Using ontologies to capture and structure knowledge about disruptions in manufacturing systems: An immune driven approach

Saber DARMOUL  
Blaise Pascal University  
LIMOS, UMR CNRS 6158  
Clermont Ferrand, FRANCE  
saber.darmoul@ifma.fr

Henri PIERREVAL  
IFMA  
LIMOS, UMR CNRS 6158  
Clermont Ferrand, FRANCE  
henri.pierreval@ifma.fr

Sonia HAJRI – GABOUJ  
INSAT  
URAI  
Tunis, TUNISIA  
sonia.gabouj@insat.rnu.tn

Abstract — Artificial immune systems have recently been identified as a promising approach to assist decision makers in managing occurrences of unexpected disruptions in manufacturing systems. However, the design of such systems requires capturing and structuring knowledge related to disruptions. Such knowledge consists of manufacturing system specificities, control principles, disruption features, etc. Based on biological immune concepts, we design a generic ontology framework to capture this knowledge. This framework can be used as an initial basis to incorporate more specific disruption knowledge models in specific domain applications. An ontology implementation is proposed using the OWL language and Protégé ontology editor.

Keywords: manufacturing, disruption management, biologically inspired modelling, ontology, biological immune system.

I. INTRODUCTION

One of the most important challenges in the field of production monitoring and control is to manage unexpected disruptions such as late deliveries, machine failures, tool breakage, rush orders, etc. Unless some strategy to deal with a disruption is implemented, the production system is forced to stop or at least to deviate from its expected operation and performance [5], thus causing severe consequences [2] such as extra costs and delays, lack of customer confidence, loss of market shares, etc.

Implementing disruption management policies in manufacturing environments requires extensive knowledge of many different aspects: (1) knowledge of the production system features, characteristics and specificities (2) knowledge of unexpected events and situations that can occur and lead to disruptions and problems (3) knowledge of the direct and potential impact of these disruptions (4) knowledge of the methods, tools and techniques that can be used to find decisions to react to a disruption and its consequences and (5) knowledge of the strategies that enable implementing these decisions in manufacturing environments.

Different experts, ranging from production system operators to engineers, can contribute to provide such knowledge. Several industrial tools (such as Enterprise Resource Planning ERP, and Manufacturing Execution Systems MES [20]) and knowledge based approaches (including maintenance [18], quality management [12], production planning, scheduling [17], monitoring and control [21]) were proposed to assist decision makers during the disruption management process. They both use and generate disruption related data and knowledge in order to support the decision making process.

Unfortunately, both production system experts and domain specific software applications (such as computer aided maintenance, computer aided quality management, etc.) still lack a tool that enables sharing common understanding of the structure of information related to production system disruptions. This is mainly due to the apparent divergence of concerns and interests between these actors. The lack of such a tool disables multiple application access to, and profitable reuse of such valuable disruption knowledge. This is mainly due to interoperability issues. Existing decision support systems emphasize disruption handling capabilities, and put little focus on formalizing and capturing the knowledge generated during the process of disruption management, which prevents storage and further exploitation of this knowledge.

Although ontologies have been proposed to overcome similar limitations in supply chain and enterprise information flow integration [14, 19], we are not aware of works that applied ontologies to capture disruption knowledge in production systems. Ontology practitioners also emphasize that there is no single “correct” way or methodology for developing ontologies.

In this context, the biological immunity offers a very promising framework to monitor and control unexpected disruptions that can occur in a production system [8]. The Biological Immune System is structured in such a way to detect and react to harmful and previously unseen disease causing elements. Thanks to the immune memory, it can remember previous occurrences of such harmful threats. This knowledge is reused to accelerate the reaction to renewed occurrences of similar invaders.

In this paper, we rely on biological immunity to propose an ontology framework to capture and structure the knowledge generated during the process of disruption management in production systems. This framework is
A biological immune system (BIS) defends its host organism against disease causing elements, called pathogens, in a way that minimizes harm to the body and ensures its continued functioning [11]. Therefore, the BIS relies on two interrelated subsystems, known as the innate immune system and the adaptive immune system. Both systems involve immune cells that have surface receptors that discriminate what belongs to the body, also called Self, from what is foreign to the body, also called Non Self.

