3

A modeling approach for mode handling of flexible manufacturing systems

Nadia Hamani and Abderahman El Mhamedi
Modélisation et Génie des Systèmes Industriels (MGSI, LISMMA- EA 2336)
IUT de Montreuil-Université de Paris 8
France

1. Introduction

Due to increasing competitiveness, Flexible Manufacturing Systems (FMS) were introduced to overcome the drawbacks of Dedicated Manufacturing Lines (DML) (Koren et al., 1999). Indeed, FMS are able to carry out several parts in small and average series while adapting quickly the production changes demand thanks to their flexibility (Ranky, 1990). Several research works focus on the design of fault tolerant control systems of FMS. However, the design of such systems, in particular the supervision function, is difficult due to increasing flexibility and complexity. Thus, the aim of our research project is to provide a fault tolerant control system dedicated to FMS. This control system ensures on line and real time management of failures. In view of a disturbance, the supervision role is to take necessary decisions to return to normal or accepted operation. The supervision according to our approach is made up of three functions: decision, piloting and mode handling. Mode handling function is the scope of this chapter.

The purpose of this chapter is to present a new modeling approach for mode handling of Flexible Manufacturing Systems (FMS). Based on a review of the modeling methods and the specification formalisms in the existing approaches, we show that the mutual benefit of functional modeling and synchronous languages is very convenient for mode handling problem. We start by introducing the context of our work and the basic concepts of the proposed modeling approach. Then we present the steps of functional modeling and we illustrate them through an example of a flexible manufacturing cell. Functional modeling is completed by generic behavioral specifications representing the states of a subsystem or the whole system. The specification method is modular, hierarchical and supports re-use concept. The established model is generic and well adapted to our control system framework. Mode handling function role within the control system is then studied. This function enables a reactive update of the availability of the resources and functions and the transmission of high level control and reconfiguration orders.

This chapter is organized as follows. After the presentation of the context of our study in Section 1, Section 2 presents the roles of mode handling function within the control system. Section 3 introduces at first the main characteristics and the basic concepts of the proposed modeling method. Then the functional modeling steps are detailed. The behavior specification of the FMS and its subsystems are then presented. The logical relationships between those subsystems are formalized. The modeling method is illustrated through an example of a flexible manufacturing cell. Finally Section 4 focuses on carrying out mode handling function within the control system framework.

2. Mode handling function

The operator supervising a production system should manage this latter both under nominal and degraded operation. Due to the complexity, he/she cannot apprehend all the constraints in a coherent and optimal way. Thus, mode handling function needs to provide a model of the system which takes into account the operating modes of the whole system and its subsystems. To this aim, it is important to use an adequate modeling method and powerful specification formalisms. The modeling method should be well adapted to the complexity and to the production system characteristics by allowing an easy representation of its operating modes. When setting up the production demand, the decisions are affecting all the subsystems of the production system. Therefore the specification formalism should support the hierarchy and represent several forms of preemption. Moreover, it must be sufficiently formal to allow carrying out formal verification.

Some research works in the literature focus on mode handling of Automated Production Systems (APS). The modeling methods and the specification formalisms used in the existing approaches are compared in (Hamani et al., 2005).

The modeling methods we studied are classified according to many criteria (Hamani et al., 2005). To deal with complexity, the system is decomposed into subsystems. Two approaches are distinguished (Toguyeni et al., 1996a), structural approaches (Biland et al., 1994; Bois et al. et al., 1992; Parayre et al., 1992) and functional ones (Ausfelder et al., 1993; Dangoumau & Craye, 2001; Kermad et al., 1994). The modeling steps, the specification of the modes and the specification formalisms are detailed in (Hamani et al., 2005) for each approach. The role of the established models within the control systems is also studied. In structural approaches, the decomposition is based on structural relationships between the resources of the system. Reconfiguration actions are then taken on the resources, the workstations and the cells. Functional approaches are concerned with the services delivered by the system rather than the resources. The decomposition is based on functional relationships between subsystems. Such relationships enable the implementation of automated reconfiguration procedures.

Our study shows also that the specification formalisms should enable an easy representation of hierarchy, concurrency and preemption. They have to be also with rigorous semantics in order to guarantee some properties using formal proofs.

The main conclusions of our comparative study show that the mutual benefit of a functional modeling approach and a powerful specification model using the graphical synchronous language Safe State Machines (SSM) (André, 2003) is very convenient for APS mode handling. Indeed, functional approaches are well suited for efficient reconfiguration procedures. Besides, the required characteristics of hierarchy, parallelism, and preemption are well represented using synchronous languages instead of common models such as Sequential Functional Charts (SFC) and Petri Nets (PN). The interests of using synchronous formalisms rely also on their strong semantics. They are compiled efficiently and it is possible, using formal techniques, to prove that the behavior of the system respects some properties (Hamani et al., 2007). In addition, synchronous approaches ensure the preservation of the verified properties between the specification and the implementation phases of the development process (What You Prove Is What You Execute: WYPIWYE) (Berry, 1989).

Mode handling function ensures both an informational role and an operational role within the control system. The information provided by mode handling function about the state of the system and the subsystems, enables updating the models of the control system. The informational, decisional and operational functions of the control system are respectively monitoring (Elkhattabi et al., 1995; Toguyeni et al., 1996b), decision (Berruet et al., 2000), piloting (Tawegoum et al., 1994) and mode handling. The reaction loop in view of a failure is given in the following: the monitoring function detects and localizes the failures in the plant; the decision function provides the new configuration of the system; Mode handling and piloting functions carry out the decisions about the new configuration.

Indeed, concerning its operational role within the control framework, mode handling with piloting implements some actions on low level control and on the plant. Mode and production goals changing are ensured by mode handling function whereas the actions on low level control are carried out both by mode handling and piloting functions. Piloting solves remaining indeterminism resulting from routing flexibility. Berruet (Berruet et al., 2000) introduces several reconfiguration solutions; they are classified according to the difficulty of their implementation by the operational functions (i.e. piloting and mode handling) and the required time.

