipeID	RecipeType	IDRecipe
hasRecipeElement	RecipeType; RecipeElement	RecipeElement

3.3. Check/study phase

3.3.1. Language conformity standard

Since the software for ontology construction and edition, Protègè, is based on the standards in accordance to the rules from the W3C, this step of the Check phase is automatically fulfilled.

3.3.2. Conceptually conformity standard

The ontology is based on the ANSI/ISA standards, as well as reviews from the literature widely recognized by the scientific community (Brandl & Emerson, 2003) in the process system engineering domain. For this reason, the conceptual conformity of the built project is guaranteed.

3.3.3. Reasoning

In general, the support for debugging defects in OWL ontologies has been fairly weak. Common defects include inconsistent ontologies and unsatisfiable concepts. An unsatisfiable concept is one that cannot possibly have any instance or it represents the empty set (e.g., owl:Nothing). However, these errors can be detected automatically using a Description Logics reasoner, which simply reports the errors, without explaining why the error occurs or how it can be resolved correctly.

The ontology proposed in this work has been checked using several reasoners included in Protègè namely FaCT++, HermiT, Pellet and Pellet incremental. The reasoners perform the identification of the domain and range of properties (relationships) among the classes that compose the ontology. As a result, they detect problems such as:

- Inverse properties must have inverse domains and ranges.
- Missing disjoints on primitive subclasses.
- The transitivity of a property should also hold for its inverse.

In general, there is a risk of inconsistencies related to unfeasible class and properties descriptions. For this reason, the taxonomic classification of the classes was checked, and the ontology consistency was validated by the use of reasoners. Only after attaining these two actions, the consistency of the asserted individuals could be also ensured by the reasoners.

Table 3
Data properties (domain & range).

Data property	Domain	Range
value	BatchSize; Inventory; Parameter; Restriction; Demand; ProductionOrder	float
link_type	ProceduralLink	string
max_value	BatchSize; ProcessParameter; Resources	float
min_value	BatchSize; ProcessParameter; Resources	float
salesPrice	Demand	float
availability	Unit	boolean
priceUnits	Demand	string
due_date	Demand; ProductionOrder	data
unit_of_measure	BatchSize; ProcessOutput; ProcessParameter; Restriction; ProductionOrder	string
isDemanded	MaterialResources	boolean

Master Recipe Class

Description	Expression
hasEquipmentRequirement ≥ 1	{MasterRecipe ∈ R | card({Area ∈ RULV : <MasterRecipe,Area> ∈ ER(hasArea)}) ≥ 1}
hasFormula = 1	{MasterRecipe ∈ R | card({Formula ∈ RULV : <MasterRecipe,Formula> ∈ ER(hasFormula)}) = 1}
hasHeader = 1	{MasterRecipe ∈ R | card({Header ∈ RULV : <MasterRecipe,Header> ∈ ER(hasHeader)}) = 1}
hasID_RecipeID = 1	{MasterRecipe ∈ R | card({RecipeID ∈ RULV : <MasterRecipe,RecipeID> ∈ ER(hasID_RecipeID)}) = 1}
hasProceduralLogic ≥ 1	{MasterRecipe ∈ R | card({ProceduralLogic ∈ RULV : <MasterRecipe,ProceduralLogic> ∈ ER(hasProceduralLogic)}) ≥ 1}
hasRecipeElement ≥ 1	{MasterRecipe ∈ R | card({RecipeElement ∈ RULV : <MasterRecipe,RecipeElement> ∈ ER(hasRecipeElement)}) ≥ 1}

Site Class

Description	Expression
hasArea ≥ 1	{Site ∈ R | card({Area ∈ RULV : <Site,Area> ∈ ER(hasArea)}) ≥ 1}
hasTransportationLink ≥ 1	{Site ∈ R | card({TransportationLink ∈ RULV : <Site,TransportationLink> ∈ ER(hasTransportationLink)}) ≥ 1}

Fig. 4. Restrictions related to the master recipe and site classes.

Along the project development, inconsistencies have been solved at the time they appeared. After finding out defects in the ontology, their resolution has been carried out, requiring an exploration of remedies with a cost/benefit analysis. In this case, repair solutions that impact the ontology minimally have been generated. Particular care and effort was taken to ensure that ontology repair is carried out efficiently. Table 4 contains the times spent by the reasoners for checking consistency once the ontology has been debugged.

3.3.4. Performance (application oriented)

In order to build the full informatics application in Java, the software NetBeans platform has been used for editing the Java code necessary to program the functions for the particular task that the ontology was designed for, according to the competence questions described in Section 3.1.3.

Table 4
Time spent by the reasoners for checking consistency.

Reasoner name	Time [sCPU]
Pellet	0.515
FaCT++	0.118
HermiT	3.609
Pellet (Incremental)	0.359

Thus the OWL API interface (Horridge, Bechhofer, & Noppens, 2007) has been adopted as a base code for its OWL APIs functionality. The OWL API is a Java interface and implementation for the W3C Web Ontology Language OWL. The latest version of the API is focused towards OWL 2 which encompasses, OWL-Lite, OWL-DL and some elements of OWL-Full. The OWL API is open source. The interface allows to access the ontology within a Java environment.

The results of running the informatics application may be saved in a data base for facilitating the access to them. These results could be used as inputs of any optimization tool used by the decision makers or potential user. Thus the results of the optimization are available to update and be included in the data instantiation of the problem in the ontological framework. This means that the ontology manages the knowledge and structures information quality, that is, where, when, how, who and to whom information must be delivered.

In order to perform the ontological modeling, three case studies from the literature have been used. Specifically, these three models are presented in Section 4 for facilitating the explanation of the development of the methodology, and have the following main features:

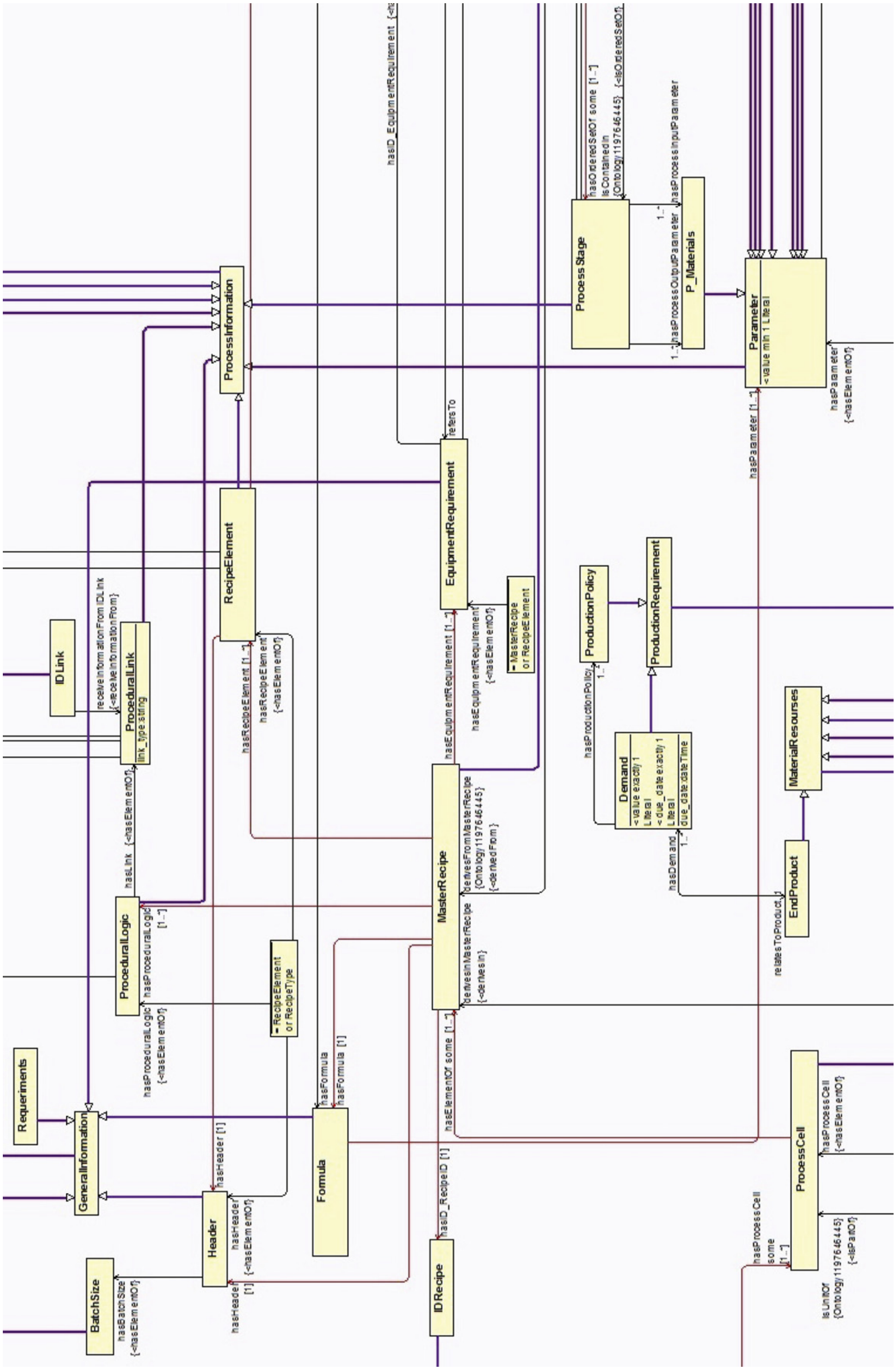

Fig. 5. UML diagram of key concepts of the ontology at operational level (master recipe, formula, header, recipe element, procedural logic, process information).

- Case study 1: Scheduling of multiproduct batch plant: it manufactures two products through three stages, adapted from Muñoz et al. (2011).
- Case study 2: Scheduling of multipurpose batch plant: it manufactures two products in five stages, adapted from Kondili, Pantelides, and Sargent (1993).
- Case study 3: Scheduling of complex batch plant: it brings together the features of the previous two case studies in one case study.

3.3.5. Revise

At this stage of the ontological framework design, it must have been tested in the previous section using trial tests which serve to check the performance of the framework. Therefore, the validation is based on the analysis of the behavior of the ontological framework in the domain, which is represented in the different tests. The conclusions of the previously performed case studies, in Sections 4.1.2, 4.2.2 and 4.3.1, serve to carry out the revision task of the ontology. In this way, the use of the ontology enables the integration of the different decision levels and supplies the process state information to different levels in the decision-making hierarchical structure.

A comparison between the process of how different levels interact in a traditional way and a proposed one has been done. A time and motion study has been used by the decision makers for the tasks of collecting data, optimizing production planning and its corresponding analysis of the results in order to take decisions (Muñoz et al., 2010). An average improvement of 20% in the number of steps for carrying out each process has been obtained, leading to savings in process time ranging between 70 and 80%.

3.4. Act phase

3.4.1. Implementation

- Performance (application environment): After the creation of the ontology, different applications have been performed in benchmark problems of the literature comprising the planning and scheduling domain. The behavior of the framework is thoroughly reported in the case studies given in Section 4.

  As a conclusion of this design step, it can be stated that the ontology is general enough to be used and reused in any scheduling optimization problem case, as it is demonstrated in Section 4.
- Distribution of the new ontology: A users manual contains in general terms the procedures for:
  (i) Instantiating a real problem.
  (ii) Checking consistency.
  (iii) Generating input files for the decision maker. This file can be used for machines (formal specification) or people(informal specification).

3.4.2. Maintenance

The maintenance phase is part of the continued use of this project. Hence, the formalization and documentation of possible changes are updated when improvements in the domain application are made.

3.5. Re-planning phase

The results of this phase are included in the previous subsections, which present the final structure of the pursuit project. However the project has been re-planned several times adding new functionalities to the studied domain. The main stages are:

(1) Construction oriented to the control and scheduling levels.
(2) Addition of the detailed planning level (site and master recipe issues).

The whole final semantic model can be inspected and downloaded in http://cepima.upc.edu/papers/ontological_model.

4. Scheduling case studies

In this section, the application of the ontological framework is demonstrated in several plant configurations related to different case studies. The contributions of the proposed framework (Fig. 6)

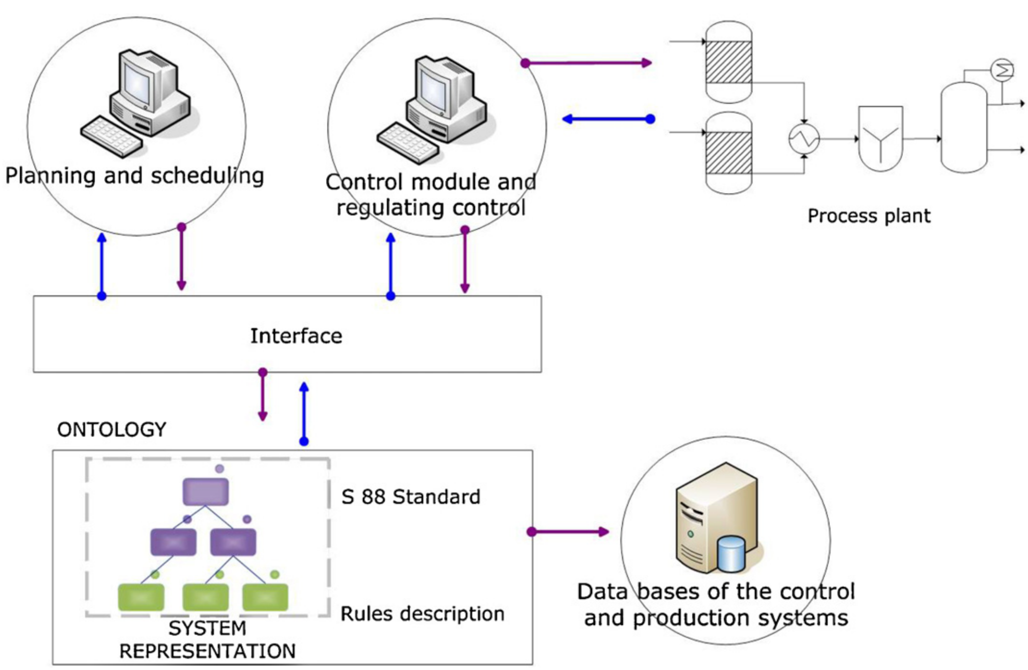

Fig. 6. General application environment.