The innate immune system involves cells such as Antigen Presenting Cells (APCs) that insure broad detection of groups of pathogens. This broad detection is achieved using Pathogen Recognition Receptors (PRRs) that are molecules present at the surface of APCs and that allow identification of different classes of pathogens. These PRR identify and bind to Pathogen Associated Molecular Patterns (PAMPs) that are molecules present at the surface of pathogens and are different from the molecules of the body. To a first approximation, a PAMP can be considered as a collection of markers, called antigens, which are specific to each pathogen. Upon detection, the innate immune system cells initiate and guide an immune response to pathogens by controlling the activation of the cells of the adaptive immune system.

The adaptive immune system involves specialized immune cells, such as B, T helper (Th) and memory cells, which can perform more specific adaptation to antigens. When a pathogen invades the body, it releases its antigens, which may damage the body cells and/or change their normal behaviour. According to the danger theory [1], a damaged body cell sends out danger signals, which establish a danger zone around the damaged cell (cf. Figure 1) and cause large numbers of circulating immune cells to be stimulated and recruited to the site of infection [11].

Among the stimulated immune cells, B cells are both sensitive to antigens and to danger signals sent out by damaged cells [1]. B cells have receptors at their surface that can match (attach to) a specific antigen. B cells that do not match antigens within a danger zone, or are too far away from a danger zone do not get stimulated and die (cf. Figure 1). Those B cells that match antigens within a danger zone get stimulated to participate in an immune response to eliminate the pathogen and its antigen.

Figure 1. Danger Theory [1].

In case of stimulation of several B cells, their activation and involvement in the immune response is coordinated by specialized immune cells, called T helper (Th) cells, in such a way to prevent anarchy and waste of resources. Therefore, stimulated B cells capture antigen, fragment it into pieces, associate those pieces to molecules called Major Histocompatibility Complex (MHC), and present those MHC molecules to Th cells, which activate the B cells according to the intensity of their match (also called affinity) to the antigen. The so activated B cells proliferate and differentiate into antibody secreting cells. Antibodies are soluble forms of the B cell surface receptors which are able to specifically attach to antigens, block them and facilitate their elimination. Some of the B and T helper cells will differentiate into memory cells. These are retained in the body for long periods of time, thus guaranteeing future protection against pathogens which are structurally similar or identical to those that elicited the immune response [11].

Biological immunity inspired the development of artificial immune systems (AIS) to address several problems related to manufacturing systems, such as anomaly detection [10], fault diagnosis [4], production planning and scheduling [6]. Yet, to the best of our knowledge [8], biological immunity has not yet been applied to address the issue of disruption knowledge representation and management. Therefore, we propose to use ontologies to design a framework to capture the knowledge generated during the process of disruption management in production systems. The next section introduces ontologies as a knowledge representation tool, which we will use to design our framework.

III. ONTOLOGIES

In the context of artificial intelligence, an ontology defines a common vocabulary for users, human experts as well as software applications, who need to share information in a domain [3]. It includes machine interpretable definitions of basic concepts in the domain and relations among them. An ontology can be defined as a formal representation of:
1. A set of concepts (also called classes) within a domain of discourse. A class can have subclasses that represent concepts that are more specific than the superclass.
2. Properties (also called roles or slots) of each concept, describing various features and attributes of the concept.
3. Restrictions (also called facets or role restrictions) on properties.

An ontology, together with a set of individual instances of classes, constitutes a knowledge base. In practical terms, developing an ontology includes:

1. Defining classes in the ontology
2. Arranging the classes in a taxonomic hierarchy (superclass – subclass inheritance diagrams)
3. Defining properties and describing allowed values for these properties
4. Filling in the values of properties for instances

Building domain ontologies is useful to many “users”, such as problem solving methods, domain independent applications, and software agents, which can use ontologies, and knowledge bases built from ontologies, as data. Many disciplines, such as semantic web [13] and knowledge based systems [7] now develop standardized ontologies to analyze domain knowledge and to enable reuse of this knowledge. However, to the best of our knowledge, no attempt has been made to formally represent and capture production systems disruption knowledge using ontologies. In the next section, we rely on biological immunity as a methodological guideline to design an ontology framework to capture disruption management knowledge.

