CONTRIBUTION TO THE GRAFCET FORMALISATION A PROPOSAL FOR A STATIC META-MODEL

Florent Couffin, Sandrine Lampérière, Jean-Marc Faure
E-mail: Jean-Marc.Faure@toulon.ismcm-cesti.fr

Laboratoire d'Ingénierie Intégrée des Systèmes Industriels (LIISI)
CESTI - ISMCM Toulon
Place Georges Pompidou - Quartier Mayol
83000 TOULON - FRANCE

Phone: (33) (0)4.94.03.88.00
Fax: (33) (0)4.94.03.88.04

ABSTRACT. This paper presents a meta-model of the Grafcet tool the aim of which is to improve its formalisation. This meta-model is described with an enriched Entity - Relationship formalism. After having described this formalism, we show how to build the meta-model. The Grafcet basic concepts and the links between them are detailed. This meta-model enables to formalise the Grafcet syntax and to fill the ambiguities due to the use of the natural language in the standards and reference books describing the Grafcet. It can also be a basic model for the data structure of tools supporting Grafcet design.

RESUME. Cet article présente un méta-modèle du Grafcet dont l'objet est d'en améliorer la formalisation. Ce méta-modèle est décrit au moyen d'un formalisme Entité - Relation enrichi. Après avoir décrit ce formalisme, nous montrons comment construire le méta-modèle visé. Nous mettons en évidence les concepts de base du Grafcet et les liens entre ces concepts. Ce méta-modèle permet de formaliser précisément la syntaxe du Grafcet et d'éliminer les imprécisions dues à la description en langage naturel utilisé dans les normes et ouvrages de référence décrivant le Grafcet. Il peut également servir de base pour la structure de données d'outils d'assistance à une conception utilisant le Grafcet.

KEY WORDS. Meta-modelling, Grafcet, Automation Engineering, Entity - Relationship, Constraints.

MOTS-CLÉS. Méta-modélisation, Grafcet, Génie Automatique, Entité - Relation, Contraintes.

1 INTRODUCTION

The Grafcet is a tool for the modelling of logical system behaviour. Two French standards [UTE 93], [AFN 95] and an international one [IEC 91] define in a textual shape elements of the Grafcet syntax and describe its evolution rules ; some examples of these rules are represented on schemata showing the evolution of some typical grafcets in different cases.

Unfortunately each one of these standards includes some lacks and ambiguities in the definition of the concepts (identification of a transition, links between steps and actions, ...). Furthermore, the scope of these standards (number of concepts and accuracy of concepts definitions) is not the same, which is an other source of ambiguity. This situation induce different interpretations of this modelling tool, which is contrary to the wishes of the original Grafcet designers and harmful to an approach of integrated engineering of automated systems.

From our point of view, these lacks and ambiguities come mainly from the only textual definition of concepts. Some attempts have been made in order to solve these problems by using formalisms enabling accurate definitions of the Grafcet concepts [LHO 93]. We intend to try and improve the Grafcet formalism by proposing more formal definitions elaborated with meta-modelling techniques. This meta-modelling process is useful in Automation Engineering and Software Engineering, fields in which a correct formalisation of the functional, informational and behavioural modelling tools is necessary [DEN 89], [MOR 91], [DEN 92], [STE 93], [REV 95], [PIE 96].

A meta-model of the Grafcet tool enables a precise definition of the syntax, that means the set of the concepts and the links between these concepts, and a description of the evolution rules. In the first case, we try and elaborate a static meta-model ; in the second case, a dynamic meta-model, taking into account evolutions, is searched. In this paper, the presentation is limited to a static meta-model ; for a global definition of the Grafcet, this meta-model has to be completed by the formal definition of the evolution rules, as described for example in [AFC 83], [FRA 87], [BON 94], [LHO 94], [BIE 96].