Table 5
Product prices and lot sizes for Case study 1.

Product	Batch benefit [m.u./batch]	Batch size [ton/batch]	Demand [ton]	Unitary energy cost [m.u./MWh]
A	30	5	20	90
B	40	6	24	90

Table 6
Recipe stage times for Case study 1 [h].

Product	Stage 1		Stage 2		Stage 3	
	Unit	Time [h]	Unit	Time [h]	Unit	Time [h]
A	R1	0.5	P1	0.5	C1	0.5
B	R1	0.5	P1	0.8	C1	0.4

to the scheduling decision-making are evaluated, emphasizing the following features: re-usability, usability and more efficient communication.

The re-usability consists of the instantiation of different problems using the same modeling framework. The specific information regarding each case study is developed in every instantiation description.

The usability refers to the capacity of transforming the data contained in the problem instantiated in the model into valuable information (information quality). The steps comprised in the usability given in this ontological model are common to all case studies. Specifically, once the problem is instantiated, the scheduling function requires the following information:

- Capacity: It refers to the raw materials quantities available in the production plant, and the maximum storage capacity of the intermediates, residues and final products.
- Demand: It contains the quantity of each product required by the customers.
- Due-date: It contains the date that each final product has to be delivered to the customer.
- Product-stage-unit: It contains the set of units that is available at each stage of the production process for all products.
- Quantities in/out: It refers to the quantities of the raw materials, intermediates and final products that are necessary to perform/that are delivered at each stage of the production recipe of each product according to the material balances.
- Processing time: The time necessary to perform the tasks of a given stage.
- Stage-process: It contains the set of stages that are contained in each master recipe of the products.
- Time horizon: It refers to the time horizon for the scheduling task.

Therefore, as mentioned in Section 3.3.4, Java code is programmed for generating the input files for the scheduling optimization tools. Fig. 7 contains the sources from the ontological model that are used to provide the optimization framework with the necessary data. Finally, the optimization software can be executed. On the whole, any scheduling problem can be modeled by

Concept	Set	Ontology
Recipe	p	∀ MasterRecipe → RecipeID
Stages	s	∀ RecipeElement | RecipeElementType = ProcessStage → RecipeElementID
Units	u	∀ EquipmentRequirement ∈ GeneralInformation → EquipmentRequirementID
Materials	i	∀ RawMaterial ∈ ProcessInputs → MaterialsID ∀ Intermediate ∈ ProcessInputs → MaterialsID ∀ By_Product ∈ ProcessOutputs → MaterialsID ∀ End_Product ∈ ProcessOutputs → MaterialsID ∀ Residue ∈ ProcessOutputs → MaterialsID
Stage-process	p,s	∀ ProcessStage → ProcessStageID
Raw materials	i	∀ RecipeElemnt ∈ MasterRecipeID → RecipeElementID ∀ RawMaterial ∈ ProcessInput → RawMaterialID
Final products	i	∀ FinalProduct ∈ ProcessOutput → MaterialID
Product-Stage-Unit	p,s,u	∀ EquipmentRequirement ∈ ∀ RecipeElement ∈ MasterRecipe → EquipmentRequirementID
Process-stage-input	p,s,i	∀ ProcessStage ∈ RecipeElement ∈ MasterRecipe → ProcessInputParameter = ParameterSource → ID
Process-stage-output	p,s,i	∀ ProcessStage ∈ RecipeElement ∈ MasterRecipe → ProcessOutputParameter = ParameterSource → ID
Quantity-process-stage-input	p,s,i	∀ Material ∈ ProcessStage ∈ RecipeElement ∈ MasterRecipe → ProcessInputParameter = ParameterSource → Value
Quantity-process-stage-output	p,s,i	∀ Material ∈ ProcessStage ∈ RecipeElement ∈ MasterRecipe → ProcessOutputParameter = ParameterSource → Value
Processing time	p,s,u	Data Base for $\forall u; \forall s : \forall p$
Storage capacity	i	Data Base for ∀ Material
Demand	i	Data Base for ∀ FinalProduct(i)

Fig. 7. Relationship between optimization inputs and classes of the ontology.

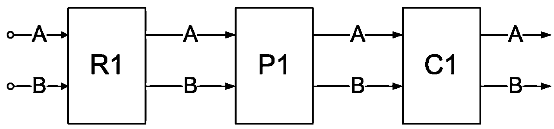

Fig. 8. Plant flowsheet for the Case study 1.

the ontological framework, and next solved by the decision maker tools.

In this work, the scheduling problem is solved using the STN-formulation (Maravelias & Grossmann, 2003) since it is able to represent multiproduct and multipurpose plant configurations. The mathematical model has been implemented in GAMS and solved using the MILP solver CPLEX 9.0.

4.1. Case study 1: multiproduct batch plant

4.1.1. Problem description

The case study consists of a multiproduct plant, manufacturing two products, i.e. A and B, through three stages (Fig. 8) adapted from Muñoz et al. (2011). Product batch sizes, optimal production times and product demands are given in Tables 5 and 6. A single unit is available for each stage and unlimited intermediate storage policy is adopted. Product changeover time and cost are disregarded. A production time horizon of 6.5 h is considered.

The scheduling objective function, profit maximization includes the benefit of each batch and the operating cost.

4.1.2. Results

The re-usability of the proposed ontological model has been demonstrated through the instantiation of this case study. Specifically a total of 151 instances represent all the necessary problem features. The reasoner needs 0.485 sCPU to check the consistency of the inferred instances. In this instantiation of physical and procedural model, equipment requirement, processing times, costs, mass balances, raw material, and others elements have been referenced from the master recipe. For illustrating purposes, Fig. 9 contains the instances derived from the master recipe of product A.

Table 7 contains all the instances of this case study for each class. Therefore, with the instances defined in Table 7 and the properties within the model, the scheduling problem can be fulfilled and optimized.

Regarding the re-usability, results shown in Gantt chart in Fig. 10 are obtained after the problem instantiation.

