Proposal for a Generic Model Dedicated to Reconfigurable and Agile Manufacturing Systems (RAMS)

Imad Chalfoun\textsuperscript{a,b,*}, Khalid Kouiss\textsuperscript{a,b}, Anne-Lise Huyet\textsuperscript{a,b}, Nicolas Bouton\textsuperscript{a,b}, Pascal Ray\textsuperscript{a,b}

\textsuperscript{a}Clermont University, French Institute for Advanced Mechanics, Pascal Institute, BP 10448, F-63000 Clermont-Ferrand, France
\textsuperscript{b}CNRS, UMR 6602, IP, F-63171 Aubière, France

* Corresponding author. Tel.: +33473288000; fax: +33473288100. E-mail address: imad.chalfoun@ifma.fr.

Abstract

This paper proposes a generic model for Reconfigurable and Agile Manufacturing Systems (RAMS). This model is based on a decomposition of RAMS along two axes. The first, horizontal, axis concerns the system's structure and its configurations. The second, vertical, axis distinguishes the logical part and the physical part. Secondly, we propose a formalization and representation for each component of this model. In addition, we will represent RAMS in a modular manner. High-level meta-models for structure, configuration and operations will be presented using the systems modelling language SysML. Finally, an assembly process will be introduced to implement this approach.

Keywords: Manufacturing systems; Reconfigurability; Agility; Generic model; SysML

1. Introduction

A manufacturing enterprise needs to stand on three equally strong legs to be stable: innovative products, reconfigurable manufacturing systems and responsive business models to sell a variety of products. In its turn, every manufacturing plant should have three goals: to produce at low cost, to enhance product quality, and to possess capabilities for rapid responsiveness. At the end of the 20\textsuperscript{th} century, manufacturing enterprises faced many challenges and changes, concerning market changes (changes in product demand, changes in current products and introduction of new products), customer orders (low cost, high quality, low volume products and custom products), government regulations (safety and environment) and system failures (maintain production despite equipment failures). It is certain that, in some cases, traditional manufacturing systems (DML: Dedicated Manufacturing Lines, FMS: Flexible Manufacturing Systems) are no longer able to respond to market conditions. Thus, global economic competition and rapid social and technological changes have forced manufacturers to face a new economic objective: manufacturing responsiveness. A new type of manufacturing system, a Reconfigurable Manufacturing System (RMS), has been developed in order to provide exactly the capacity and functionality needed, exactly when it is needed. [1-2-3-4-5-6-7-8]

In recent years, the evolutionary progress and paradigm shifts of manufacturing systems have been moving towards agile manufacturing systems. The concept of agility will reduce the time to reach market with appropriate products/services. Agile manufacturing is defined as the capability of surviving and prospering in a competitive environment of continuous and unpredictable change, by reacting quickly and effectively to changing markets, driven by customer-designed products and services. [3-9-10]

The aim of this paper is to propose a generic model for Reconfigurable and Agile Manufacturing Systems (RAMS) which permits the modelling of real manufacturing systems and allows a good description of reconfiguration activities. The reconfiguration function of a RAMS has already been characterized. [11]

In the first part, we will explain the organizing principle. The key point is to describe the RAMS along two axes. The horizontal axis concerns the system...
structure and its configurations. The structure defines the system components. Therefore, the configuration describes how the structure components are used and arranged. The vertical axis distinguishes the logical part and the physical part. In the proposed model, the concept of “operations” plays a key role, since they link the physical part to the logical part.

Secondly, we will represent the RAMS in a modular manner, so we propose a formalization and representation for each component of this model. High-level meta-models for the structure, the configuration and the operations will thus be presented using the systems modelling language SysML.

Finally, an assembly process application will be introduced to implement these developments.

2. Organizing principle

We will explain in this section how we organize a generic model for a RAMS. The design phase proposed is based on a decomposition of RAMS along two axes (Fig 1).