The interactions of mode handling with all the functions of the control system are necessary to achieve its role. Indeed, mode handling receives as inputs high level control orders and reconfiguration decisions and the failures and provides as outputs the current configuration of the system. This latter characterizes the state of the system and of its subsystems (resources, operations) and the operating parameters. In order to achieve its role mode handling function needs the models representing the behavior of the system. Reconfiguration goals are reached under some performance and time constraints and using some methods.

3. The modeling method

The advantages of functional modeling approaches and synchronous languages for mode handling of APS are shown in the previous section. Our approach is based on a functional modeling and a synchronous reactive approach using SSM (André, 2003).

Besides, this approach is well adapted to FMS because it is based on the mission concept which represents the flexibility of production of such systems. A mission (or a production goal) is a subset of Logical Operating Sequences (LOS) which are carried out simultaneously.

The model represents the subsystems and their operating modes for an existing system and also for a system being designed. The modeling approach is based on generic concepts. For a given system, the predefined functional and behavioral subsystems are instantiated to generate the model.

3.1 Basic concepts

The Logical Operating Sequence (LOS) concept was introduced for the specification of control models in (Cruette et al., 1991). A LOS is a subset of ordered machining functions that the FMS performs on some parts.

A function (f) is the response of a system or a component to a given stimulus when it works normally independently of its environment (Toguyéni et al., 2003). A LOS is noted LOS $f_1 ... f_n$ or LOS f_i (i = 1, ..., n). The functions performed by the FMS are mainly: machining functions, transfer functions and storage functions.

Let us remind the operation concept and the accessibility relations between operations. In an FMS, an operation (Op) is defined as a function carried out by a resource (a component) of the FMS (Berruet et al., 2000; Toguyeni et al., 2003). An operation is noted $Op_{Ri,\ fi}$ where f_i is the performed function and R_i the resource which implements it. As for functions, several kinds of operations are defined: machining operations, transfer operations and storage operations.

The concept of elementary operation was also introduced (Berruet et al., 2000; Toguyeni et al., 2003). An elementary operation is an operation carried out only once, continuously, i.e. without another alternative during the normal execution of the operation. Several kinds of elementary operations are also defined: elementary machining operations, elementary transfer operations, etc. The concept of elementary operation is important for FMS modeling because it is a final step in the decomposition process.

The accessibility relation represents the link between some resources concerning the parts flow. Thus, the accessibility relation characterizes the existence of a parts flow between two resources (Amar et al., 1992). A resource has a direct accessibility with another resource if a part moves from one resource to another one without going through an intermediate resource. Loading or unloading a machine tool using a robot is a typical example of direct accessibility. Indirect accessibility relation results from transitivity.

The accessibility relation between operations results directly from the existing accessibility between resources. It means a directed flow of parts between two operations (precedence relation).

The Characteristic Area (CA) concept (Amar et al., 1992) characterizes a physical area (machining, transfer or storage areas) mobile or not, able to receive a part. A physical area is being characteristic if it is a machining area or if it is in an external accessibility with a physical area of another production medium. From a functional point of view, a CA is either a machining resource or an access point to a resource. By extension to the CA concept, Toguyéni (Toguyéni et al., 2003) introduced the Main Characteristic Area (MCA) concept. It is a CA which corresponds to a machining area or a storage area used as an input or an output area for the parts.

The MCA enables to introduce the elementary transfer concept proposed in (Toguyéni et al., 2003). An elementary transfer (TrE) is performed by one resource between two Characteristic Areas. It is noted $\mathrm{TrE}_{R_i}^{S \to D}$ with S a source CA, D a destination CA, and R_i the transfer resource.

The other concepts we need for the modeling process are introduced in the following subsection.

3.2 The FMS functional model

An FMS produces a set of parts simultaneously. Each part has its own LOS. Usually we desire to change the production goal that is why it is interesting to introduce the mission concept (Hamani et al., 2006).

Definition 1. A mission ($\mathcal{M}$) is a subset of Logical Operating Sequences which are carried out simultaneously.

Each function of a LOS is implemented by one or more operations performed by distinct resources. In order to achieve a mission of the FMS, it is necessary to carry out the corresponding LOS. This needs the implementation of machining functions and also the corresponding machining operations. The corresponding transfer operations should also be performed. Thus, we propose to determine, for each machining area of the FMS, the transfers that ensure successful loading and unloading of a part. The part is loaded from a machining area or from the cell input. Then it is unloaded onto another machining area or onto the cell output. So transfer operations are generic entities which depend only on the plant and do not depend on production goals.

For each MCA corresponding to a machining area, the Access Transfers (TrA) concept is introduced (Hamani et al., 2006). This concept represents the subset of transfer operations that allow loading (unloading) a part from (onto) a machining area.

Definition 2. Access Transfers (TrA) associated with a machining area (or a MCA) are the subset of elementary transfer operations that connect this area with the other areas of the FMS. The TrA corresponding to a machining MCA are noted $\mathrm{TrA}_{\mathrm{machining\_MCA}}$.

In order to determine TrA, a first step consists in listing symmetrical transfers between MCA representing both sources and destination areas. Then it is necessary to refine these transfers until obtaining elementary transfer operations.

Once the Access Transfers are determined, it is necessary to identify elementary transfers which compose them. If there is a direct accessibility between two MCA then $\mathrm{Tr}_{\mathrm{MCA\_source} \rightarrow \mathrm{MCA\_destination}}$ corresponds to an elementary transfer. If not, it is necessary to refine the transfers between the CAs until obtaining elementary transfers. The possible paths are then established and those which are redundant are linked together with a logical OR.