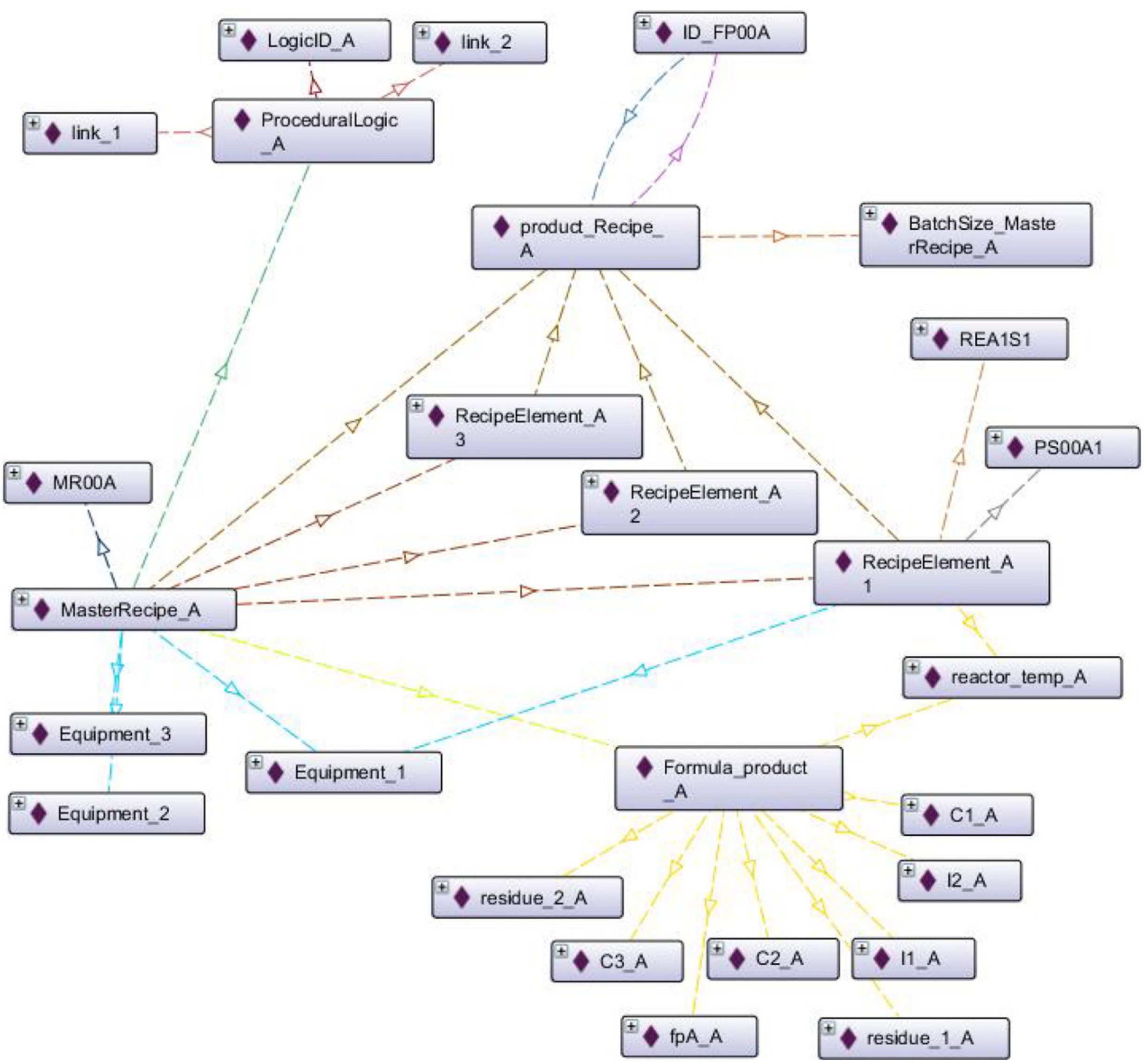

Fig. 9. Instances derived from master recipe of product A in Case study 1.

Table 7
Instances of each class for Case study 1.

Class	Instances
BatchProcess	Separation_liquid;
BatchSize	BatchSize_MasterRecipe_A; BatchSize_MasterRecipe_B;
ControlModule	ControlModule_ElectroValves; ControlModule_Pumping; ControlModule_Temperatur; ControlModule_none;
Demand	Demand_A_May; Demand_B_May;
EndProduct	EndProduct_A; EndProduct_B;
EnergeticResourses	electricity;
EquipmentModule	ElectroValve1; ElectroValve2; ElectroValve3; ElectroValve4; ElectroValve5; ElectroValve6; ElectroValve7; ElectroValve8; Pump1; Pump2; Pump3; Resistence;
EquipmentRequirement	Equipment_1; Equipment_2; Equipment_3;
Formula	Formula_Product_B; Formula_product_A;
Header	product_Recipe_A; product_Recipe_B;
IDEndProduct	ID_FP00A; ID_FP00B;
IDEquipment	ID_D-120; ID_D-130; ID_P-110;
IDEquipmentModule	ID_J-112; ID_J-113; ID_J-122; ID_J-123; ID_J-124; ID_J-132; ID_J-133; ID_J-134; ID_K-111; ID_K-112; ID_K-113; ID_R-110;
IDIntermediate	IN001A; IN001B; IN002A; IN002B;
IDLink	ID_LA1; ID_LA2; ID_LB1; ID_LB2;
IDLogic	LogicID_A; LogicID_B;
IDProcessStage	PS00A1; PS00A2; PS00A3; PS00B1; PS00B2; PS00B3;
IDRawMaterial	ID_RM001; ID_RM002; ID_RM003;
IDRecipe	MR00A; MR00B;
IDRecipeElement	REA1S1; REA1S2; REA1S3; REB1S1; REB1S2; REB1S3;
IDResidue	ID_R001; ID_R002;
Intermediate	Intermediate_1A; Intermediate_1B; Intermediate_2A; Intermediate_2B;
InventoryMaterialResources	InventoryMaterialResources_C1; InventoryMaterialResources_C2; InventoryMaterialResources_C3; InventoryMaterialResources_FPA; InventoryMaterialResources_FPB;
MasterRecipe	MasterRecipe_A; MasterRecipe_B;
P_Materials	C1_A; C1_B; C2_A; C2_B; C3_A; C3_B; I1_A; I1_B; I2_A; I2_B; fpA_A; fpB_B; residue_1_A; residue_1_B; residue_2_A; residue_2_B;
P_Temperature	reactor_temp_A; reactor_temp_B;
P_Time	PT_A_S1; PT_A_S2; PT_A_S3; PT_B_S1; PT_B_S2; PT_B_S3;
ProceduralLink	link_1; link_2; link_3; link_4;
ProceduralLogic	ProceduralLogic_A; ProceduralLogic_B;
ProcessingTime	ProcessingTime_S1; ProcessingTime_S2; ProcessingTime_S3;
ProcessOperation	Reaction; Separation;
ProcessStage	ProcessStage_A1; ProcessStage_A2; ProcessStage_A3; ProcessStage_B1; ProcessStage_B2; ProcessStage_B3;
ProductionPolicy	Earliness; EnvironmentalImpact; Makespan; Profit; Tardiness;
RawMaterial	RawMaterial_A; RawMaterial_B; RawMaterial_C;
RecipeElement	RecipeElement_A1; RecipeElement_A2; RecipeElement_A3; RecipeElement_B1; RecipeElement_B2; RecipeElement_B3;
Residue	Residue_1; Residue_2;
Temperature	reactor_temp;
Unit	Reactor; Separator_1; Separator_2;

