Equivalence of Behaviors Between Centralized and Multi-model Approaches

Gregory FARAUT, Laurent PIÉTRAC, Eric NIEL
Laboratoire Ampère, INSA-Lyon
Bât. St-Exupery, 20 av Albert Einstein
69621 Villeurbanne, France
First.Lastname@insa-lyon.fr

Abstract— **This paper presents a comparison between centralized approach and multi-model approach based on Supervisory Control Theory (SCT). The centralized approach uses both the whole process and specification to compute the controlled process. The multi-model approach, on the other hand, is used basing on various modal perspectives. This approach allows to build smaller models, which lead to smaller scale and better understanding of the latter. The comparison is made basing on a number of conditions, which all ensure the identical behavior between the controlled processes of each discussed approach. An example of a manufacturing system illustrating the comparison is also presented.**

I. INTRODUCTION

Discrete-event systems (DES) encompass a wide variety of physical systems such as manufacturing systems, supply chain networks, operating systems and communication systems. Some current problems in these systems are related to the verification of safety and liveness properties. Research evidence can be found in numerous works on model checking, such as [1], [2]. These research efforts study the properties of the controller model only after this latter has already been built. Meanwhile, the research works of [3]–[5], focus on formal approaches to ensure that properties are respected throughout the design cycle. The Supervisory Control Theory (SCT), as expounded by [6], is one of these approaches. Indeed, the SCT has significantly improved DES research area, particularly by introducing other properties such as controllability, observability and, more recently, diagnosability to design the controller in a formal manner.

However, major issues in SCT are complexity and scalability. In reality, systems would be rather large and represented with complex models, which are difficult to understand even if computation process of the system succeeds. As found in different approaches, to solve scalability of models, some conditions are needed. Examples include the hierarchical approach of [7]. This approach requires a condition of consistence. Modular approach, used in [8], [9], has an issue in confliction of supervisors. Last but not least, decentralized approach discussed in [10], [11], does not ensure the maximal permissive language and needs a study to choose the best rule: conjunctive or disjunctive computation. Furthermore, if these approaches reduce the scalability by using system decomposition when conditions are respected, they do not improve the comprehension of the models as they use both the whole process and the specification. An extension of these approaches concerns the concurrent discrete event systems (CDESs) [12], [13]. They consist of a large number of subsystems that operate concurrently. A subsystem is a small part of the global system. The events set of subsystems may be disjoint, in this case the subsystems operated concurrently and asynchronously, or they may partially overlap, yielding a degree of synchronization through the simultaneity of events. These works focus on the conditions to obtain the maximal admissible sublanguage by only computing the maximal admissible sublanguage of each subsystem and synchronizing them. However, a strong usual hypothesis is the shared events between subsystems which are controllable. Furthermore, the specifications are taken globally during the study, that leading to increase the complexity of it.

More recently, an approach based on modal standpoint with a control reconfiguration has been adopted in the studies of [14]–[16]. This approach aims to reduce the complexity and providing a better understanding of the model. Using the modal standpoint is common in industry to describe and design a system. It is worth mentioning that a mode is the behavior representing the used components and the requirements which have to be respected to perform the task. However, a system has neither to use all components nor to respect all the requirements. If the system needs to use any other component or has to respect any other requirement, it then has to change mode. On top of that, complexity increases drastically due to the fact that a system processes differently according to phase, mode or degraded situations. The mode management focuses on requirements necessary to ensure the switching between modes is possible, even if some requirements between modes are opposite.

In our previous work, presented in [17], [18], we proposed a framework using modal standpoint in the design of a system. The multi-model approach decomposes the system controller in several models representing a set of needed components to realize tasks such a set of requirements has to be respected. Each model, called mode, represents the expected behavior of the system and each mode is studied independently and separately. The framework enables the possibility of switching between modes and ensures the requirements of each mode to be respected. Furthermore, each model keeps a small size, that allows a better understanding of the latter.

More generally about SCT, whatever the chosen approach, a usual question is the relation about the computed behaviors between the classical centralized approach and the others. In this paper, we present a formal comparison to identify the conditions to ensure that the behavior computed by the multi-model approach is identical to the behavior computed by the centralized approach, *independently of uncontrollable events between modes*. In particular, the behavior in each mode is built with a subset of requirements, instead of all requirements like in centralized or concurrent approaches. Based on this equivalence, the design may be split to be made in parallel, keeping a good understanding of models and ensuring in the same time to computation gives the same behavior that if the design has been done by centralized approach, which can't be split.

This paper is structured as follows; the second section is an overview of the studied approaches. The comparison between the approaches is made in the third section. The fourth section illustrates an example, showing the conditions being respected.