Once TrA are established for each machining area, they are gathered together with machining operations corresponding to the same area in order to obtain an aggregate operation (Hamani et al., 2006).

Definition 3. An aggregate operation corresponding to a machining area is a subset of elementary machining operations and TrA. An aggregate operation associated with a machining MCA is noted $\mathrm{Op}_{\mathrm{MCA\_machining}}$.

Aggregate operations are generic entities which represent the possible choices for implementing machining functions by the FMS. So with each function of a LOS is associated its possible achievements. They are aggregate operations for which the machining operation is defined. A machining function is then implemented by an aggregate operation or some aggregate operations performed by distinct resources.

Finally we should determine machining resources and transfer resources which perform the elementary machining and transfer operations of the aggregate operations.
The specification steps of the functional model are described in the following. The modeling method is illustrated using a flexible manufacturing cell (Fig. 1) with two machines M_1 and M_2 and INPUT/OUTPUT buffers.

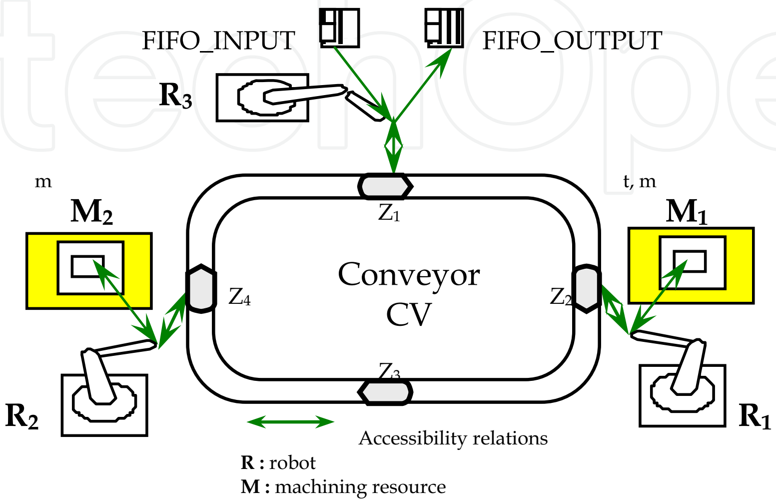

Fig. 1. The machining flexible cell

The machines are loaded with a transport system using three robots R_1, R_2 and R_3 and a conveyer (CV). Moving direction is $Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Z_4 \rightarrow Z_1$. M_1 is loaded using R_1 and M_2 is loaded using R_2. The parts are loaded onto the conveyor using the robot R_3. The machining functions performed by the system are turning (t) and milling (m). Turning function is carried out by M_1 and milling by M_1 and M_2. The functional requirements provide three missions. The CA of this machining cell are storage areas FIFO_INPUT, FIFO_OUTPUT, conveyor zones Z_1, Z_2, Z_3 and Z_4 and the machining areas M_1 and M_2. The MCA are storage areas FIFO_INPUT, FIFO_OUTPUT and machining areas M_1 and M_2.

1st step: Identification of the entities of the model
- List the missions that the FMS should carry out
- List for each mission its corresponding Logical Operation Sequences
- For each LOS determine the corresponding machining functions
- A machining function is implemented by one or several elementary machining operations. Each one is belonging to an aggregate operation.

- Determine the aggregate operations of the FMS (see the 2nd step)
- For each aggregate operation, identify the resources performing it (see the 3rd step)

According to functional requirements, three missions can be selected: $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$. The corresponding LOSs are the following:
$\mathcal{M}_1$: LOS_1 and LOS_2; $\mathcal{M}_2$: LOS_1, LOS_2 and LOS_{12}; $\mathcal{M}_3$: LOS_1, LOS_2 and LOS_{21}

The machining functions which compose each LOS are the following:
LOS_1: turning; LOS_2: milling; LOS_{12}: turning then milling; LOS_{21}: milling then turning

The turning function is performed by the elementary machining operation $Op_{M1,t}$ belonging to the aggregate operation Op_{M1}. The milling function is performed by the elementary machining operation $Op_{M1,m}$ belonging to the aggregate operation Op_{M1} or by the elementary machining operation $Op_{M2,m}$ belonging to the aggregate operation Op_{M2};

2nd step: Determination of aggregate operations
For each machining area of the FMS:

- Identify elementary machining operations which are performed in this area
- Identify the TrA corresponding to this area
- Gather elementary machining operations together with TrA to obtain aggregate operations

For the machining area M_1:
The elementary machining operations performed by M_1 are $Op_{M1,t}$ and $Op_{M1,m}$;

Access Transfers corresponding to the machining area M_1 are[1]:

TrA_{M_1} = AND ($Tr_{MCA\_source \to M_1}$, $Tr_{M_1 \to MCA\_destination}$).

TrA_{M_1} = AND [OR ($Tr_{IN \to M_1}$, $Tr_{M_2 \to M_1}$), OR ($Tr_{M_1 \to M_2}$, $Tr_{M_1 \to OUT}$)],

TrA_{M_1} = AND {OR [AND ($TrE_{R_3}^{IN \to Z_1}$, $TrE_{CV}^{Z_1 \to Z_2}$, $TrE_{R_1}^{Z_2 \to M_1}$), AND ($TrE_{R_2}^{M_2 \to Z_4}$, $TrE_{CV}^{Z_4 \to Z_1}$, $TrE_{CV}^{Z_1 \to Z_2}$, $TrE_{R_1}^{Z_2 \to M_1}$)], OR [AND ($TrE_{R_1}^{M_1 \to Z_2}$, $TrE_{CV}^{Z_2 \to Z_3}$, $TrE_{CV}^{Z_3 \to Z_4}$, $TrE_{R_2}^{Z_4 \to M_2}$), AND ($TrE_{R_1}^{M_1 \to Z_2}$, $TrE_{CV}^{Z_2 \to Z_3}$, $TrE_{CV}^{Z_3 \to Z_4}$, $TrE_{CV}^{Z_4 \to Z_1}$, $TrE_{R_3}^{Z_1 \to OUT}$)]};