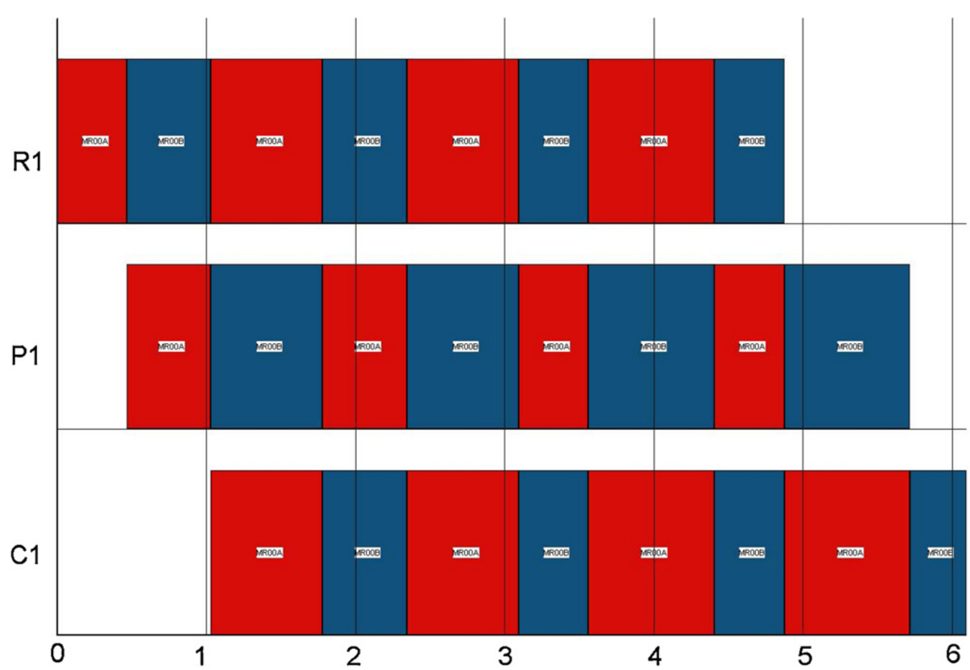

Fig. 10. Gantt chart resulting from the optimization of Case study 1.

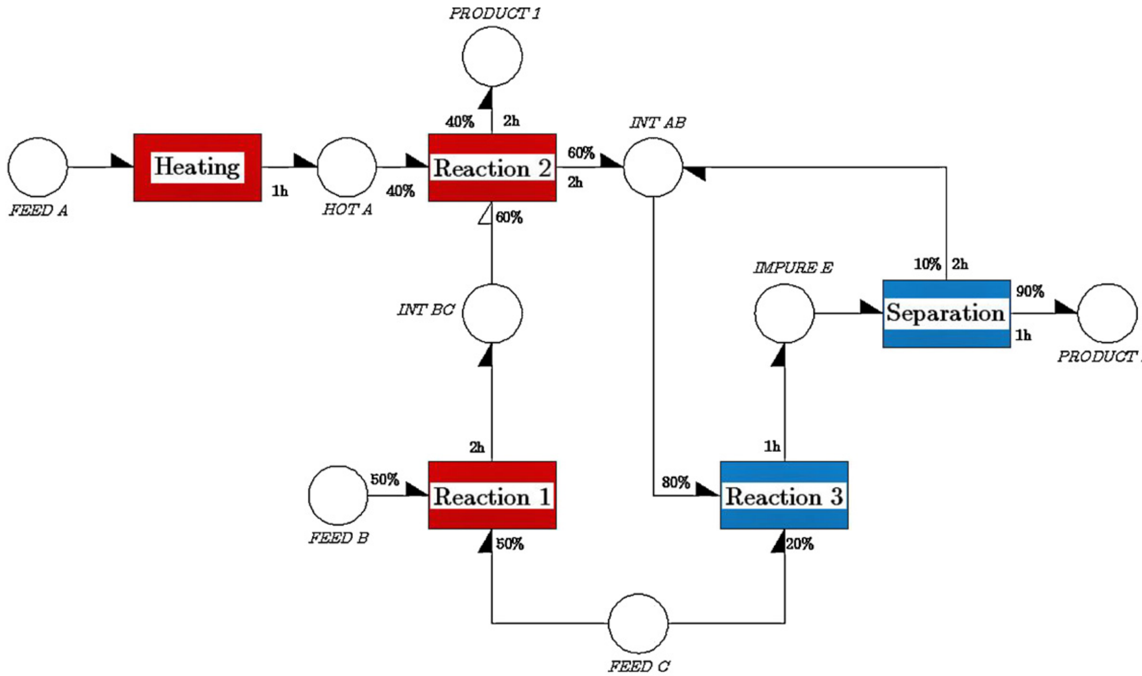

Fig. 11. STN representation for the Case study 2.