The first axis, which is horizontal, describes the system structure and its configurations, knowing that we can match a single structure to a set of different configurations. The structure defines the system elements and its capabilities as resources and products and the connections between them. At the same time, the configuration describes how the structure elements are used and how they are organized.

The second axis, which is vertical, distinguishes the logical part describing the functions to perform and the physical part describing the organization of different material elements that will carry out these functions. [6-12]

The logical structure is composed of products and functions. This part describes the functions which must be performed on the product, and also defines the process plan which leads to the final product. The physical structure describes the arrangement of the different material elements that compose the system as the stationary resources (machines, buffers,...), the transport resources (robots, conveyors,...) and the different connections between them. For example, should they be arranged in a serial line, a pure parallel system, or some combination? The connections represent the potential transfer links between the stationary resources; these connections are associated with the transport resources which can execute them.

The logical aspect of the configuration (software) is effectively the implementation of control programs for each component of the structure. The physical configuration (hardware) represents the resources used and the capabilities required for each resource. Therefore, the reconfiguration is the ability to modify the number of resources, capabilities and organization to meet the needs of production.

The operations then play a key role in the model, since they combine the functions with the resources, and they connect the structure to its configurations and the physical part to the logical part. They will be detailed in section five.

This is the organizing principle of the generic model which allows us to take into account the various aspects of the reconfiguration such as the variable number of machines, assigning different tasks and several layouts.

In the next section, we will propose a formalization and meta-model for the structure, and a formalization for the connections.

Fig. 1. Organization of the RAMS generic model

3. Structure

The structure defines the elements constituting the system, as well as the connections between them.

3.1. Structure formalization

The structure can be represented by the following pair:

\[ S : \{S_p, S_l\} \]  

Where \( S_p \) refers to the physical structure and \( S_l \) to the logical structure. \( S_p \) is described by the triplet:

\[ S_p : \{R, Capab, Conn\} \]  

Where \( R \) is the set of structure resources (stationary resources, transport resources and automated guided vehicles), \( Capab \) describes the capabilities of each resource, and \( Conn \) represents a connection application such as:

\[ Conn : R_s \times R_s \times R_t \rightarrow \{0, 1\} \]

Each connection links two stationary resources together through a resource transport which carries out the transfer.

\( S_l \) is represented by the quintet:

\[ S_l : \{F, G, Pr, Prec, Affec\} \]

Where \( F \) represents the functions set (\( F_{tr} \): work functions, \( F_{ts} \): transport functions and \( F_{st} \): storage functions).
functions), \( G \) is the set of logical sequences and \( Pr \) is the set of product types.

The application \( Prec \) defines the precedence relation between functions within a logical sequence:

\[
Prec: G \times F \times F \rightarrow \{0, 1\}
\]

The application \( Affec \) defines the assignment of one or more logical sequences for all products types:

\[
Affec: G \times Pr \rightarrow \{0, 1\}
\]

### 3.2. Structure meta-model

The Systems Modelling Language (SysML) provides the Block Definition Diagram (BDD) which allows us to represent the structure into modular components, or blocks, with relations between them (Fig 2). [13]

---

**Fig. 2. Structure meta-model**

This figure represents the structure \( S \) of a particular viewpoint, called a meta-model, that illustrates the modular components to be built and the network of relationships which ensures the data flow between these components. The goal of this meta-model is to build the RAMS structure in a modular manner, which will facilitate system reconfiguration and collaborate with other components to perform the agility required.

The “System Structure” block represents the system structure and has two relations, with the physical structure and with the logical structure.

The physical structure is composed of resources, capabilities and connections. Three types of resources are considered: stationary resources (machines, emergency machines and buffers), transport resources (conveyors, robots and loaders/unloaders) and an Automated Guided Vehicle (AGV). With regard to machines, the “Port” block defines the characteristic locations for transfers. The connections are represented by the “Connections” block; they link the machine ports via a conveyor or a robot.