II. Overview of the studied approaches

A. Supervisory Control and centralized approach

The SCT underpins the study of DES control. Initially, the SCT is based on language theory and uses automata to formalize a model of the system behavior. An automaton G is a 5-tuple such as $G = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is a set of the states, Σ is a set of events, $\delta : Q \times \Sigma^* \to Q$ is the extended transition function, q_0 is the initial state and $Q_m \subseteq Q$ is the set of marked states. $L(G)$ is the language generated by the automaton G written as $L(G) = \{s \in \Sigma^* | \delta(q_0, s) \; is \; defined\}$ and $L_m(G)$ is the marked language generated by G written as $L_m(G) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$. $L(G)$ represents the set of all possible trajectories i.e. all possible system behaviors, whereas $L_m(G)$ represents the subset of trajectories, leading to a marked state. Furthermore, in SCT, the set of events Σ is partitioned into two disjoint subsets Σ_c and Σ_{uc} which comprise controllable and uncontrollable events respectively.

The SCT, through the centralized approach, suggests a design of the uncontrolled process G_{cent} and the model of specification E_{cent}, which represents requirements to respect. With these models, a supervisor S_{cent} is adjoined to restrict the behavior of G_{cent} in a feedback manner. The restricted behavior represents the undesired behavior which does not respect the requirements. Nevertheless, the supervisor S_{cent} can only forbid controllable events. Formally, S_{cent} is a function defined by $S_{cent} : L(G_{cent}) \to 2^{\Sigma}_{cent}$. If the supervisor S_{cent} exists, i.e. there is a supervisor able to sufficiently forbid controllable events to obtain the desired behavior. Consequently, there exists a model H_{cent} representing the controlled process. H_{cent} generates a sub-behavior of G_{cent} as it respects the requirements imposed by E_{cent}. In this case, H_{cent} is controllable according to G_{cent}. Formally, $L_m(H_{cent}) = L_m(G_{cent} \times E_{cent})$. In the opposite, if H_{cent} is initially not controllable w.r.t. G_{cent}, due to uncontrollable events, the marked language generated by the controlled process H_{cent} will be the most permissive sub-language that respects the requirements and that is controllable according to the marked language generated by G_{cent}. Further discussion has been made in [19], [20]. As yet, $L_m(H_{cent}) = [L_m(G_{cent} \times E_{cent})]^{\uparrow c}$ where $\uparrow c$ is the function that computes the supremal controllable sub-language.

To sum up, in centralized approach, the controlled process H_{cent} is built with the process G_{cent}, representing the unrestricted behavior of the system, and the model of specification E_{cent}, representing the requirements to respect. In practice, the state space of the system grows exponentially and the model of specification are often complex.

B. Multi-model approach

A system is composed of a variety of components (machines, actuators, sensors, etc.) activated to perform and achieve a task in accordance to functional and safe requirements. Nonetheless, neither all components are used all time nor all requirements have to be respected.

In [18], the multi-model approach is used with a modal standpoint w.r.t. the followed definition:

***Definition** 1:* A mode is defined by a set of components and a set of specifications such it represents a temporary behavior of the system.

It means the system is considered by several modes in which a subset of components is used and a subset of requirements has to be respected. Taking a subset of components and specifications into account, instead of a whole set of these two, allows a better understanding of the models thanks to their smaller sizes.

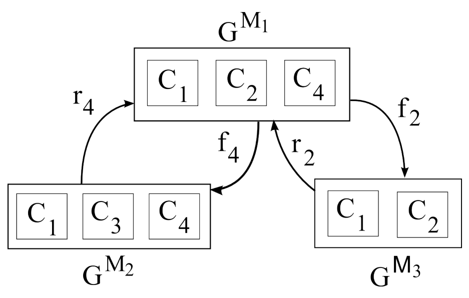
Formally, the set of components used in a system is denoted by $\mathcal{C} = \{C_1, C_2, \ldots, C_i\}$, where $i \in \mathbb{N}$ and $i \geq 1$. A component C_i is modeled by an automaton G^{C_i} where $G^{C_i} = (Q^{C_i}, \Sigma^{C_i}, \delta^{C_i}, q_0^{C_i}, Q_m^{C_i})$.

To have a modal standpoint, a mode is denoted M_j and the set of modes is denoted $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$, where $n \in \mathbb{N}$ and $n \geq 1$ (by convention, M_1 is the initial active mode). To represent the switch behavior of the system, the automaton $G^{\mathcal{M}}$ is defined. The language generated by this automaton represents the language of commutation, i.e. the requirements in relation to the switch control of the system. Further discussion is found in [18]. Furthermore, we define $\mathcal{C}^{M_j}$ as the set of components used in the mode M_j.