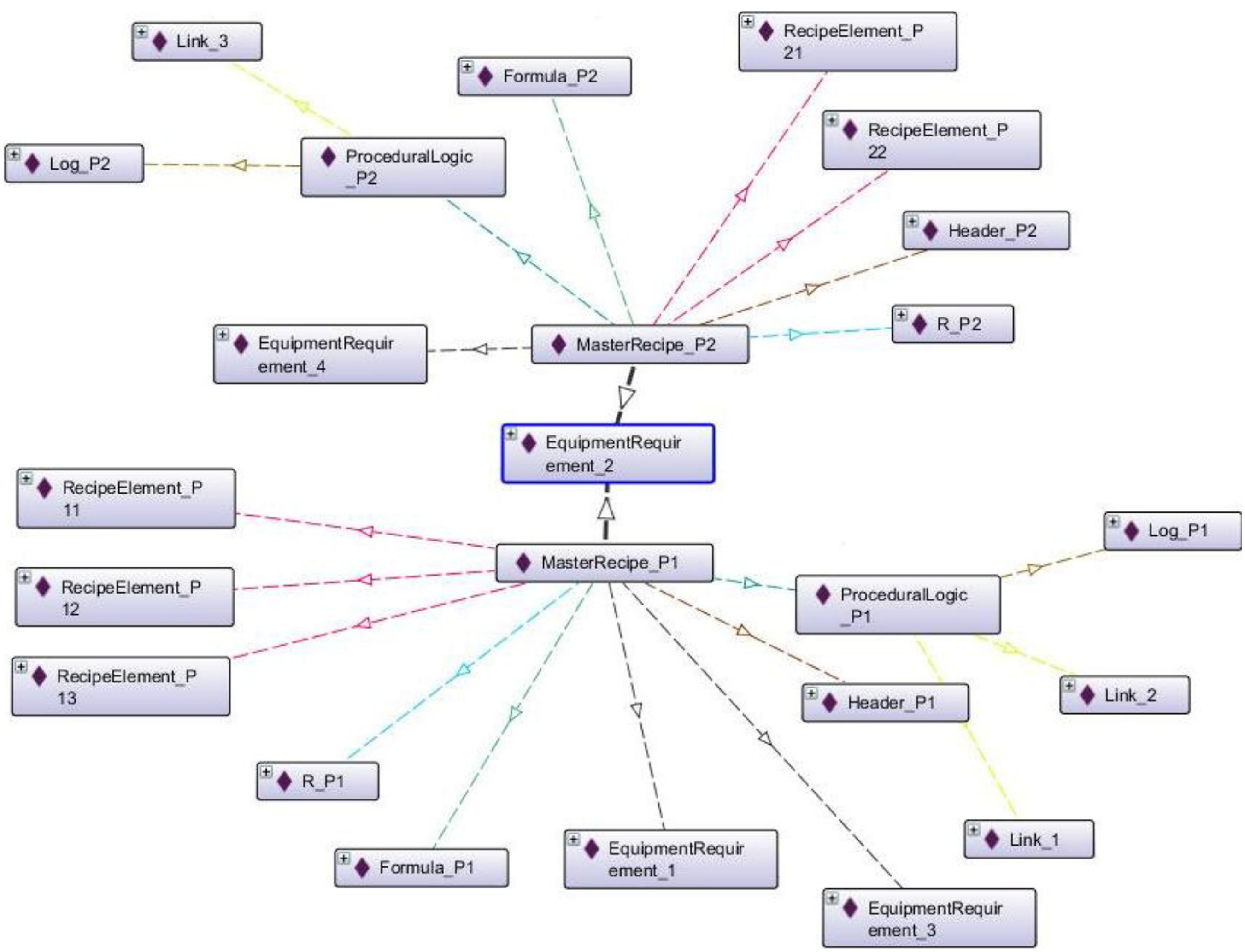

Fig. 12. Instances derived from the master recipes in Case study 2.

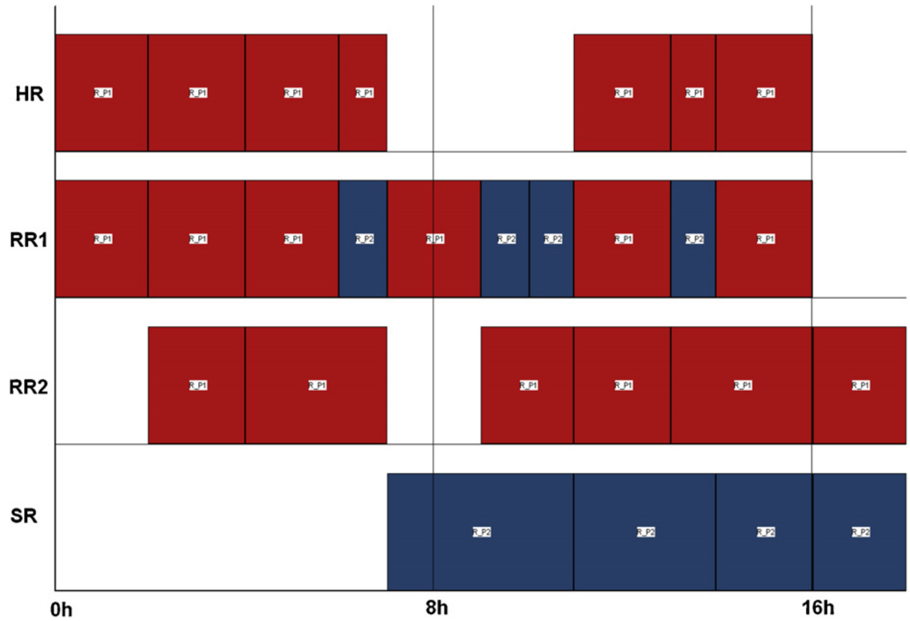

Fig. 13. Gantt chart resulting from the optimization of Case study 2.

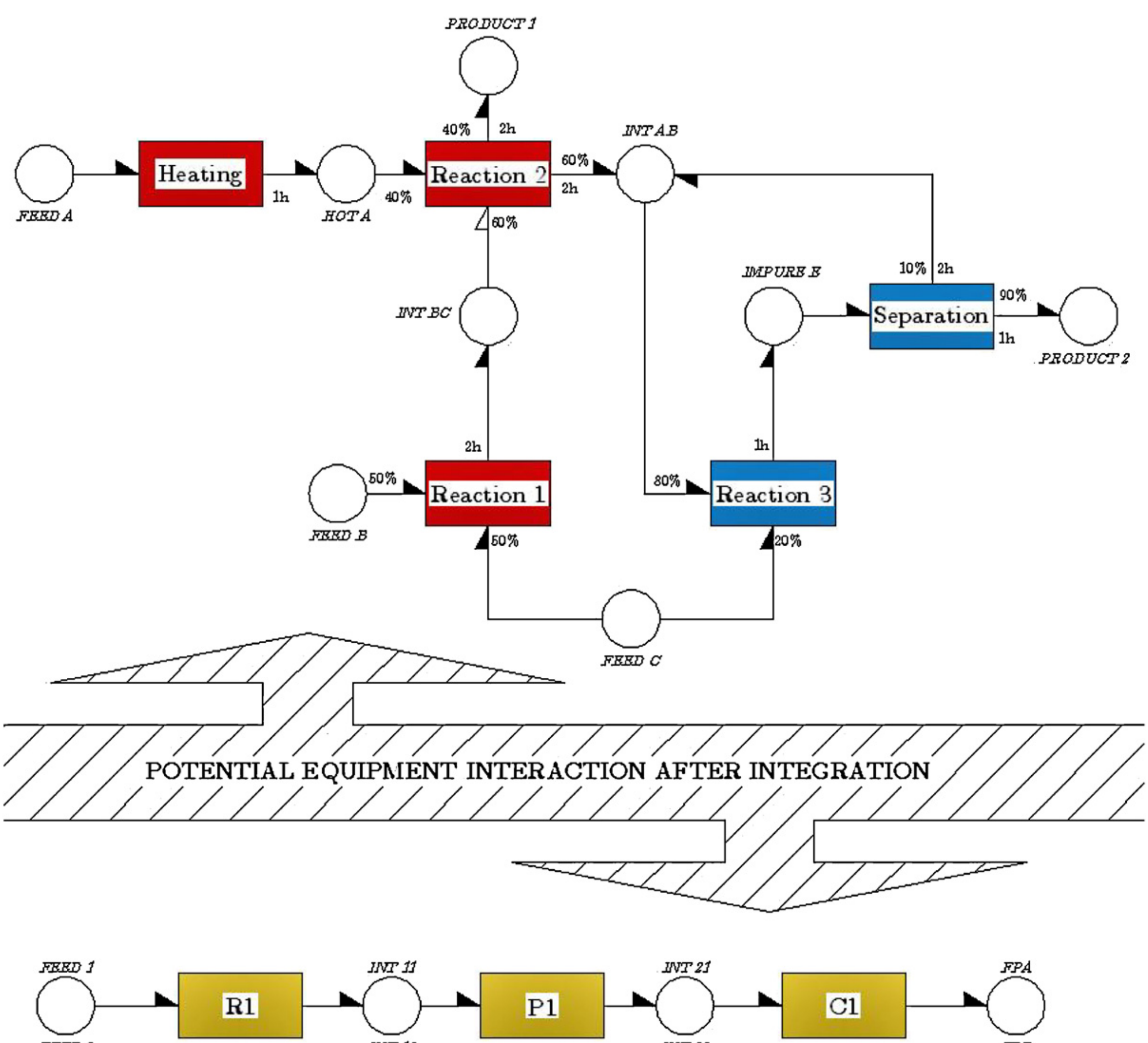

Fig. 14. STN representation showing the potential integration of the two production lines for Case study 3.