With respect to the logical structure, three types of functions are defined: work functions, which are involved in the realization of products, transport functions, which are associated with connections, and storage functions, which connect the loader/unloader to the buffers. The “Products” block represents all types of products that are described by the process plan (“Process Plan” block) representing the function sequences. The “Precedence” block defines the precedence relation between functions within a process plan. The “Assignment” block assigns one or more process plans to each type of product.

### 3.3. Connections

The connections represent potential transfer links between machines; these connections are associated with the transport resources (conveyors, robots) that can run them.

We suppose that a configuration is already given. This configuration therefore defines: number of movable machines \( N \) and both IN/OUT buffers, number of stages \( m \) with two IN and OUT stages, number of machines in each stage \( np[m] \), machine arrangement, connections between them, Cartesian coordinates \((x_i, y_i)\) for each machine at the workshop level and \((x_{in}, y_{in})\), \((x_{out}, y_{out})\), which are respectively the coordinates of the IN and OUT buffers.

We represent the machine at stage \( i \) and local number \( j \) by \( P[i,j] \) as an element of the transposed matrix of:

\[
P = (P[i,j])_{3 \times 3 \times m} \text{ with } 1 \leq \max (np[m])
\]

Such that \( \max (np[m]) \) is the maximum between the number of machines in each stage.

The Cartesian coordinates of \( P[i,j] \) are:

\[
P[i,j].coordinates = (x_i, y_i)
\]

The transport resource associated with \( P[i,j] \) is \( R[i,j] \).

The aim is to integrate the concept of agility with the reconfiguration function. For this, we propose to define the following machines for each machine from IN to OUT buffers, considering that this will aid agility in the control programs.
The following machines of $P_{i,j}$ are represented in a vector:

$$P_{i,j}.\text{FollowingMachines}$$

The number of following machines of $P_{i,j}$ is:

$$\text{size}(P_{i,j}.\text{FollowingMachines})$$

We define connections in the workshop considering that $X_{i,j}$ means that $X$ is connected with $Y$:

Between $IN$ and the first stage ($m = 1$), the connections are performed by the loader:

For $i = 1$ to $\text{np}[1]$

$IN, P_{i,j}$ performed by the loader;

Between the last stage $m$ and $OUT$, the connections are performed by the unloader:

For $i = 1$ to $\text{np}[m]$

$P_{m,i}, OUT$ performed by the unloader;

Between the first stage $m = 1$ and the last stage $m$, the connections are performed using the transport resources (conveyor, robot) associated with machine $P_{i,j}$:

For $i = 1$ to $m - 1$
For $j = 1$ to $\text{np}[i]$
For $k = 1$ to $\text{size}(P_{i,j}.\text{FollowingMachines})$

$P_{m,i}, P_{i,j}.\text{FollowingMachines}[k]$ performed using $R_{i,j}$ which is associated with the $P_{i,j}$;

We have therefore modelled the connections in a manner to move and connect easily the machines from its coordinates and neighbours. This reflects the system’s mobility that helps to reconfigure RAMS quickly and contributes to the agility of the system.

In the next section, we will present the formalization and meta-model for the configuration.

4. Configuration

Configurations define the various uses and organizations of the structure elements so that each configuration meets its objective.

4.1. Configuration formalization

We can represent each configuration by the following pair:

$$C_{i} : (C_{p}, C_{l})$$

(9)

Where $C_{p}$ is the physical configuration and $C_{l}$ the logical configuration.

$C_{p}$ defines the physical configuration, described by the pair:

$$C_{p} : (R_{\text{util}}, \text{Capab_assoc})$$

(10)

Such that $R_{\text{util}}$ is the resource set used in this configuration; this set is included in the global resource set $R$. $\text{Capab_assoc}$ defines the required capabilities associated with each resource and thus, for each resource, certain modules (tools and/or devices) will be eligible for use.