IV. AN ONTOLOGY FRAMEWORK TO MANAGE PRODUCTION SYSTEM DISRUPTIONS

The following subsections abstract the main concepts of the biological immunity and adapt them to the context of disruption management in production systems.

A. Self

Self corresponds to a definition of a set of normal functioning conditions of a production system. This definition can be provided by the production system experts. These normal functioning conditions can be described using for example reference values (or ranges of values) of:

- Key performance indicators (KPI),
- Technical parameters and/or specifications for products, resources, and activities,
- Predefined metrics,
- Etc.

For example, here are some instances of self:

- S1 = (normal supply time of raw material RM1 from supplier Sp1 = 100 units of time)
- S2 = (normal supply quantity of raw material RM1 from supplier Sp1 = 400 units)
- S3 = (normal transportation time of raw material RM1 by transporter Tp1 = 30 units of time)
- S4 = (normal inventory level of raw material RM1 ≥ 1 000 units)

B. Antigen

An antigen corresponds to a characterizing feature of a disruption, such as an excessive deviation or an erroneous value of a parameter. For example, here are two instances of antigen:

- Ag1 = (measured supply time of raw material RM1 = 150 units of time)
- Ag2 = (supply quantity controlled at receipt of raw material RM1 = 300 units)

These antigens may correspond to symptoms of a disruption like supplier late delivery or unavailability.

C. Pathogen Associated Molecular Pattern (PAMP)

A PAMP corresponds to a couple of (Antigen, Self) that describes which excessive deviation or erroneous value of a parameter (Antigen) does not satisfy which expected normal functioning condition (Self). For example, here are two instances of PAMP:

- PAMP1 = (Ag1, S1)
- PAMP2 = (Ag2, S2)

D. Pathogen

A pathogen is a type of disruption, which is characterized by a set of PAMPs. Figure 2 illustrates the structure of a pathogen. PAMPs allow describing sets of revealing symptoms of each disruption type.

- S1 = (normal supply time of raw material RM1 from supplier Sp1 = 400 units)
- S3 = (normal transportation time of raw material RM1 by transporter Tp1 = 30 units of time)
- S4 = (normal inventory level of raw material RM1 ≥ 1 000 units)

B. Antigen

An antigen corresponds to a characterizing feature of a disruption, such as an excessive deviation or an erroneous value of a parameter. For example, here are two instances of antigen:

- Ag1 = (measured supply time of raw material RM1 = 150 units of time)
- Ag2 = (supply quantity controlled at receipt of raw material RM1 = 300 units)

These antigens may correspond to symptoms of a disruption like supplier late delivery or unavailability.

C. Pathogen Associated Molecular Pattern (PAMP)

A PAMP corresponds to a couple of (Antigen, Self) that describes which excessive deviation or erroneous value of a parameter (Antigen) does not satisfy which expected normal functioning condition (Self). For example, here are two instances of PAMP:

- PAMP1 = (Ag1, S1)
- PAMP2 = (Ag2, S2)

D. Pathogen

A pathogen is a type of disruption, which is characterized by a set of PAMPs. Figure 2 illustrates the structure of a pathogen. PAMPs allow describing sets of revealing symptoms of each disruption type.

For example, disruptions of type supplier unavailability are characterized by excessive supply time (such as PAMP1), and/or missing quantity (such as PAMP2).

E. Danger signal

The danger theory helps us to model the propagation of a disruption in a production system. A danger signal corresponds to an evaluation of the impact (direct and potential consequences) of a disruption on the normal
functioning conditions of the production system. Processes that enable propagation of danger signals, deduction and evaluation of consequences are beyond the scope of this paper [8]. Nevertheless, the ontology allows capturing the results of these processes once they are carried out.