The Fig. 1 is an example of switching modes. We have three modes, M_1, M_2 and M_3. The components C_1, C_2 and C_4 are used in mode M_1. From this mode, a switching is possible to the mode M_2, by the event f_4 generated by the component C_4, or to M_3, by the event f_2 generated by the component C_2. Then, the mode M_3 is composed of the component C_1 and of the component C_2.

The process G^{M_j}, representing the behavior of the system in the mode M_j, is built by parallel composition of components used in the mode M_j. It is defined by:

$$G^{M_j} = G^{\mathcal{M}} || (||_{C_k \in \mathcal{C}^{M_j}} G^{C_k}). \tag{1}$$



$\mathcal{M} = \{M_1, M_2, M_3\}$
$\mathcal{C} = \{C_1, C_2, C_3, C_4\}$
$\mathcal{C}^{M_1} = \{C_1, C_2, C_4\}$
$\mathcal{C}^{M_2} = \{C_1, C_3, C_4\}$
$\mathcal{C}^{M_3} = \{C_1, C_2\}$

Fig. 1. Example of mode decomposition

The parallel composition is the classical composition of automata, like defined in [20].

The specification model E^{M_j} represents the requirements that have to be respected in the mode M_j. If E^{l,M_j} is one of these that has to be respected in the mode M_j, we define:

$$E^{M_j} = ||_{l\in\mathbb{N}} E^{l,M_j}. \tag{2}$$

Finally, the model H^{M_j}, representing the controlled process in the mode M_j, is built the same way as the model H_{cent} of centralized approach, but over the subset of components. These used in the mode M_j. $L_m(H^{M_j}) = L_m(G^{M_j} \times E^{M_j})$ if $L_m(H^{M_j})$ is controllable w.r.t. $L_m(G^{M_j})$, either way $L_m(H^{M_j}) = [L_m(G^{M_j} \times E^{M_j})]^{\uparrow c}$.

To summarize, instead of building one process and one specification, the multi-model approach built numerous processes G^{M_j} and specifications E^{M_j}, which compute the controlled processes H^{M_j} in the considered mode. Each H^{M_j} represents a temporary behavior of the system and taken both, they represent the full behavior.

The main difficulty in using the modal standpoint is to ensure all switching that happen in a mode lead effectively into another mode. This difficulty is not the subject of this paper, and do not talk about in the following. Nevertheless, we proposed a framework to solve it. more information can be found in [17], [18]. From now, the problem is about the behavior computed by the framework proposed in previous works and the behavior computed by the usual centralized approach.

III. Comparison between approaches

Our objective is to identify the conditions to have an equivalence of behaviors between the controlled process H, computed by centralized approach, and the full behavior, called H_{mod}, representing the behaviors generated by the both controlled process H^{M_j} built by multi-model approach.

This section focuses on the conditions to be respected in order to have this equivalence giving, at the end, the following theorem:

***Theorem** 1:* For $H_{mod} = ||_{M_j\in\mathcal{M}} H^{M_j}$, the behaviors computed by centralized and multi-model approaches are equivalent such $H_{cent} = H_{mod}$ if some conditions are respected.

As said, the controlled processes H^{M_j} represent a temporary behavior of the system. Then, use the parallel composition to build the controlled process H_{mod}, representing the full behavior, is coherent, in particular because the system is in only one mode at each instant.

A. Equivalence of controlled processes

Regarding the centralized approach, H_{cent} is usually defined by $H_{cent} = (G_{cent} \times E_{cent})$. In the multi-model approach, H^{M_j} is defined by $H^{M_j} = G^{M_j} \times E^{M_j}$. In [20], the authors pointed out if both automata used in product have the same set of events, then the product operation is equivalent to the parallel operation, i.e. $G \times E = G||E$.

Furthermore, to have a complete equivalence of behaviors between the controlled process H_{cent} and H_{mod}, the next hypothesis needs to be true:

***Hypothesis** 1:* The controlled process H^{M_j} is controllable to respect the process G^{M_j}.

Indeed, whatever the approach, in the case where the controlled process is not controllable to respect the process, the maximal permissive behavior is looking for, i.e. $H = [G||E]^{\uparrow c}$. In this case, we have: $L_m(H) = [L_m(G||E)]^{\uparrow c} \subseteq L_m(G||E)$.
This restriction on multi-model approach leads to a look for the maximal permissive behavior on each controlled processes H^{M_j} in modes M_j such as $L_m(H^{M_j}) = [L_m(G^{M_j}||E^{M_j})]^{\uparrow c} \subseteq L_m(G^{M_j}||E^{M_j})$.