In order to precise the vocabulary, we first give the definition of *meta-model, Model* and *model*, some examples are given for the Grafcet tool. We call *Model* a set, a priori consistent and complete, of concepts, association rules of these concepts and interpretation rules. A *model* is a structure built by association of occurrences of the Model concepts ; the interpretation rules described in the Model enable the construction and the reading of the models. In our case, the Model is the Grafcet, with its syntax and its evolution rules (enabling the models interpretation). Every grafcet is a model of the wished behaviour of a logical system defined previously.

A *meta-model* is a representation in a suitable formalism of the concepts and rules of the Model. This formalism replaces the natural language to define the *discourse universe* ; in our case, the discourse universe is the Grafcet Model. So a meta-model is a model of the Model.

In the meta-modelling approach, the choice of the formalism used to describe the meta-model is important. This formalism must be accurate enough and have an expression power sufficient to describe the structural characteristics of the Model.

In this paper we introduce the formalism chosen to express the meta-model ; then we show in what way this formalism enables to improve the definition of the Grafcet concepts by building step by step its meta-model. The conclusion indicates the possible applications of the work and the evolution perspectives.

2 THE MODELLING TOOL USED FOR THE DESCRIPTION OF THE META-MODEL

The meta-model built in this paper is a conceptual data model described with a modelling tool named OMD-GA for "Outil de Modélisation des Données pour le Génie Automatique", what means Data Modelling Tool for Automation Engineering [LAM 94].

This tool is based on the Entity - Relationship formalism [CHE 76]. The aim of that kind of modelling tool is the representation of a discourse universe regardless of any implementation constraint.

Since the origin of the modelling tool [BAS 89], [PER 89], the *object* way has been let apart because the model designers was only interested in the structural aspect of the systems ; they did not consider the dynamic description.

Nevertheless some object modelling constructs have been introduced to answer the Automation Engineering specific needs. The modelling tool OMD-GA is an extension of the basic Entity - Relationship formalism ; the extensions consist in the specialisation - generalisation and a particular relationship type for identification: the aggregation relationship type. Within the framework of the BASE-PTA model standardisation [AFN 96], other extensions stemming from the studies of some formalisms of the same type [HAB 88], [ROC 88], [MOR 94], especially constraints on roles and on relationships, and explicit constraints have been introduced. These constraints improve the expression power of the original modelling tool.

2.1 Basic concepts

The modelling tool is based on the concepts of entity type and relationship type [CHE 76]. An entity type gathers the entities (elements of the discourse universe) of the same nature and having the same properties. A relationship type gathers the relationships (semantic links between the entities) of the same nature and having the same properties ; these links are specified by roles. The cardinalities limit the number of times an occurrence of an entity type can be linked to a given relationship type. Moreover the entity types and the relationship types are characterised by attributes representing their elementary properties. Among the attributes associated to an entity type, one or several of them characterise that an occurrence of the entity type can only appear once. These attributes are then called *identifiers* (there are preceded by a #).

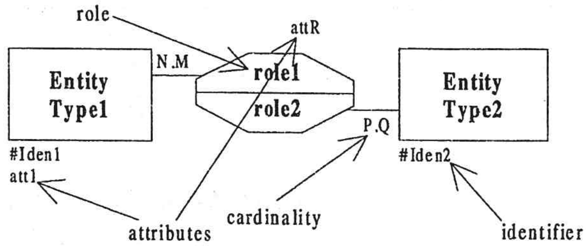

Figure 1. Representation of the basic concepts of the modelling tool

Note:

The relationship types are sometimes given a reference *Rxx* to make easier their notation.

2.2 Extensions

Only the extensions used in the meta-model are described.

2.2.1 Constraints on roles

A constraint on roles expresses existence conditions of an occurrence of an entity type with regards to its participation to occurrences of several relationship types.