For example, these processes may establish that the supply delay described by antigen Ag1 impacts the supply quantity (S2) and introduces a danger signal DG1 representing a missing supply quantity. This missing quantity can be ignored by the transportation order and the rest of the quantity transported. In this case, danger signal DG1 impacts the inventory level and causes danger signal DG3 representing a potential consequence of material unavailability (inventory shortfall due to a missing partial supply quantity).

Antigen Ag1 also impacts the transportation time (S3), thus causing a danger signal DG2 representing a transportation delay. The transportation of all the supply quantity is delayed. In this case, Danger signal DG2 may further propagate and cause danger signal DG4 representing a potential consequence of material unavailability (inventory shortfall due to the total supply quantity missing). Consequently, there are four instances of danger signals:

- DG1 = missing supply quantity = 200 missing units of raw material RM1
- DG2 = transportation delay = 80 time units
- DG3 = inventory shortfall = 200 missing units of raw material RM1
- DG4 = inventory shortfall = 400 missing units of raw material RM1

F. Danger Associated Pattern (DAP)

We introduce the concept of DAP to represent a triplet of (Threat, Self, Output danger signal) that describes which threat (PAMP or input danger signal) impacts or risks to impact which Self. An estimation of this direct or expected impact is set as output danger signal. For example:

- DAP1 = (PAMP1, S2, DG1) is an instance of DAP where the threat is due to the disruption feature (delay of a supplier).
- DAP2 = (PAMP1, S3, DG2) is an instance of DAP where the threat is due to the disruption feature (delay of a supplier).
- DAP3 = (DAP1, S4, DG3) is an instance of DAP where the threat is due to the danger signal representing the delay of the supplier.
- DAP4 = (DAP2, S4, DG4) is an instance of DAP where the threat is due to the danger signal representing the delay of the transporter.

G. Danger

A danger is a type of risk that is expected to threaten the entities and/or activities of a production system. Figure 3 represents the structure of a danger, which is characterized by a set of DAPs that allow describing sets of revealing symptoms of each type of risk. For example, risks of type inventory shortfall are characterized by supply delays (such as DAP1), transportation delays (such as DAP2), and/or missing quantity (such as DAP3).

H. Cell

Body cells correspond to:

- Production system physical entities, such as parts and products, resources (e.g. machines, tools, operators and materials), and activities
- Production system logical entities, such as orders (e.g. supply, work and transportation), routings, bills of material, etc.

Each cell is related to other cells by constraints (e.g. technical, financial, environmental, legal, etc.) and relationships (e.g. time and precedence). The structure of a cell is shown in Figure 4. Self is used to describe both the normal functioning conditions of each cell, and the constraints and relationships between cells. DAPs are used to describe the impact of disruptions (antigens) and risks (danger signals) on a cell.
and/or its consequences. Here are some examples of antibodies:

- Ab1: update normal supply time of raw material RM1 from supplier Sp1 to 150 units of time
- Ab2: provide the 200 units missing quantity of raw material RM1 from supplier Sp2 in lieu of supplier Sp1

J. B cell

B cells are associated to monitoring and control tools and algorithms (such as scheduling, dispatching, sequencing, etc.) that can be used to find reaction decisions (antibodies) to react to a disruption and/or its direct and potential consequences. As it is shown in Figure 5, B cells are sensitive both to disruption features (antigens) and consequences (danger signals). They suggest monitoring and control decisions (antibodies) that are used to react to a disruption and/or its consequences.

![Figure 5. Structure of B cell concept.](image)

Examples of B cells:

- B1 = (Ag1, Ab1). This is an instance of a B cell that suggests updating the supplier delivery time. This B cell can be associated to the instruction of the quality management system that specifies the criteria for supplier selection and evaluation.
- B2 = (Ag1, Dg1, Ab2). This is an instance of a B cell that suggests finding an alternative supplier to provide the missing supply quantity. This suggestion may come from a Supply Chain Management tool that finds alternative suppliers able to provide a raw material.