Finally, if the hypothesis is not true, this is the next equation that should be taken into account:

$$H^{M_j} \subseteq (||_{M_j\in\mathcal{M}} G^{M_j})||(||_{M_j\in\mathcal{M}} E^{M_j}) \tag{3}$$

The equation (3) implies, like in the decentralized approach, that it is not possible to ensure the maximal behavior.

With the equation $H_{mod} = ||_{M_j\in\mathcal{M}} H^{M_j}$, and the hypothesis (1), the equation of the theorem (1) can be expressed as follows:

$$G_{cent}||E_{cent} = ||_{M_j\in\mathcal{M}}(G^{M_j}||E^{M_j}) \tag{4}$$

Due to the associative and commutative properties of the parallel composition, then the equation (4) can be written under this form:

$$G_{cent}||E_{cent} = (||_{M_j\in\mathcal{M}} G^{M_j})||(||_{M_j\in\mathcal{M}} E^{M_j}) \tag{5}$$

Then, by identification, to have an equivalence on controlled processes between approaches, the next lemmas have to be true :

***Lemma** 1:* $G_{cent} = ||_{M_j\in\mathcal{M}} G^{M_j}$

This lemma means the behaviors of the process G, built in centralized approach, and the composition parallel of processes G^{M_j}, built in multi-model approach, are equivalent.

***Lemma** 2:* $E_{cent} = ||_{M_j \in \mathcal{M}} E^{M_j}$

This lemma means the restrictions, representing the specifications and generating a behavior, in centralized and multi-model approach are equivalent.

B. Equivalence of Processes

In this section, lemma (1) is proved.

In the centralized approach, the process G_{cent} is built by parallel composition of all automata representing a behavior, i.e. by all components and, in the case that interest us, by the automata representing the modal standpoint. Then, the process G_{cent} is computed by the following equation:

$$G_{cent} = G^{\mathcal{M}} || (||_{C_k \in \mathcal{C}} G^{C_k}) \tag{6}$$

In the multi-model approach, G^{M_j}, representing the process in the considered mode, is computed by equation (1) as explained in section 2.2.

With the last hypothesis, the next proposition can be made:

***Proposition** 1:* Let G, the process built by centralized approach, given by equation (6), and let G^{M_j} be defined in multi-model approach: $G^{M_j} = G^{\mathcal{M}} || (||_{C_k \in \mathcal{C}^{M_j}} G^{C_k})$

Then, the process G has the same behavior that the parallel composition of all the processes in the modes, i.e. the next equation is true:

$$G = ||_{M_j \in \mathcal{M}} G^{M_j} \tag{7}$$

Then, the lemma (1) is proved with the properties (commutative, associative, etc.) of the parallel composition, the equation (6) and from the equation (1).

$$\begin{aligned}
& G^{M_j} = G^{\mathcal{M}} || (||_{C_k \in \mathcal{C}^{M_j}} G^{C_k}) \\
\Leftrightarrow \quad & ||_{M_j \in \mathcal{M}} G^{M_j} = ||_{M_j \in \mathcal{M}} (G^{\mathcal{M}} || (||_{C_k \in \mathcal{C}^{M_j}} G^{C_k})) \\
\Leftrightarrow \quad & ||_{M_j \in \mathcal{M}} G^{M_j} = (||_{M_j \in \mathcal{M}} G^{\mathcal{M}}) || (||_{M_j \in \mathcal{M}} (||_{C_k \in \mathcal{C}^{M_j}} G^{C_k}))
\end{aligned}$$

but

$$\begin{aligned}
||_{M_j \in \mathcal{M}} G^{\mathcal{M}} &= ||^n G^{\mathcal{M}} \\
&= (\underbrace{G^{\mathcal{M}} || G^{\mathcal{M}} || \ldots || G^{\mathcal{M}}}_{\text{n times}}) \\
&= G^{\mathcal{M}}
\end{aligned}$$

and $||_{M_j \in \mathcal{M}} (||_{C_k \in \mathcal{C}^{M_j}} G^{C_k}) = ||_{C_k \in \mathcal{C}} G^{C_k}$

then

$$||_{M_j \in \mathcal{M}} G^{M_j} = G^{\mathcal{M}} || (||_{C_k \in \mathcal{C}} G^{C_k})$$

$$||_{M_j \in \mathcal{M}} G^{M_j} = G_{cent}$$

Thus far, if each component is used at least once in a set of components in mode, i.e. if $\mathcal{C} = \bigcup_{M_j \in \mathcal{M}} \mathcal{C}^{M_j}$ is true, then the parallel composition of all these components, even if a component is used many times, is equivalent to the process built by centralized approach.