The elementary machining operations $Op_{M1,t}$ et $Op_{M1,m}$, gathered together with TrA_{M1} enable to obtain the aggregate operation Op_{M1} = AND [OR ($Op_{M1,t}$, $Op_{M1,m}$) , TrA_{M_1}];

3rd step: Determination of the resources which perform elementary operations
For each aggregate operation:

- Associate with each elementary machining operation the resource or the configuration of the resource (in the case of a polyvalent resource) which performs it
- Associate also with each elementary transfer operation the resource(s) performing it, redundant resources are linked with a logical OR

The aggregate operation associated with the machining area M_1 is performed by the following resources:

- The polyvalent machining resource M_1 performs the elementary operations $Op_{M1,t}$ and $Op_{M1,m}$.

For transfer resources:

---

[1] This notation uses the logical AND and OR and uses also three distinct levels: '{' for the first level, '[' for the second level and '(' for the third level.

- R_1 performs the elementary transfer operations $TrE_{R_1}^{Z_2 \to M_1}$ and $TrE_{R_1}^{M_1 \to Z_2}$
- R_2 performs the elementary transfer operations $TrE_{R_2}^{M_2 \to Z_4}$ and $TrE_{R_2}^{Z_4 \to M_2}$
- R_3 performs the elementary transfer operations $TrE_{R_3}^{IN \to Z_1}$ and $TrE_{R_3}^{Z_1 \to OUT}$
- CV performs the elementary transfer operations $TrE_{CV}^{Z_1 \to Z_2}$, $TrE_{CV}^{Z_2 \to Z_3}$, $TrE_{CV}^{Z_3 \to Z_4}$ and $TrE_{CV}^{Z_4 \to Z_1}$

The aggregate operation corresponding to the machining area M_2 is obtained in the same manner.

The functional model is represented with the following entities:

- The missions
- The logical operating sequences
- The machining functions
- The aggregate operations
  - elementary machining operations
  - access transfers (subset of transfer operations)
- Transfer resources, machining resources

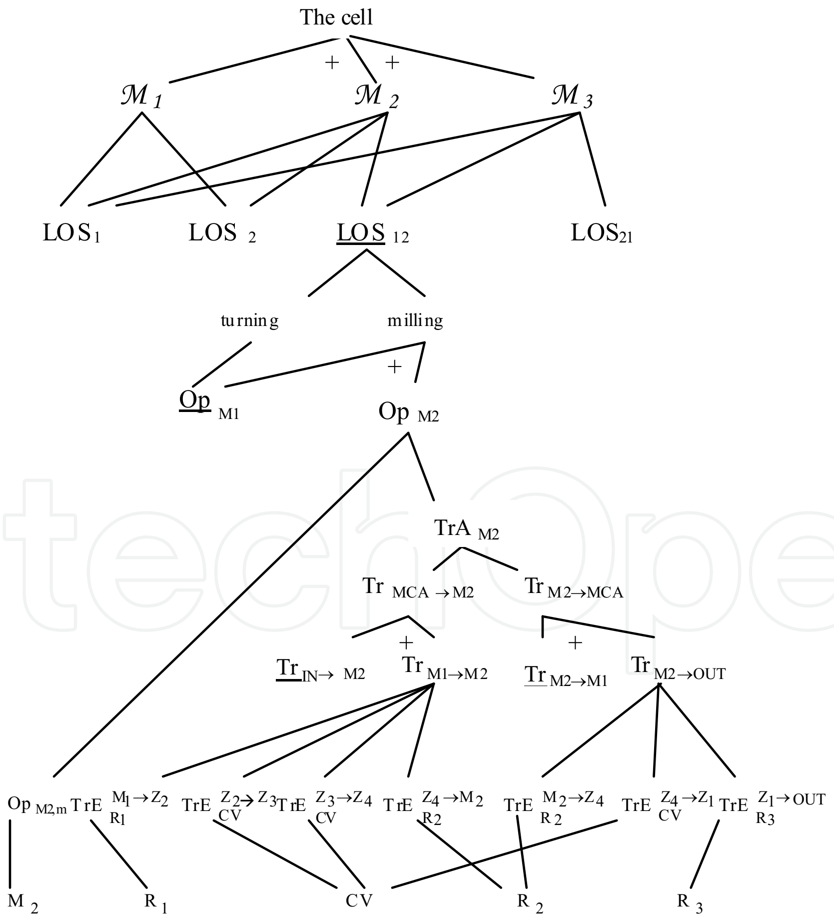

Fig. 2. The functional model of the machining cell

During operation the active functional model is a subset of the potential functional model. The current mission enables to define this subset.
The functional model (AND/OR graph) of the machining cell (Fig. 1) is represented in Fig. 2. The underlined entities are not developed on this figure.

In Fig. 2 the AND nodes do not have any notation, however OR nodes are denoted using +. OR nodes correspond either to an inclusive OR or an exclusive OR according to the constraints given in the functional requirements. For example, an exclusive OR is necessary for safety reason. For instance when we have two machining operations that utilize the same resource, and only one can utilize the resource at any time. The functions turning and milling are examples of entities linked with a logical AND to form LOS_{12}. The transfers $Tr_{IN \rightarrow M2}$ and $Tr_{M1 \rightarrow M2}$ are linked with a logical OR.

3.3 Behavior specification

Functional modeling is completed by behavioral specification. We propose in the following at first generic specifications representing the states of a subsystem (that we call an entity) or the whole system, then the specifications of the relationships between the entities of the model.