We define two applications:

$$\text{RessUsed} : R_{\text{util}} \times R \rightarrow \{0, 1\}$$

(11)

This defines the resources used in this configuration.

$$\text{CapabAssoc} : \text{Capab_assoc} \times \text{Capab} \rightarrow \{0, 1\}$$

(12)

This application enables the capabilities associated with each resource in this configuration to be specified.

The physical configuration is therefore a key indicator to evaluate this configuration in terms of cost.

The logical configuration is represented by the set of control programs corresponding to products, process plans and configurations:

$$C_{l} : \{\text{Prog}\}$$

(13)

The “Implement” application allows a control program to be implemented for each resource used in the current configuration:

$$\text{Implement} : \text{Prog} \times R_{\text{util}} \rightarrow \{0, 1\}$$

(14)

4.2. Configuration meta-model

In the same way as for the structure, the high-level meta-model allows us to represent the configuration in modular block components with relations between them (Fig 3).

In the next section, we will present the operations, which are principal element in the generic model.
5. Operations

We distinguished six types of operations at the workshop level: main operations, transport operations, storage operations, activation operations, assignment operations and implementation operations. These operations relate the system structure with different configurations and also connect the physical part to the logical part (Fig. 4).

The main operations are defined by the "Main Operations" block; it is an example of a work function associated with machines. Each machine must carry out at least one main operation and each function is implemented by one operation; for example: \( \text{Op}_1 (M_1, F_{tr.1}) \). The following application describes the operations implementing work functions on associated machines:

\[ \text{Op}_\text{main} : \text{Op} \times F_{tr} \times R \rightarrow 0, 1 \]  \hspace{1cm} (15)

Transport operations are represented in the "Transport Operations" block, each transport operation being related to a transfer (connection) and to a corresponding transport function. The second application defines the operations associating the transport functions with the different connections:

\[ \text{Op}_\text{transport} : \text{Op} \times F_{ts} \times \text{Conn} \rightarrow 0, 1 \]  \hspace{1cm} (16)

Storage operations store products in IN and OUT buffers; they are described in the "Storage Operations" block. St0 (Pr0, IN) is used to store the raw product Pr0 in the IN buffer and Stj (Prj, OUT) allows us to store the final product (produced on a machine in stage m) in the OUT buffer. We define an application which describes these operations:

\[ \text{Op}_\text{storage} : \text{Op} \times F_{st} \times \text{Buffer} \rightarrow 0, 1 \]  \hspace{1cm} (17)

Assignment operations are used to enable the resources used in a configuration. This resource part will be eligible for use:

\[ \text{Op}_\text{assignment} : \text{Op} \times R \times \text{util} \rightarrow 0, 1 \]  \hspace{1cm} (18)

Activation operations enable us to specify the modules (tools and/or devices) of the resources used according to the configuration requirements. For example, we connect a screw gun and at the same time we disconnect a power drill on a machine provided with these tools:

\[ \text{Op}_\text{activation} : \text{Op} \times \text{capab.assoc} \times R \times \text{util} \rightarrow 0, 1 \]  \hspace{1cm} (19)

Implementation operations are allocated in order to implement control programs on each resource used in this configuration. For example, within a process plan, we enable the required operations on a machine and at the same time we disable the other operations that this machine is capable of doing:

\[ \text{Op}_\text{implementation} : \text{Op} \times \text{Prog} \times R \times \text{util} \rightarrow 0, 1 \]  \hspace{1cm} (20)

Regarding the sequence of operations represented by the "sequences of operations" block, each sequence is associated with a process plan for a product.

Therefore, the operations represent another key indicator to evaluate this configuration in terms of operating time.

Fig. 4. Operations meta-model

6. Assembly process application

In this paper, we discuss the process of assembling electronic components on printed circuit boards.

The physical structure is represented by stationary resources (IN, OUT, M1, M2, M3, M4, and M5), transport resources (loader, unloader, R1, R2, Cv1, Cv2), various connections (arrows) and the capabilities of each resource, knowing that each resource can carry out several operations. For example, M1 is able to perform the following tasks: Soldering SMD components (Surface Mounted Device), soldering simple electronic components manually and/or automatically, and mounting components.