C. Equivalence of specifications

In this section, lemma (2) is proved.

In practice, the system designer builds one model of specification according to requirements to be respected. There are usually two kinds of requirements. These requirements are defined as follows:

***Definition** 2:* A *permanent* requirement is a requirement that has to be respected whatever the situation or happens in the system.

The permanent requirements concern for example safety requirements. This kind of requirement is denoted by the model of specifications E_{perm}.

***Definition** 3:* A *temporary* requirement is a requirement that has to be respected in some particular situation, and for a limited time.

The temporary requirements concern for example all tasks the system has to achieve in a mode. When the system switches to another mode, it has not to respect these requirements anymore, because each mode has its own particular requirements. This kind of requirement is represented by the model of specifications E_{temp}.

In centralized approach, from these definitions, the model of specifications E_{cent} is built by the next equation:

$$E_{cent} = E_{perm} || E_{temp} \tag{8}$$

In multi-model approach, we also consider a permanent specification to be respected whatever the activated mode, while a temporary specification belongs to a particular mode, or many modes but not all. However, in the considered approach, the temporary requirement can be designed by smaller models of specification than in the centralized approach because they have only to be adapted for the considered mode. These models are denoted $E^{M_j}_{temp}$.

Then, E_{perm} is used in each mode to build the specification E^{M_j}, and $E^{M_j}_{temp}$ is the only specification that has to be respected in the mode M_j (and not in others). Then, E^{M_j} is computed by:

$$E^{M_j} = E_{perm} || E^{M_j}_{temp} \tag{9}$$

From equation (8) and equation (9), the lemma (2) may be developed to be proved as follows:

$$\begin{aligned}
E_{cent} &= ||_{M_j \in \mathcal{M}} E^{M_j} \\
\Leftrightarrow \quad E_{perm} || E_{temp} &= ||_{M_j \in \mathcal{M}} (E_{perm} || E^{M_j}_{temp}) \\
\Leftrightarrow \quad &= (||_{M_j \in \mathcal{M}} E_{perm}) || (||_{M_j \in \mathcal{M}} E^{M_j}_{temp})
\end{aligned}$$

It is obviously true that $E_{perm} = ||_{M_j \in \mathcal{M}} E_{perm}$, because the parallel composition of the same model computes this model. Then, lemma (2) is true if the next condition is verified:

$$E_{temp} = ||_{M_j \in \mathcal{M}} E^{M_j}_{temp} \tag{10}$$

Thus, the discussion only focuses on specifications that are not common between modes, i.e. the temporary specifications $E^{M_j}_{temp}$.

The theorem is true if the condition defined by equation (10) is respected.

In centralized approach, the system designer builds the specification E_{temp}. As said, this model may be very difficult to design due to numerous specifications to be respected at different period of time. This difficulty increases if some specifications between modes are opposite. The designer has to take into account the switching behavior in this model. Furthermore, this can be extremely complex if the switching events are uncontrollable. In this case, the system designer has to be careful and not to forget any commutative and uncontrollable event that might occur. If the designer has the opportunity to decompose this model into numerous smaller models, without changing the result, keeping a good understanding on these smaller models, and being able now to compute models in same time, then the design will be shorter.

To conclude, the theorem is proposed in final form:

Theorem 2: For $H_{mod} = ||_{M_j \in \mathcal{M}} H^{M_j}$, the behaviors computed by centralized and multi-model approaches are equivalent such $H_{cent} = H_{mod}$ if $E_{temp} = ||_{M_j \in \mathcal{M}} E_{temp}^{M_j}$

IV. Example

In this section, an example of manufacturing system is presented to illustrate a modal standpoint by using a multi-model approach. In particular, the design is done w.r.t lemma (2).

A. Requirements

The studied system, illustrated in Fig. 2.(a), is composed of three components. Initially, an item arrives on the conveyor C_{conv}, illustrated by Fig. 2.(b). The conveyor transfers the item to a machine C_1, illustrated by Fig. 2.(c). When the first machine has finished, the conveyor moves the item to the (next) machine C_2, illustrated by Fig. 2.(d).

The system has four modes. The first one is the *Initial* mode. When the system is initialized, modeled by the event *Init_Ack*, the system switches to the *Cycle* mode. The system works in this mode to produce items until an uncontrollable event is generated - event *Stop_Req*. From this event, the system switches to the *End_Cycle* mode. In this mode, the system has to finish its cycle, and generates the event *Stop_Req_ok*, before switches to the *Stop* mode. Before stop the cycle, the system produces the last item without transferring a new one on the conveyor. The switching behavior, representing by $G^{\mathcal{M}}$, is illustrated by the Fig. 2.(e).