Table 8
Instances of each class for Case study 2.

Class	Instances
BatchSize	BatchSize_P1; BatchSize_P2;
ControlModule	HeatingReactorControlModule; ReactorControlModule; SeparationControlModule;
Demand	DemandProduct1; DemandProduct2;
EndProduct	Product1; Product2;
EquipmentModule	HeatingReactorModule; ReactorModule; SeparationModule;
EquipmentRequirement	EquipmentRequirement_1; EquipmentRequirement_2; EquipmentRequirement_3; EquipmentRequirement_4;
Formula	Formula_P1; Formula_P2;
Header	Header_P1; Header_P2;
IDEndProduct	FP001; FP002;
IDEquipment	HR; RR1; RR2; SR;
IDIntermediate	IN001; IN002; IN003; IN004;
IDLink	IDLink_1; IDLink_2; IDLink_3;
IDLogic	Log_P1; Log_P2;
IDProcessStage	PS_P1_H; PS_P1_R1; PS_P1_R2; PS_P2_R1; PS_P2_S;
IDRawMaterial	RM001; RM002; RM003;
IDRecipe	R_P1; R_P2;
IDRecipeElement	RE_P1_S1; RE_P1_S2; RE_P1_S3; RE_P2_S1; RE_P2_S2;
Intermediate	HotA; ImpureE; IntAB; IntBC;
InventoryMaterialResources	Inv_A; Inv_B; Inv_C;
MasterRecipe	MasterRecipe_P1; MasterRecipe_P2;
P_Materials	AB_P1; AB_P2_In; AB_P2_Out; A_P1; BC_P1; B_P1; C_P1; C_P2; E_P2; HA_P1; P1; P2;
P_Time	P_Time_Heating_P1; P_Time_Reaction1_P1; P_Time_Reaction1_P2; P_Time_Reaction2_p1; P_Time_Separation_P2;
ProceduralLink	Link_1; Link_2; Link_3;
ProceduralLogic	ProceduralLogic_P1; ProceduralLogic_P2;
ProcessingTime	ProcessingTime_Heating; ProcessingTime_Reaction1; ProcessingTime_Reaction2; ProcessingTime_Reaction3; ProcessingTime_Separation;
ProcessOperation	Heating; Reaction; Separation;
ProcessStage	ProcessStage_P1_Reaction1; ProcessStage_P1_Reaction2; ProcessStage_P1_heating; ProcessStage_P2_Reaction; ProcessStage_P2_Separation;
ProductionPolicy	Earliness; EnvironmentalImpact; Makespan; Profit; Tardiness;
RawMaterial	FeedA; FeedB; FeedC;
RecipeElement	RecipeElement_P11; RecipeElement_P12; RecipeElement_P13; RecipeElement_P21; RecipeElement_P22;
Unit	HeatingReactor; Reactor1; Reactor2; Separation_Unit;

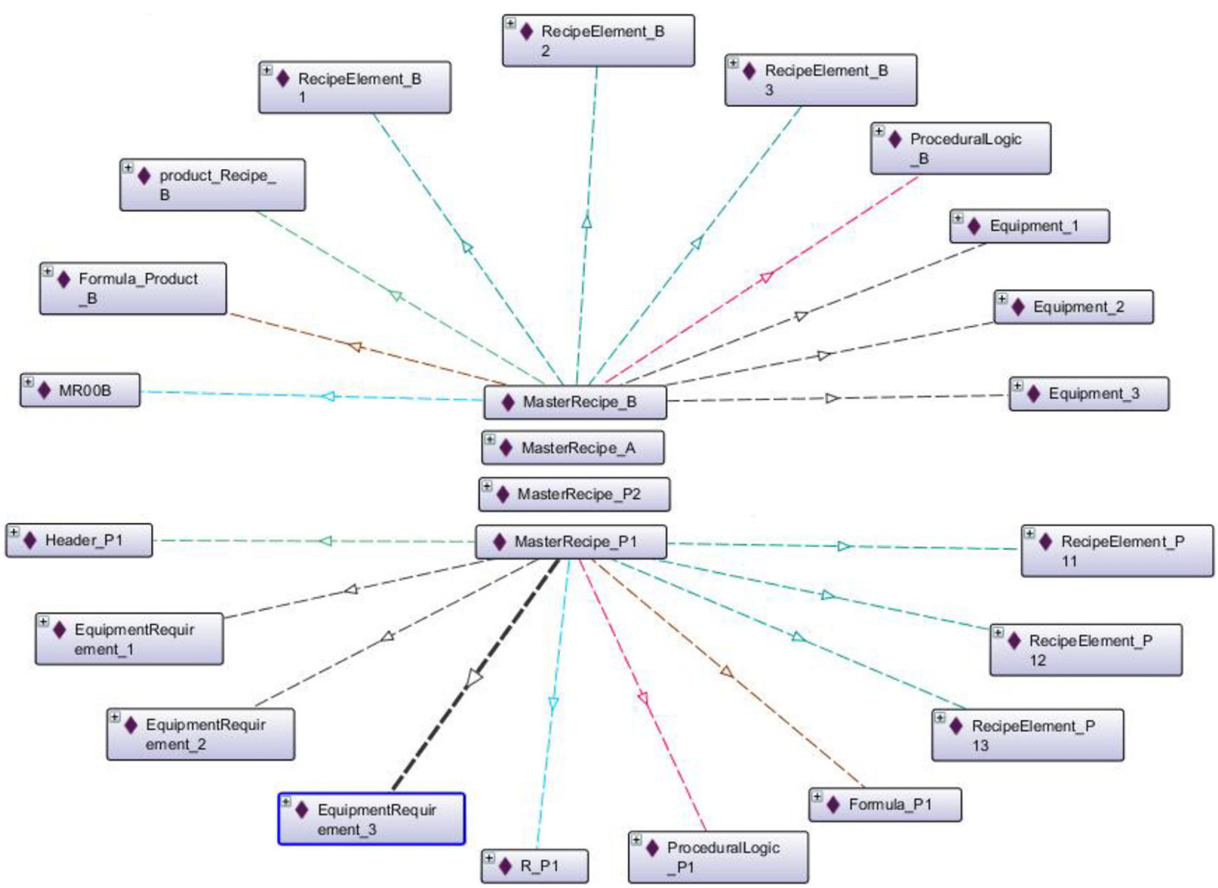

Fig. 15. Instances derived from the master recipes in Case study 3.

Fig. 16. Gantt chart resulting from the optimization of Case study 3.

4.2. Case study 2: multipurpose batch plant

4.2.1. Problem description

The case study consists of a multi-purpose batch plant, which is a benchmark case study of the scheduling area (Kondili et al., 1993). Fig. 11 shows the process flowsheet as well as the processing times and mass balances.

A single unit is available for each stage and unlimited intermediate storage policy is adopted. Product changeover time and cost are disregarded.