The logical structure is composed of products, three types of boards, five process plans to obtain these products by carrying out work functions; and functions:

\[ \{ \text{Pr}_1, \text{Pr}_2, \text{Pr}_3 \} \]

Knowing that each work function is represented by a single operation that must be implemented at least once. For example, F_tr1 represents soldering SMD components:

\[ F_{tr} = \{ F_{tr1}, F_{tr2}, F_{tr3} \} \]  \hspace{1cm} (21)

Each transport function is implemented by a single operation which is linked to a connection, such as F_ts1 from IN to M1:

\[ F_{ts} = \{ F_{ts1}, ..., F_{ts10} \} \]  \hspace{1cm} (22)

Each storage function is implemented by a single operation; they can store the raw product Pr0 and the final product Prj in buffers IN and OUT respectively:

\[ F_{st} = \{ F_{st.in}, F_{st.out} \} \]  \hspace{1cm} (23)

Next, we assign the process plans to products, for example, G1, G2 to Pr2. We define the precedence...
relation between functions within each process plan such as:

\[ G_3 = \{F_{tr1}, F_{tr2}, F_{tr3}\} \]  \hspace{1cm} (25)

In practice, to set up a new configuration, we start from the product. To launch a product \( Pr_2 \), we choose the corresponding process plan, such as \( G_i \).

Then we choose the resources capable of executing all the process plan functions. The resources used are:

\[ R_{_util} = \{IN, OUT, M_3, M_5, loader; unloader; R_2, Cv_2\} \]  \hspace{1cm} (26)

Which represent the first part of the physical configuration.

We define an organization of these resources with connections to carry out the process plan (Fig 5).

![Fig. 5. Assembly of electronic boards – configuration 1](image)

Regarding the operations, we define the main ones. For example, \( Op_{1}(M_1, F_{tr1}) \) to solder SMD components, \( Op_{2}(M_3, F_{tr3}) \) to mount components and \( Op_{3}(M_5, F_{tr2}) \) for engraving. Similarly, we define transport, storage, activation, assignment and implementation operations.

At the operations assignment level, which represents the second part of the physical configuration, we can for example enable the devices capable of performing control, mounting components and wiring on machine \( M_3 \), and at the same time we disable all other tools and devices. We also define the sequence of operations, such as: \( Op_{st\_in}, Op_{ts\_1}, Op_{1}, Op_{ts\_2}, Op_{a}, Op_{ts\_3}, Op_{5}, Op_{ts\_4} \) and \( Op_{st\_out} \).

Finally, the logical configuration is represented by control programs compatible with the current configuration and sent to each resource used. This represents the implementation operations.

Let us suppose, for example, a failure occurs suddenly on machine \( M_3 \), the habit is to stop the line to repair and then to continue the production plan. The schedule will then be shifted to repair the model. On the contrary, the generic model of RAMS in this paper helps to reconfigure this line quickly and effectively by modifying the currently used resources, coordinates, associated capabilities and control programs for each component of the new structure. Therefore, we reassign new operations to new resources (Fig 6).

![Fig. 6. Assembly of electronic boards – configuration 2](image)

Thus, the reconfiguration function reduces the time with all the positive consequences in terms of logistics. Consequently, using a RAMS will help to improve the quality of service and provide a several of advantages.

7. Conclusion

In this paper, we have presented a generic model for Reconfigurable and Agile Manufacturing Systems (RAMS). The structure and configuration were detailed along two axes and we discussed the logical and physical parts which are linked by operations. RAMS was thus represented in a modular manner using SysML. We applied these developments to an assembly process system. This generic model can also be deployed on other assembly systems that regularly need to be reconfigured quickly and effectively.

In the future, this approach will be integrated into a global process to design RAMS. Shortly, we will attempt to evaluate the configuration performance, proposing a process with different criteria, and subsequently we will address the implementation of a configuration.

References