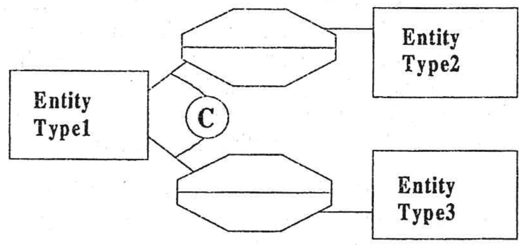

Figure 2. Representation of a constraint C on roles

Symbol C	Name	Definition
(T)	Totality constraint on roles	Each occurrence of the Entity Type1 must be linked to at least one occurrence of one of the two considered relationship types.
(X)	Exclusion constraint on roles	Each occurrence of the Entity Type1 can be linked to either one occurrence of the relationship type with the Entity Type2, or to one occurrence of the relationship type with the Entity Type3, but not to both of them.

2.2.2 Constraints on relationships

A constraint on relationships expresses existence conditions of occurrences of relationship types, what means the existence conditions of a couple of occurrences of entity types with regards to its participation to occurrences of several relationship types.

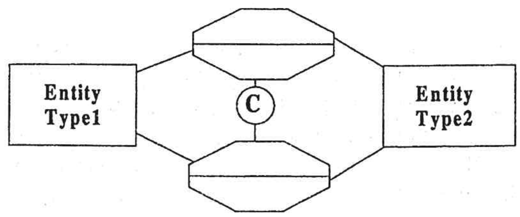

Figure 3. Representation of a constraint C on relationships

Symbol C	Designation	Definition
(X)	Exclusion constraint on relationships	Each couple of occurrences of entity types (Entity Type1, Entity Type2) can not be linked to an occurrence of both relationship types.

2.2.3 Aggregation relationship type

The aggregation relationship type is a particular relationship type enabling to specify that an entity type is identified by another one. The identified entity type is called aggregated entity type, the identifying entity type is called aggregating entity type. The *relative identifier* term replaces the absolute identifier term. Each occurrence of the aggregated entity type (Entity Type2) is identified by an occurrence of the aggregating entity type (Entity Type1) and by its relative identifier *IdenR2* (see Figure 4).

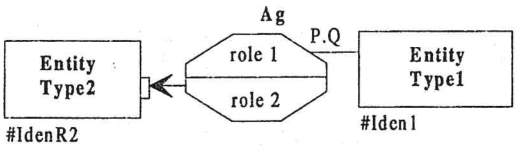

Figure 4. Representation of an aggregation relationship type

2.2.4 Explicit constraints

As for constraints on roles or cardinalities, the explicit constraints affect occurrences and aim to specify some properties of the discourse universe that can not be represented with other constraints. The defined constraints are described with a declarative language based on the predicate calculus [KLE 71].

We describe more precisely the constraints enabling to formalise *scope rules* (manipulation rules) between occurrences of entity types.

Let us introduce a function labelled $\Re$ the definition set of which is the set of the couples (Ri, e) defined on the Cartesian product of the set of the relationship types and the union of the parts (within the meaning of the theory of sets) of the occurrences of each entity type, so that e is a subset of the occurrences of one of the two entity types (called entity type source) linked to the relationship type Ri. $\Re$ gets a couple (Ri, e) and sends back the set of the occurrences of the other linked entity type (called entity type target) of the relationship type Ri, these occurrences being linked to an element of e by an occurrence of Ri.

For example, let us formalise the scope rules between the elements of a program (see Figure 5). Our aim is to specify that a function (modelled by the entity type FCT) can only manipulate (modelled by the relationship type R3) a software variable (modelled by the entity type SV) if this function is defined in (modelled by the relationship type R2) the program (modelled by the entity type PRG) in which the variable is declared (modelled by the relationship type R1). We can note that this scope rule is a condition of membership of a set, what is often found.