1. Specification of the behavior

For an entity or the whole system, states are divided into two board categories. The modes and operating parameters of the system, they are the states required by the operator and in fact he/she could select them. The degradations and the failures of the resources change the availability of the entities of the model, such states are unexpected.

By **required states**, we mean all the modes which are selected before setting up the production. There are one or more families of modes and the criteria of regrouping them into families vary from a system to another, and according to its complexity. The FMS behavior at each instant is then defined by the set of the simultaneously active states in each one of these families of modes. The initial situation corresponds to states initially selected in each family. After selecting the required modes, setting up the production is made through the activation of the selected mission. This involves activating the operations and the resources which take part in the selected mission. The missions introduced in the previous paragraph are now associated with a state transition graph called the graph of mission handling.

Definition 4. The graph of mission handling is the state transition graph whose states correspond to the missions of the FMS and the transitions represent the change of a mission to another.

The **unexpected states** concern the availability of the entities of the system; the monitoring function detects the failures of the resources. The change-of-state of the resources affects the elementary operations and therefore the machining functions, the LOSs, and the selected mission. All the entities of the model are thus characterized by their availability. Several states belonging to distinct families (including the selected mission) define the FMS current mode before setting up the production. We introduce now the current mode concept.

Definition 5. The current mode of FMS is the subset of states that are simultaneously active and belonging to distinct families. The selected mission belongs to this subset.

The entities of the functional model which take part in the selected mission can be activated if they are available. Thus two conditions are necessary to the activation of an entity: belonging to the current mode and being available.

Fig. 3 shows the states of an entity. The activation (deactivation) of an entity is represented by the working mode. The functioning mode represents the availability (unavailability) of an entity.

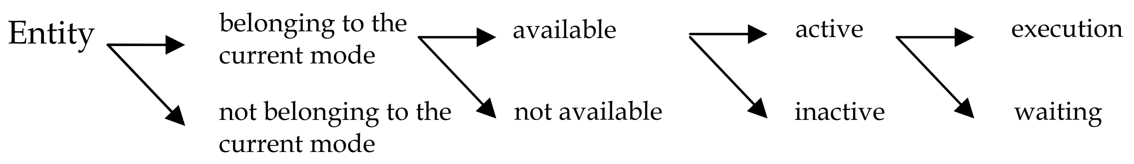

Fig. 3. The states of an entity

In the following the formal specification of the behavior using the synchronous formalism SSM is presented.

Formal specification using Safe State Machine: It is necessary to represent the activation (deactivation) at all the hierarchy layers of the functional model (see subsection 3.2) as well as the availability of the resources and the functions of the FMS.

The model should then handle concurrently the information flows downwards to transmit high level control and reconfiguration orders and upwards to follow up the reports and failures detected by the monitoring function. The reactivity needed for this bidirectional exchange of information, in addition with concurrency and preemption, requires a synchronous approach. Let us remind that synchronous languages were introduced to overcome the problems of modeling reactive systems (Halbawachs, 1993).

The visual synchronous formalism Safe State Machine (SSM or SyncCharts (Synchronous Charts) (André & Péraldi, 1993; André, 1996; André, 2003) is used to specify the behavior of the entities of the functional model. SSM inherits many features from Statecharts (Harel, 1987) but offers several forms of preemption and benefits of more strict semantics fully compatible with that of the synchronous language Esterel (Berry & Gonthier, 1992; Boussinot & De Simone, 1991). SSM supports also, with very rigorous semantics, hierarchy, communication, concurrency and various forms of preemption, which characterize our modeling approach. Instantaneous broadcast of signals is the unique communication medium. The presentation of the syntax of SSM is beyond the scope of this paper (see (André, 1996; André, 2003). In addition, SSM takes advantage of an industrial development environment (Esterel Studio™), which provides necessary tools to the development phases of systems, from specification to implementation. It enables also verifying and validating the specified models (Hamani et al., 2007).

The behavior of each entity of the model, whatever its level, is represented by a SSM including two modes (Fig. 4). At any time we know, if an entity, belonging to the current mode of the FMS, is in normal state e_N, degraded state e_D or out of order state e_OUT according to the functioning mode point of view; and if it is activated e_Active or inactive (in stopping state) e_OFF according to the working mode point of view. We take as initial states, the stopping state e_OFF for the working mode and the normal state e_N for the functioning mode.

There is a relationship between the two represented modes. Indeed, the activation of an entity e using start_e can be made only if the entity is available (i.e. it should not enter the out of order state). This is represented using 'start e and not e_OUT'. As soon as the entity is active, it enters the working state e_ON if the conditions of its activation are appropriate (in Fig. 4, there is no condition, we will see in the next subsection that such conditions depend on the relationship between the entities). The deactivation of the parent entity using stop_e depends also on the deactivation of its child entities (if they exist). The stopping state is effectively reached when the entity ends its activity. Indeed, when the entity reaches the final state (see the state End distinguished by its double outline), the macro-state e_Active is exited through the normal termination (whose tail is a small triangle). Notice that in presence of concurrency which characterizes the FMS model, using final states and normal termination is more suitable for synchronization (André, 2003).

According to the previous specification, the input and the output signals represented in Fig. 4 are the following: for working mode, the input signals start_e and stop_e correspond respectively to the activation and deactivation orders. They lead respectively entering e_ON and e_OFF states. For functioning mode, the input signals e_n, e_d and e_out provoke respectively entering in normal state e_N, degraded state e_D, and out of order state e_OUT. Local signals are also needed for specification. They are signals which are neither controllable nor directly observable from outside.