The scheduling objective function consists of production maximization within 18 h time horizon.

4.2.2. Results

The re-usability of the proposed ontological model has been proved through the instantiation of the multi-purpose batch plant. Specifically a total of 111 instances represent all the necessary problem features. The reasoner needs 0.172 sCPU to check the consistency of the inferred instances. In this case, the production process has been divided in two main recipes. For illustrating purposes, Fig. 12 contains the instances derived from the master recipes.

Table 8 contains all the instances of this case study for each class. As defined before the instances of specific classes are required by the scheduling function in order to implement the scheduling optimization. Therefore, with the instances defined in Table 8 and the properties within the model, the scheduling problem can be fulfilled and optimized.

Regarding the re-usability, results shown in the Gantt chart in Fig. 13 are obtained after the problem instantiation.

4.3. Case study 3: multipurpose batch plant

This case study comprises the model integration of the two previous production plants. Specifically, the re-usability of the ontology has allowed to directly instantiate this case study based on the two previous. Even more, in this case study different features regarding plant configuration (equipment, storage capacity, raw materials) can be turned off or on as convenient. Therefore, Fig. 14 presents the two STN representations corresponding to the aforementioned production lines. In this case equipment resources are not shared. However, equipments could be used by any new recipe if equipment integration was pursued.

4.3.1. Results

The re-usability of the proposed ontological model has been proved by the instantiation of the two production lines. Specifically a total of 244 instances represent all the necessary problem features. The reasoner needs 1.42 sCPU to check the consistency of the inferred instances. For illustrating purposes, Fig. 15 contains the instances derived from the master recipes.

Regarding the re-usability, results shown in Gantt chart in Fig. 16 are obtained after the problem instantiation.

5. Conclusions

This work contributes to improve communication within the plant process environment, and represents a step forward to support the integration of different software tools applicable to the management and exploitation of plant database information, resulting into an enhancement of the entire scheduling function. Specifically, once the problem is instantiated, the modification of processing times, product demand or due date, resulting from plant redesign or new orders, is straight-forward and can be traced back to adequate databases.

Thus, this ontology enhances the way for achieving a successful scheduling decision-making supporting tool which adapts and recognizes the different elements found in the master recipe. Moreover, a general semantic framework is proposed, which is able to model any scheduling plant layout, proving its re-usability. Furthermore, it has been proved the ontology usability by its application to an optimization framework. As a whole, the main contributions of this environment and the model behind are re-usability, usability, higher efficiency in communication and coordination procedures.

In addition, it has been proved the adequacy of an ontology as a means for sharing information about a general model for different problem representations. As a result, it solves the problem of integration, standardization and compatibility of heterogeneous modeling systems. Even more, the response time for decision-making task could be reduced and better decisions adopted owing to faster availability of higher quality data and the improved visibility of the existing relationships between the scheduling function and other hierarchical levels functions.

Further work is underway to unveil the full potential to implement a large-scale semantic web approach to the overall enterprise processes decisions.

Acknowledgements

Financial support received from CIMAT México is fully acknowledged. In addition, financial support received from the Spanish government and the European Community (FEDER) (DPI2006-05673; DPI2009-09386) is also fully appreciated.

References

Biomedical Informatics Research (BMIR), S. C. (2007). *Protègè 3.4 and protègè 4.1*. http://protege.stanford.edu.

Brandl, D., & Emerson, D. (2003, September). *Batch markup language batchml*.

Grossmann, I. (2005). Enterprise-wide optimization: A new frontier in process systems engineering. *Aiche Journal*, *51*(7), 1846–1857.

Grossmann, I., Erdirik-Dogan, M., & Karuppiah, R. (2008). Overview of planning and scheduling for enterprise-wide optimization of process industries. *At-Automatisierungstechnik*, *56*(2), 64–79.

Gruber, T. (2008). *Ontology*. Ling Liu and M. Tamer Özsu. http://tomgruber.org/writing/ontology-definition-2007.htm.

Harjunkoski, I., Nystrom, R., & Horch, A. (2009). Integration of scheduling and control-theory or practice? *Computers & Chemical Engineering*, *33*(12), 1909–1918.

Horridge, M., Bechhofer, S., & Noppens, O. (2007). Igniting the owl 1.1 touch paper: The owl api. In *OWLED 2007, 3rd OWL experienced and directions workshop*

Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2007). A practical guide to building owl ontologies using Protègè 4 and CO-ODE tools. *Tech. rep*. The University of Manchester.

International Society for Measurement and Control. (1995, October). *Batch control. Part 1. Models and terminology*.

International Society for Measurement and Control. (2001). *Batch control. Part 2. Data structures and guidelines for languages*. International Standard, ISA Society.

International Society for Measurement and Control. (2006, August). *Batch Control. Part 4. Batch Production Records*.

International Society for Measurement and Control. (2007, January). *Batch control. Part 5. Automated equipment control models & terminology*.

Kallrath, J. (2005). Solving planning and design problems in the process industry using mixed integer and global optimization. *Annals of Operations Research*, *140*(1), 339–373.

Klatt, K. U., & Marquardt, W. (2009). Perspectives for process systems engineering-personal views from academia and industry. *Computers & Chemical Engineering*, *33*(3), 536–550.

Klein, M., Fensel, D., Kiryakov, A., & Ognyanov, D. (2002). Ontology versioning and change detection on the web. In *13th international conference on knowledge engineering and knowledge management (EKAW02)* (pp. 197–212).

Kondili, E., Pantelides, C., & Sargent, R. (1993). A general algorithm for short-term scheduling of batch operations. I. Milp formulation. *Computers & Chemical Engineering*, *17*(2), 211–227. http://www.sciencedirect.com/science/article/pii/009813549380015F

Maravelias, C. T., & Grossmann, I. E. (2003). New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Industrial & Engineering Chemistry Research*, *42*, 3056–3074.

Maravelias, C. T., & Sung, C. (2009). Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering*, *33*(12), 1919–1930.

Morbach, J., Wiesner, A., & Marquardt, W. (2009). Ontocape 2.0 a (re)usable ontology for computer-aided process engineering. *Computers & Chemical Engineering*, *33*, 1546–1556.

Muñoz, E., Capon-Garcia, E., Moreno-Benito, M., Espuna, A., & Puigjaner, L. (2011). Scheduling and control decision-making under an integrated information environment. *Computers & Chemical Engineering*, *35*, 774–786.

Muñoz, E., Espuña, A., & Puigjaner, L. (2010). Towards an ontological infrastructure for chemical batch process management. *Computers & Chemical Engineering*, *34*(5), 668–682, selected paper of symposium ESCAPE 19, Krakow, Poland, June 14–17, 2009.

Shirasuna, M. (2007). Optimización de la producción en una empresa de cervezas usando el estándar isa s88 (caso polar). In *Rockwell automatization* June.

Venkatasubramanian, V., Zhao, C., Joglekar, G., Jain, A., Hailemariam, L., Suresh, P., et al. (2006). Ontological informatics infrastructure for pharmaceutical product development and manufacturing. *Computers & Chemical Engineering*, *30*(July), 1482–1496.