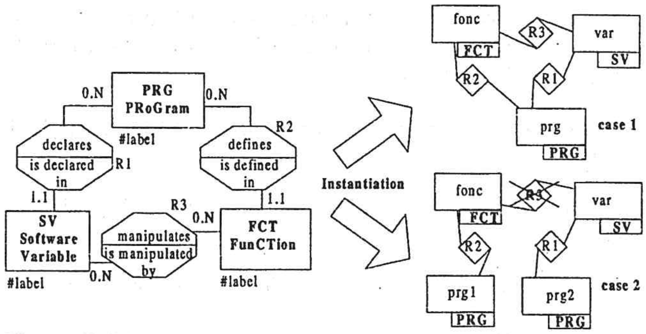

Figure 5. Model and instanced models: examples of explicit constraints

The different elements represented on the instanced model (Figure 5) are:

the occurrences of entity types represented by:

a big rectangle containing the values of the identifiers,

a small rectangle (on the bottom right of the big rectangle) containing the abbreviation of the instanced entity type,

the occurrences of relationship types represented by:

a diamond containing the reference (Rxx) of the instanced relationship type,

the arcs linking these occurrences to the occurrences of the source and target entity types.

To specify that the relationship type R3 can only be instanced between two occurrences of the entity types FCT and SV if these occurrences are linked to a same occurrence of the type entity PRG respectively by an occurrence of the relationship type R2 (for FCT) and by an occurrence of the relationship type R1 (for SV), an explicit constraint is defined:

$\forall$ x $\in$ FCT, $\forall$ y $\in$ SV $\bullet$ (R3(x, y)) $\Rightarrow$ ($\Re$(R2, {x}) = $\Re$(R1, {y})) in which:

- $\in$ means "denotes an occurrence of the type" and $\bullet$ means "so that",
- the predicate R3(x, y) denotes the existence of an occurrence of the relationship type R3,
- the equality in the second predicate is an equality between sets.

Let us apply this constraint to these two occurrences of a model (see Figure 5):

- case 1: $\Re$(R2, {(FCT, fonc)}) = {(PRG, prg)} = $\Re$(R1, {(SV, var)}), the proposition corresponding to the second predicate is true, so the relationship type R3 can be instanced between the occurrences of entity types (FCT, fonc) and (SV, var).
- case 2: ($\Re$(R2, {(FCT, fonc)}) = {(PRG, prg1)}) $\neq$ ($\Re$(R1, {(SV, var)}) = {(PRG, prg2)}), the proposition corresponding to the second predicate is false, the

relationship type R3 can not be instanced between the occurrences of entity types (FCT, fonc) and (SV, var).

Notes:

As an occurrence of an entity type is identified by the couple (label of the type, value of the identifier of the occurrence), the function $\Re$ is written down:

$\Re$(reference, {(abbreviation, occurrence_identifier_value)}), in which:

reference denotes the reference of the considered relationship type,

abbreviation denotes the abbreviation of the considered entity type,

occurrence_identifier_value denotes a value of the identifier of the occurrences of the entity type.

Moreover a shortened notation can be used in the case of aggregated entity types to write down the relative identification of their occurrences:

<occurrence_identifier> ::= { <aggregating_entity_type_label> '.' <aggregating_entity_type_identifier> '.' } <relative_identifier>

3 PRESENTATION OF THE GRAFCET META-MODEL

3.1 Method and hypotheses

The meta-model aims at describing the **Grafcet** discourse universe ; the following steps, usually used during the conception of a conceptual data model, are proposed:

definition of the entity types,

choice of an identification mechanism for each entity type (absolute or relative identification),

definition of the simple relationship types,

definition of the constraints on roles and on relationships, and of the explicit constraints,

definition of the attributes (non identifiers) of the entity types and of the relationship types.

The proposed meta-model is a data conceptual model of a grafcet in its finale version, what means that the meta-model does not represent a grafcet during its design.

The meta-model does not depend on any treatment. To ensure its consistency, we must check that every object (entity type, relationship type, attribute, ...) can not be deducted from any treatment of other objects. This ensures information are not duplicated during the creation of the models, what could introduce inconsistency during these models update.

3.2 Entity types

3.2.1 Objectives

The relevant entity types for a grafcet description are defined in this chapter, these are all the objects involved in an activity of specification or design of a logical system, what implies that every entity type must have a label, a definition and an identification mechanism. The first two items have been pointed out from the mentioned standards and from the book [BOU 92a].