A safety requirement forbids the conveyor to be activated while the both machines 1 and 2 are working. This forbidden case results in damage to the system, and potentially injuring the technician. The safety requirement is represented by the model of specification E_{safety} and is illustrated by Fig. 3.(a). This safety requirement is a *permanent* requirement.

In each mode, the system has to perform a task. The Fig. 3.(b) represents a liveness requirement. This specification concerns all the tasks that the system has to perform regards to the considered mode. This model, called $E_{liveness}$, has been designed by the system designer in regards to the desired behavior in the centralized approach.

As illustrated, the specification $E_{liveness}$ in centralized approach is not obvious to design. In particular, the designer has to study what the system has to do in each mode moreover where and how a switching to other modes can happen. An example is given by the uncontrollable event *Stop_Req*. Where may it happen and what happens after that ? The aim is thus to show the advantage of a modal standpoint, by using a multi-model approach, if the condition is verified.

B. Equivalent behaviors

The focus targets the comparison between the liveness specification designed in the centralized approach $E_{liveness}$, and liveness specifications designed in the multi-model approach.

To recall, the system has three components, then $\mathcal{C} = \{C_{conv}, C_1, C_2\}$. It also has four modes, then $\mathcal{M} = \{Init, Cy, End\_Cy, Stop\}$.

By using the multi-model approach, four processes are built - one for each mode - by parallel composition of the mode automaton and the used component in the considered mode.

$$\begin{aligned} G^{Init} &= G^{\mathcal{M}} || G^{C_{conv}} || G^{C_1} \\ G^{Cy} &= G^{\mathcal{M}} || G^{C_{conv}} || G^{C_1} || G^{C_2} \\ G^{End\_Cy} &= G^{\mathcal{M}} || G^{C_{conv}} || G^{C_1} || G^{C_2} \\ G^{Stop} &= G^{\mathcal{M}} || G^{C_{conv}} || G^{C_1} \end{aligned}$$

In this example, the modes G^{Cy} and G^{End_Cy} use all components of the system, like the process G in centralized approach. This is the worst case for us, nevertheless, the fact is the system taken in example is relatively simple.

From these models, the system designer builds a model of liveness specification $E_{liveness}^{M_j}$ representing, in each mode, the requirement the system has to respect to achieve the task in the considered mode. These specifications of mode are illustrated in Fig. 4.

On each model, the system designer can focus only on what is the local desired behavior and without taking into account other specifications. The local desired behavior in the *initial* mode is illustrated Fig. 4.(a). Fig. 4.(b) represents the requirement to perform the task of producing items. The system designer does not consider in this model what happens if the uncontrollable event *Stop_Req* occurs. In this case, the model is smaller, and less complex. The Fig. 4.(c) represents the requirement to finish a cycle before switching to *stop* mode. This requirement is easy to design, and used with the latter Fig. 4.(b) allows to generate the behavior when the uncontrollable event happens. Finally, the last Fig. 4.(d) represents the desired behavior to stop the system.

To sum up, the models of specification of E^{M_j} are build as follows:

$$\begin{aligned} E^{Init} &= E_{safety} || E_{liveness}^{Init} \\ E^{Cy} &= E_{safety} || E_{liveness}^{Cy} \\ E^{End\_Cy} &= E_{safety} || E_{liveness}^{l,End\_Cy} || E_{liveness}^{Cy} \\ E^{Stop} &= E_{safety} || E_{liveness}^{Stop} \end{aligned}$$

These models of specification have been designed w.r.t. theorem (2). Then, the designer is sure the behavior generated by the parallel composition of the controlled processes H^{M_j}, according to $E^{M_j} = E_{safety} || E_{liveness}^{M_j}$, is equivalent to the behavior generated by the controlled process H_{cent} synthesized by centralized approach. Furthermore, the models in multi-model approach are smaller, helping to detect mistakes of design and to have a better comprehension of these models.

V. Conclusion

We presented a comparison between the centralized and multi-model approaches. The multi-model approach is based

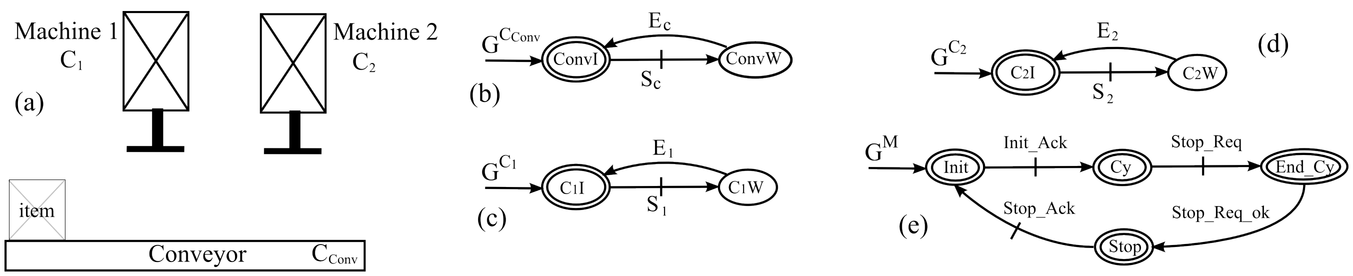


Fig. 2. Manufacturing system example : (a) the studied system; (b,c,d) process of components G^{C_i}; (e) mode automaton $G^{\mathcal{M}}$.