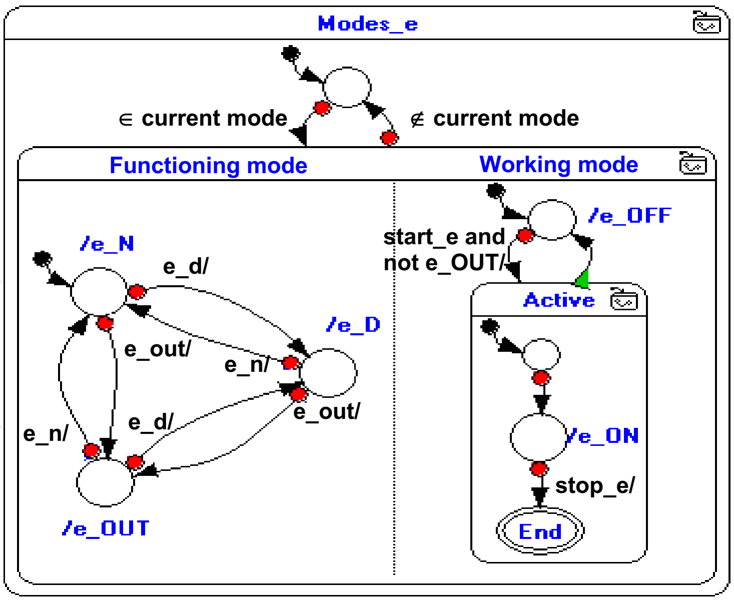

Fig. 4. The SSM representing an entity

2. Specification of the logical relationships

In the relation between a parent entity (level i) and a set of child entities (level i-1), the child entities are linked either with a logical AND or a logical OR. The information exchange between a couple of parent-child entities is given in the following.

- For working mode, the activation or deactivation request (using an input signal or a local signal) is a downward information flow from a parent entity to its child entities. The reports of such requests are propagated upwards.
- For functioning mode, failure events provoke the change-of-state of the resources i.e. the entities at the lower level of the model. For intermediate levels of the model this information flow is propagated upwards from the child entities to their parent entity.

The change-of-state of the entities of the model belonging to successive levels is carried out according to precise rules which depend on the kind of the logical relationship between the entities. If a parent entity has only one child then, for working mode, the activation (deactivation) of the parent entity provokes the activation (deactivation) of its child. The parent entity is effectively in execution if and only if the child entity is in execution. For functioning mode, mode changing of the child entity provokes mode changing of the parent entity.

In the case of a parent entity which has several child entities, state calculations corresponding to the case of two child entities (binary case) using an example of an entity e and its child entities e_1 and e_2 are given in (Hamani et al., 2007). The behavior is obtained using the interactive simulator XES (XEtented Simulator) (Esterel Studio™). For m child entities, the functioning mode is obtained in a recursive way. For working mode, the specifications are obtained in the same way as for the binary case.

3. Modular representation using reference models

The specification of any entity of the model differs according to whether it has one or more child entities or it has no child entity (the leaves of the graph). It depends also on the logical relationship between the child entities. Thus, we have defined reference models, a representation in SSM which is similar to the procedures (or functions) used in programming languages.

The representation of the deep hierarchy of the model as well as the I/O and local signals are taken into account. They are presented in the following paragraphs.

The representation of the model: We represent the model according to its hierarchy (see Fig. 2). The behavior of the entities is obtained by instantiation of the reference models. The behavioral models of the entities belonging to the same level of the hierarchy are orthogonal. The encapsulation mechanism offered by SSM allows linking two successive levels (level i and level i-1). The integration of single behavioral models should respect some communication rules. The main rules are presented in the following.

Communication rules: The relation parent-child defines the existing information exchange and the logical relationships taken into account. The I/O and local signals are declared for each behavioral model of an entity. Nevertheless, it is necessary to respect some rules for a coherent integration of the entire model.

- For functioning mode, only the resources i.e. the leaves of the model have input signals. Indeed, for functioning mode, mode changing is detected only for the resources and then propagated upwards.

- For working mode, only the entities of the top level of the model have input signals which correspond to the orders emitted at supervision level. The entities belonging to the other hierarchical levels of the model receive these requests which are propagated downwards thanks to local signals.
- All input signals used for selecting the configurations of the model (redundancies) should be declared. For example, for the model of Fig. 2, the input signals $C_{M2,m}$, $C_{M1,m}$ allow selecting milling operation on M_2 or on M_1 respectively.

The model (Fig. 2) is specified using 7 generic models and 43 instances of those generic models for nearly 1452 lines of Esterel code generated from the SSM models. This model declares 36 input signals and 212 output signals.

4. Carrying out FMS model handling

The functional model is characterized, on one hand, by functional relationships between the entities belonging to successive levels (vertical interactions); and on the other hand, by direct or indirect accessibility relations between the entities belonging to the same level (horizontal interactions).

Considering the behavior, each entity of this model is characterized by its functioning mode and its working mode. These two modes are constrained: the activation of an entity can be made only if it is available.

The availability of the entities which take part in the realization of the selected mission allows knowing if this mission is realizable or not. The missions we introduced represent the production goals that the system should perform, each function or resource which takes part in this realization must be available.

We explain now the behavior of the mode handler. In the exploitation phase, the behavioral model should handle the information flows downwards to transmit the orders from Human Machine Interface and upwards to propagate the information provided by the monitoring function. As shown in Fig. 5, the downward flow of high level control and reconfiguration orders is propagated through the LOSs corresponding to the selected mission. The flow is then propagated through the operations and the resources which take part in the realization of the selected mission. As for the downward flow, the upward flow transmits the results obtained by the execution of high level control and reconfiguration orders and process the information about the availability of the resources and the operations. Taking into account this information exchange in the control system, the communications of mode handling model with control and monitoring models should be specified.