3.2.2 Definition of the Grafcet concepts

A **global grafcet** (GG) describes the behaviour of a logical system.

A **step** (ST) characterises an invariant behaviour of a part or of the whole considered logical system.

An **action** (AC) is associated to a step and defines what must be done each time the associated step is activated.

A **transition** (TR) expresses the evolution ability between several steps by following a directed link. This evolution is realised by the clearing of the transition what implies a change in the step activity.

A **receptivity** (RE), also called transition condition, is a logical proposition associated to a transition ; it can either be true or false.

A **connected grafcet** (CG) is a grafcet (what means here a set of steps and transitions) so that it always exists a continuous path between two elements (step or transition). We will show latter in this paper that the definition of this concept is too weak (according to us) and can be discussed.

A **partial grafcet** (PG) is defined as a set of several connected grafcets. As we disagree with the notion of connected grafcet, too *topological* and *not enough structuring*, we choose to define in this paper the partial grafcet as a subset of a global grafcet whose aim is the description of the behaviour of a logical subsystem ; so that we provide the partial grafcet with a structuring nature.

A **macro-step** (MS) is the only representation of an unique set of steps and transitions, this set is called expansion of the macro-step.

An **external variable** (EV) is a Boolean variable characterising the system which behaviour is described by the global grafcet. These variables correspond to the system inputs and outputs.

An **internal variable** (IV) is a Boolean variable characterising the state (activity) of a step. This limitation of the notion of internal variable to the only variable of step can be seen as restrictive, but it is in accordance with the one of [AND 94].

3.3 Identification mechanisms

3.3.1 Objectives

The designer of a model (and more especially of a grafcet, in this paper) has to manipulate objects, so he must be able to point out without ambiguity each of these objects. Moreover this identification of each object must be consistent with the usual rules or conventions (the model must be as *accurate* as possible to the object it represents). That is the reason why the choice of an identification mechanism is of a great importance. The chapter concerning the modelling tool explains that an entity type could either have an absolute identifier, or a relative one (modelled by an aggregation relationship type). Let us define an *identification hierarchy* for the Grafcet objects.

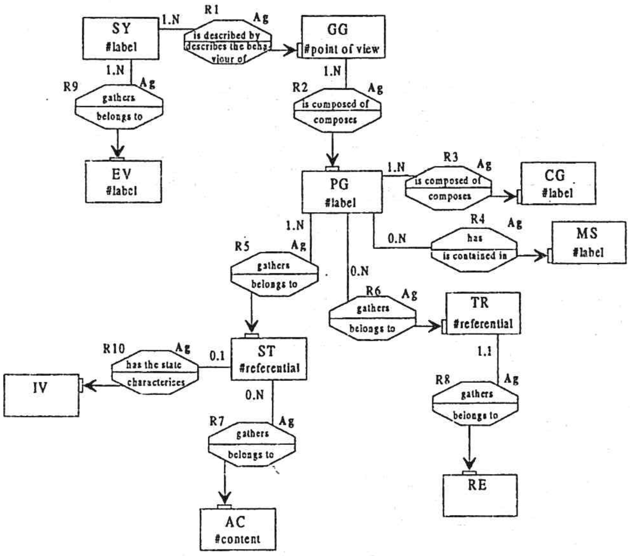

Figure 6. ***Identification of the entity types***

The following chapters deal with the problem of the identification of the entity types and demonstrate that some definitions of the grafcet model are still, according to us, inaccurate and even non-existent.

3.3.2 Global grafcet and system

By definition a global grafcet describes the behaviour of a logical system, it must be identified with regards to this system. Moreover this grafcet corresponds to a particular point of view on the system (aim of the grafcet designer), for example *grafcet of the wished behaviour of the machine 100.* So the entity type global grafcet is identified through an aggregation relationship type by an entity type **system** (SY). This entity type represents the system which behaviour is described. The relative identifier of the entity type global grafcet is the point of view of the grafcet designer (command, working ways, ...).