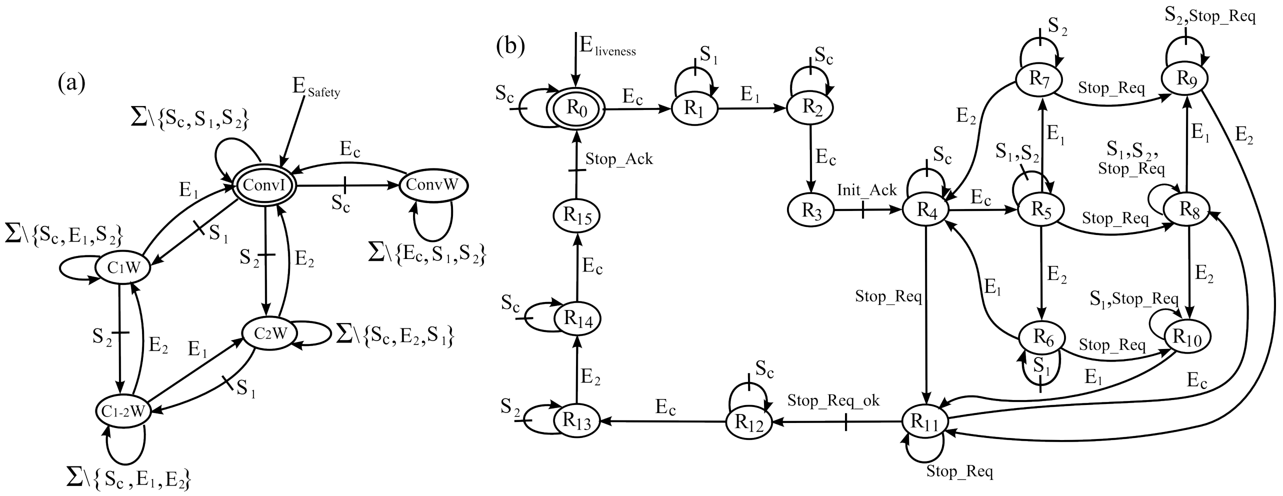


Fig. 3. Manufacturing system example : (a) Safety specification E_{safety}. This specification avoids to use the conveyor and machines in the same time; (b) Liveness specification $E_{liveness}$. This specification includes all behaviors to perform the desired task in regards to the current system mode.

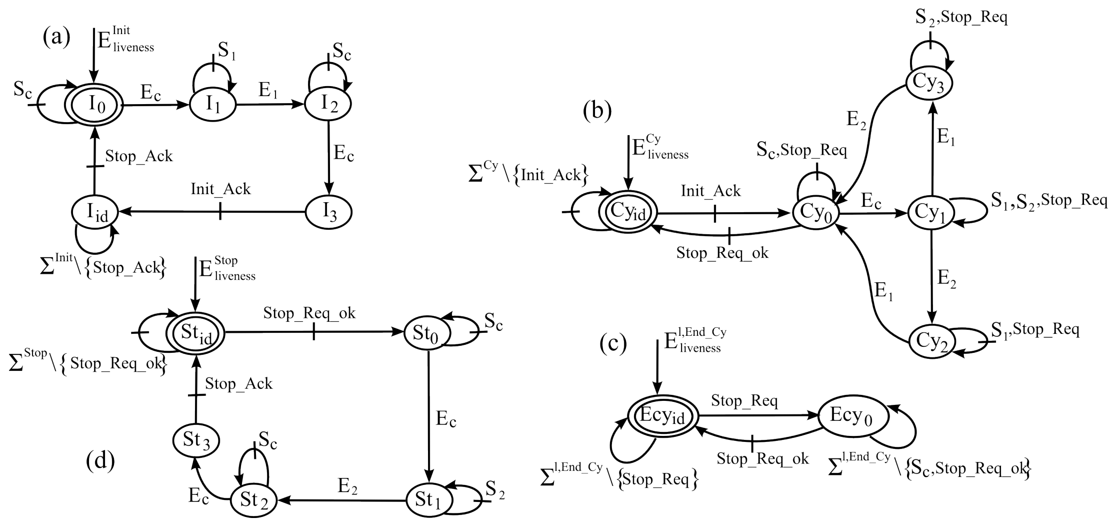


Fig. 4. Liveness specification $E^{M_j}_{liveness}$: (a) specification of the *Initial* mode; (b) specification of the *Cycle* mode to produce items; (c) specification in addition of (b) to finish cycle; (d) specification to stop the system (mode *Stop*).

on the decomposition of the system into numerous sub-systems representing a particular behavior. Each sub-system has its own properties to be respected.