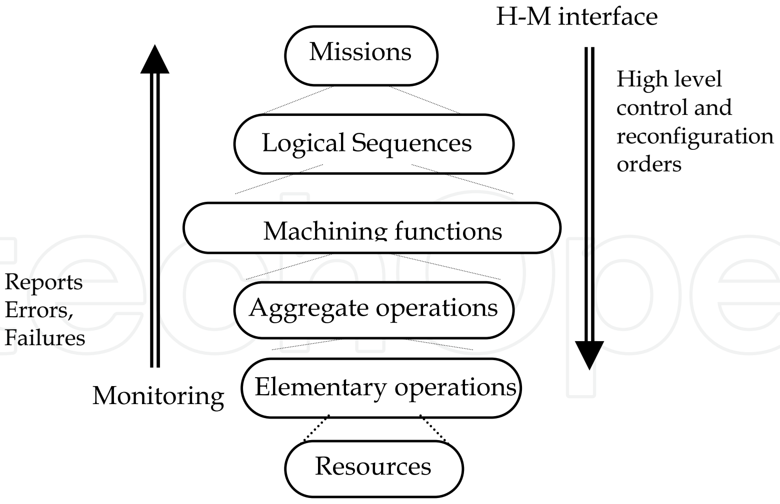

Fig. 5. The information flows

The mode handler is a reactive system (Harel, 1985) which continuously reacts to stimuli (inputs) of the environment by sending back other stimuli (outputs). Thus, mission and working mode changing are transmitted downwards. The change-of-state of functioning mode caused by the failures detected on the plant is transmitted upwards. These failures cause either the degradation (degraded state) or the breakdown (out of order state) of the resources.

Consider the reactions of a SSM associated with an entity of the model. Indeed, each entity e at level i can receive Either a request from level i+1 indicating a mission or a working mode change or a mode changing of one or more child entities of level i–1.

5. Conclusion

We have presented in this paper a new modeling method for FMS mode handling. We introduced the mission concept which corresponds to a subset of LOSs that the system carries out simultaneously. The idea is to decompose and identify the FMS functional entities which take part in the realization of its missions. The functional model is obtained by a modular and hierarchical decomposition leading to the elementary machining and transfer operations. The structural aspect completes this description by associating resources with the elementary operations they perform. The proposed modeling steps are illustrated through an example of a manufacturing cell. This modeling method is characterized by its generic concepts which allow their implementation in the computer aided tool CASPAIM_soft (Ndiaye et al., 2002).

The behavior of the obtained model is specified using the synchronous formalism SSM. Indeed, the strong hierarchy and the concurrency which characterize the functional model as well as the reactivity of information processing, justify using a synchronous approach. The behavior associated with the entities of the FMS model and the logical relationships (AND, XOR and OR) between them are also generic. An instantiation of the predefined models make it possible to obtain the model of a given system. Within the design phase, the reuse of functional entities and predefined behavioral models is one of the main characteristics of our approach. In addition, the behavior specifications using the synchronous formalism SSM enables to benefit fully from formal analysis tools (Hamani et al., 2007).

Within the exploitation phase, the role of the model is to handle the missions and the configurations. The information flows within the control system framework are provided both by high level control and monitoring function.

Further works aim at first to implement the introduced concepts in the information system CASPAIM_soft. Within the control system, we would like to complete the reaction loop in view of a failure by studying the integration and the communication between all those functions of supervision, monitoring and low level control. Therefore the model should be enhanced in order to allow this information exchange.

6. References

Amar, S.; Craye, E., Gentina, J.C. (1992). Une méthode hiérarchique de spécification et de prototypage des systèmes de production flexibles. *Revue Auto. Prod. Info. Indus*, 26, 5-6, (1992) 483-514.

André, C. & Péraldi, M.A. (1993). *Synchronous approach to industrial process control*, Technical Report RR 93-10, I3S, Sophia-Antipolis, France.

André, C. (1996). Representation and analysis of reactive behaviors: a synchronous approach, *Proceedings of IMACS Multiconference in Computational Engineering in Systems Applications (CESA 96)*, pp. 19-29, Lille, France.

André, C. (2003). *Semantics of Safe State Machines*, Guyancourt, France, http://www.esterel-technologies.com

Ausfelder, C.; Maik, J.P., Kermad, L., Gentina, J.C., Delfieu, D., Moisand, R., Sahraoui, A.E.K. (1993). Integration of operating modes in the control of flexible manufacturing systems combining synchronous and asynchronous approaches, *Proceedings of IEEE International conference on Systems Man and Cybernetics (SMC 93)*, pp. 234-239, Le Touquet, France.

Berruet, P.; Toguyéni, A.K.A., Craye, E. (2000). Towards implementation of recovery procedures for FMS supervision. *Comp. in industry*, 43, (2000) 227-236.

Berry, G. (1989). Real-time programming: general purpose or special purpose languages. *Info. Processing*, 89, (1989) 11-17.

Berry, G.; Gonthier, G. (1992). The Esterel synchronous programming language: design, semantics, implementation. *Science of Comp. Prog.*, 19, 2, (1992) 87-152.

Biland, P.; Deplanché, A.M., Elloy, J.P. (1994). A model for operating modes of computer integrated manufacturing systems, *Proceedings of the European workshop on Integrated Manufacturing Systems Engineering (IMSE 94)*, Grenoble, France.

Bois, S.; Craye, E., Gentina, J.C. (1992). Manager of working modes. *CIM Integ. Des. in Manufa.*, 21, 3, (1992) 201-207.

Boussinot, F. & De Simone, R. (1991). The Esterel language: another look at real time programming, *Proceedings of the IEEE*, pp. 1293-1304.

Cruette, D.; Bourey, J.P., Gentina, J.C. (1991). Hierarchical specification and validation of operating sequences in the context of FMS. *Comp. Integ. Manufa. Sys.*, 4, 3, (1991) 140-156.

Dangoumau, N. & Craye, E. (2001). Modes management for safe production systems, *Proceedings of the 10th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 01)*, Vienna, Austria.