K. T helper cells

T helper (Th) cells store immune responses. An immune response corresponds to a set of monitoring and control decisions that were applied to react to a disruption and its consequences. Immune responses reflect the tradeoffs that were found to coordinate between several available reaction decisions. The structure of T helper cells is shown in Figure 6.

![Figure 6. Structure of danger concept.](image)

V. ONTOLOGY IMPLEMENTATION

An important step in the development of the proposed ontology based framework is the selection of a language for ontology description, and a tool for ontology implementation [3]. As a language, we choose OWL, which is the most recent development in standard ontology languages [15]. OWL has a rich set of logical operators that allows to define and describe concepts, and to build hierarchies of concepts (also called taxonomies). Furthermore, the OWL logical model allows the use of reasoners, which can check the mutual consistency of the statements and definitions in the ontology, and can also recognize which concepts fit under which definitions [15]. As a tool, we choose Protégé [16], which is a free, open source ontology editor and knowledge-base framework. Protégé ontologies can be exported into a variety of formats, including OWL. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

Using the Protégé – OWL editor, we edit an ontology example to illustrate the proposed framework. As shown in figure 7, each concept introduced in section IV is implemented as an ontology class. Each class is specialized by further subclasses, which reflect concepts more closely related to entities of both the production system, and the disruption management process.

More particularly, the cell class holds the physical and logical entities of a production system, such as customers, suppliers, resources, raw materials, parts, products, operations, etc. The normal functioning conditions of these entities, such as values of key performance indicators, technical specifications, etc., are described by individuals in the Self class. Figure 8 shows how instance S1 of self (cf. subsection IV.A) is implemented in the ontology.

The pathogen class introduces a classification of disruptions, which is compliant with existing ones [9]. Failures in relation with customers include for example rush orders and cancellation of orders. Product failures include quality problems, such as wastes and rejects, and inventory problems. Resource failures include disruptions such as machine failures, tool breakage, operator unavailability, etc. supply failures include late deliveries, transportation problems, etc. Figure 9 shows how instance PAMP1 of PAMP (cf. subsection IV.C) is implemented in the ontology. This instance is used to define an instance of a disruption (pathogen) of type supplier failure (late delivery).
Finally, the BCell class is specialized into subclasses that are related to tools that can be used to generate decisions to react to a disruption and its consequences.

This ontology can be tailored to the specificities of a production system by extending the classes and defining further subclasses. Besides, adding more instances enriches the knowledge base and allows mining it for example for rules to determine characteristics of disruptions or characteristics of reaction decisions.

![Figure 7. Ontology example.](image)

**VI. CONCLUSION**

This paper suggested an ontology framework to capture and structure the knowledge involved during the process of disruption management in production systems. This is achieved based on the study and adaptation of the structure and relations between concepts from the biological immune system.

The framework enables direct representation of (1) disruptions, through a classification of disruption types, including disruption symptoms (e.g. deviation from normal behavior) and disruption features (e.g. disruption duration); (2) disruption impacts, through a classification of direct and potential disruption consequences (e.g. extra costs, delays, quality problems, etc.); and (3) decisions and strategies that led to the elimination or reduction of disruptions and their impacts. This framework is generic enough to allow deriving more specific knowledge models to represent and deal with disruption related knowledge in specific production systems. Ontologies built upon the proposed framework allow further exploitation and reuse of the stored disruption knowledge, particularly by domain specific application software. More particularly, multi agent systems can rely on such ontologies to represent, share and handle a common representation of disruption knowledge.

Our future work focuses on investigating the possibility to extend the ontology framework to allow dealing with imprecise, incomplete, ill structured and/or uncertain production system data and knowledge. Integration of the proposed ontology with other production system ontologies (such as product driven control ontologies), integration with specific software applications and business process tools as well as interoperability issues are also subject of interest and require further investigation.

**REFERENCES**


Figure 8. Implementation of instance S1 of self.

Figure 9. Implementation of instance PAMP1 of PAMP.