In this paper, we prove under the respect of a condition concerning the temporary requirements, the behavior computed by the parallel composition of the behavior of each mode is equivalent to the behavior computed by the centralized approach. The advantage of the modal standpoint is in the decomposition such each mode has its own requirements to be respected and uses subset of all components. Then, the designer may split the design to focus on each functioning part of the system, independently of others. Furthermore, he keeps a good understanding in working on smaller models.

Inspired by the multi-model approach, we are currently working on a generic model with an objective to ensure that the parallel composition of the model of specification is identical to the one of the centralized behavior.

References

[1] A. Chutinan and B. Krogh, "Computational techniques for hybrid system verification," *Automatic Control, IEEE Transactions on*, vol. 48, no. 1, pp. 64 – 75, jan 2003.

[2] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986– 1001, July 2003.

[3] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, June 1987. [Online]. Available: citeseer.ist.psu.edu/harel87statecharts.html

[4] K. McMillan, "Symbolic model checking: an approach to the state explosion problem," Ph.D. dissertation, Carnegie Mellon University, 1992.

[5] R. Alur, T. A. Henzinger, and P.-H. Ho, "Automatic symbolic verification of embedded systems," in *IEEE Real-Time Systems Symposium*, 1993, pp. 2–11. [Online]. Available: citeseer.ist.psu.edu/alur96automatic.html

[6] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, Jan 1989.

[7] R. J. Leduc, B. A. Brandin, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory control-part i: serial case," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1322–1335, Sept. 2005.

[8] M. Nourelfath and E. Niel, "Modular supervisory control of an experimental manufacturing system," *Control Engineering Pratice*, vol. 12, no. 2, pp. 205–216, February 2004.

[9] J. Komenda, J. van Schuppen, B. Gaudin, and H. Marchand, "Supervisory control of modular systems with global specification languages," *Automatica*, vol. 44, no. 4, pp. 1127–1134, Apr. 2008.

[10] S. Jiang and R. Kumar, "Decentralized control of discrete event systems with specializations to local control and concurrent systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 30, no. 5, pp. 653–660, October 2000.

[11] G. Barrett and S. Lafortune, "Decentralized supervisory control with communicating controllers," *IEEE Transactions on Automatic Control*, vol. 45, no. 9, pp. 1620–1638, 2000.

[12] Y. Willner and M. Heymann, "Supervisory control of concurrent discrete-event systems," *International Journal of Control*, vol. 54, pp. 1143–1169, 1991.

[13] K. Schmidt, H. Marchand, and B. Gaudin, "Modular and decentralized supervisory control of concurrent discrete event systems using reduced system models," jul. 2006, pp. 149 –154.

[14] K. Andersson, J. Richardson, B. Lennartson, and M. Fabian, "Synthesis of hierarchical and distributed control functions for multi-product manufacturing cells," in *Automation Science and Engineering, 2006. CASE '06. IEEE International Conference on*, Oct. 2006, pp. 325–330.

[15] O. Kamach, L. Pietrac, and E. Niel, "Design of switching supervisors for reactive class discrete event systems," *Information Control Problems in Manufacturing 2006*, vol. 12, no. 1, pp. 289–294, 2006.

[16] R. Sampath, H. Darabi, U. Buy, and L. Jing, "Control reconfiguration of discrete event systems with dynamic control specifications," *Automation Science and Engineering, IEEE Transactions on*, vol. 5, no. 1, pp. 84 –100, Jan. 2008.

[17] G. Faraut, L. Piétrac, and E. Niel, "A new framework for mode switching in sct," in *European Control Conference 2009 - ECC09*, August 2009.

[18] G. Faraut, L. Pietrac, and E. Niel, "Formal approach to multimodal control design: Application to mode switching," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 4, pp. 443–453, Nov. 2009.

[19] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM Journal of Control and Optimization*, vol. 25, no. 3, pp. 637–659, 1987.

[20] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems [Second Edition]*, C. G. Cassandras and S. Lafortune, Eds. Springer, 2007.