Elkhattabi, S.; Craye, E., Gentina, J.C. (1995). Supervision by the behavior modeling, *Proceedings of the IEEE International Conference on Systems Man and Cybernetics (SMC 95)*, pp. 1416-1422, Canada.

Esterel Studio™ : http://www.esterel-technologies.com

Halbawachs, N. (1993). *Synchronous Programming of Reactive Systems*, Kluwer Academic Publishers, Amsterdam.

Hamani, N.; Dangoumau, N., Craye, E. (2009). Reactive mode handling of flexible manufacturing systems. *Mathematics and Computers in Simulation,* 79, (2009) 1421-1439.

Hamani, N.; Dangoumau, N., Craye, E. (2005). A comparative study of mode handling approaches, *Proceedings of the 35th International Conference on Computers & Industrial Engineering (CIE 05)*, Istanbul, Turkey.

Hamani, N.; Dangoumau, N., Craye, E. (2006). A functional modeling approach for FMS mode handling, *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 06)*, Saint Etienne, France.

Hamani, N.; Dangoumau, N., Craye, E. (2007). Safe State Machines for reactive mode handling of manufacturing systems, *Proceedings of the International Modeling and Simulation Multiconference (IMSM-IMAACA 07)*, Buenos Aires, Argentina.

Harel, D. (1985). On the development of reactive systems, *Proceedings of NATO Advanced Study Institute on Logics and Models for Verification and Specification of Concurrent Systems*, pp 477-498, NATO ASI series F, Vol. 13, Springer-Verlag.

Harel, D. (1987). StateCharts: a visual approach for complex systems. *Sc. of Comp. Prog.*, 8, 3, (1987) 231-275.

Kermad, L.; Craye, E., Bourey, J.P., Gentina, J.C. (1994). The exploitation modes of flexible manufacturing systems, *Proceedings of IEEE International conference on Systems Man and Cybernetics (SMC 94)*, pp. 1521-1526, San Francisco, USA.

Koren, Y.; Heisel, U., Moriwaki, T., Ulsoy, G., Van Brussel, H. (1999). Reconfigurable Manufacturing Systems. *Annals of the CIRP*, 48, 2, (1999) 527-540.

Ndiaye, D.; Bigand, M., Corbeel, D., Bourey, J.P. (2002). Information system for production engineering: contribution to maintaining consistency of composite data using an object oriented approach. *Int. J. of Comp. Integ. Manufa.*, 15, 3, (2002) 233-241.

Parayre, T.; Sallez, Y., Soenen, R. (1992). Model of the state of operation of automated systems, *Proceedings of the 8th International PROLAMAT conference (PROLAMAT 92)*, pp. 553-575, Tokyo, Japan,

Ranky, P. (1990). *Flexible Manufacturing Systems in CIM*, CIMware Ltd, Guildford, U.K.

Tawegoum, R.; Castelain, E., Gentina, J.C. (1994). Real time piloting of flexible manufacturing systems, *Euro. J. of Opera. Res.*, 78, (1994) 252-261.

Toguyéni, A.K.A.; Elkhattabi, S., Craye, E. (1996a). Functional and/or structural approach for the supervision of flexible manufacturing systems, *Proceedings of IMACS-IEEE multiconference on Computational Engineering in Systems Applications (CESA 96)*, pp. 716-721, Lille, France.

Toguyeni, A.K.A.; Craye, E., Gentina, J.C. (1996b). A framework to design a distributed diagnosis in Flexible Manufacturing Systems, *Proceedings of IEEE International conference on Systems, Man and Cybernetics (SMC 96)*, pp. 2774-2779, Beijing, China.

Toguyéni, A.K.A.; Berruet, P., Craye, E. (2003). Models and algorithms for failure diagnosis and recovery in flexible manufacturing systems, *Int. J. of Flexi. Manufa. Syst.*, 15, 1, (2003) 57-85.

Contemporary Robotics - Challenges and Solutions
Edited by A D Rodi

ISBN 978-953-307-038-4
Hard cover, 392 pages
Publisher InTech

Published in print edition December, 2009

This book book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of contemporary robotics and autonomous systems. The volume is organized in four thematic parts according to the main subjects, regarding the recent advances in the contemporary robotics. The first thematic topics of the book are devoted to the theoretical issues. This includes development of algorithms for automatic trajectory generation using redudancy resolution scheme, intelligent algorithms for robotic grasping, modelling approach for reactive mode handling of flexible manufacturing and design of an advanced controller for robot manipulators. The second part of the book deals with different aspects of robot calibration and sensing. This includes a geometric and treshold calibration of a multiple robotic line-vision system, robot-based inline 2D/3D quality monitoring using picture-giving and laser triangulation, and a study on prospective polymer composite materials for flexible tactile sensors. The third part addresses issues of mobile robots and multi-agent systems, including SLAM of mobile robots based on fusion of odometry and visual data, configuration of a localization system by a team of mobile robots, development of generic real-time motion controller for differential mobile robots, control of fuel cells of mobile robots, modelling of omni-directional wheeled-based robots, building of hunter- hybrid tracking environment, as well as design of a cooperative control in distributed population-based multi-agent approach. The fourth part presents recent approaches and results in humanoid and bioinspirative robotics. It deals with design of adaptive control of anthropomorphic biped gait, building of dynamic-based simulation for humanoid robot walking, building controller for perceptual motor control dynamics of humans and biomimetic approach to control mechatronic structure using smart materials.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Nadia Hamani and Abderahman El Mhamedi (2009). A Modeling Approach for Mode Handling of Flexible Manufacturing Systems, Contemporary Robotics - Challenges and Solutions, A D Rodi (Ed.), ISBN: 978-953-307-038-4, InTech, Available from: http://www.intechopen.com/books/contemporary-robotics-challenges-and-solutions/a-modeling-approach-for-mode-handling-of-flexible-manufacturing-systems

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China

www.intechopen.com

51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821