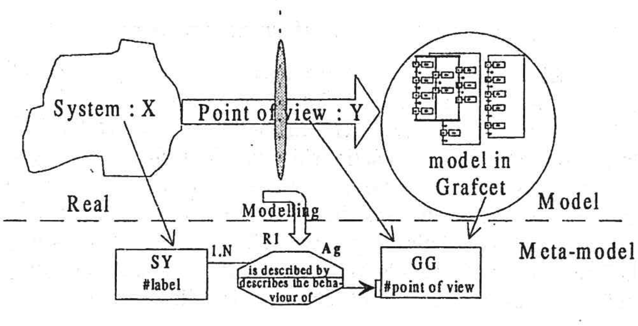

Figure 7. Identification of the global grafcet

3.3.3 Partial grafcet

Every partial grafcet is defined with regards to the global grafcet to which it belongs ; that means to the set of the occurrences of the entity type partial grafcet. Two partial grafcets can have the same label if they do not belong to the same global grafcet. Then the entity type partial grafcet is aggregated to the entity type global grafcet. This constraint is important since the partial grafcet is in an explicit way defined with regards to the global grafcet, and no more as a gathering of connected grafcets. We express this constraint through an aggregation relationship type. It is quite interesting to note that this point of view is not inconsistent with the definition of the partial grafcet as a way to describe the behaviour with a given structuration criterion.

3.3.4 Connected grafcet

In accordance to the structuration principle, the connected grafcet is defined as a component of a partial grafcet. The entity type connected grafcet is aggregated to the entity type partial grafcet. The relative identifier of the entity type connected grafcet can be the result of a particular structuration, for example the label of a *task* [FRO 93]. Nevertheless as the meaning of this relative identifier is difficult to define, it is just replaced by a label.

3.3.5 Macro-step

The same reasoning as for the entity type connected grafcet is available for the identification of the entity type macro-step. Nevertheless in practical uses macro-steps are designed by an alphanumerical referential *Mxx* to *shorten* their meaning. These referentials (designations) are bijective to their meaning, for example, *M3: Building-up of a set of rods.* So the referential and its meaning are equivalent in a formal point of view. The meaning is considered as the label (identifier) of the entity type macro-step.

3.3.6 Step and transition

The identification of the entity type step could be done relative to either the entity type connected grafcet or the partial grafcet one. As the step characterises the whole system or a part of it, we choose to aggregate the entity type step to the entity type partial grafcet and the numerical referential of the step is the relative identifier. By the same the transitions characterise the evolution of the whole considered system or a part of it ; so the entity type transition is aggregated to the entity type partial grafcet.

To distinguish an initial step, one just needs to add a Boolean attribute *initial* to the entity type step.

3.3.7 Action and receptivity

An action (respectively a receptivity), for example *put out jack*, can occur several times in a same partial grafcet ; the actions are associated to the steps (respectively the receptivities to the transitions). In order to distinguish the actions (respectively the receptivities) the entity type action is aggregated to the entity type step (respectively the entity type receptivity to the entity type transition). The entity type action can be identified by its content ; so a step can not be linked to several equal actions (the standards and books do not explain this). As only one receptivity can be associated to a transition (cardinality 1.1), the receptivity has no specific identifier. Formally the entity types receptivity and transition could be gathered in one entity type.

To insure the proposed model is consistent, the content of the receptivities and actions are to be formalised ; the Conway diagrams or the Backus-Naur Form could be used.

3.3.8 External variable

The external variable characterises the described system ; so it is identified by it. The internal variable characterises the state of a step. So the entity type external variable is aggregated to the entity type system and the entity type internal variable is identified with regards to the entity type step. In the case of the entity type external variable, its label is its relative identifier ; the entity type internal variable and the entity type receptivity have no specific identifier. Their label is sometimes given but is formally useless (in the model we propose it is only a reference, as it is for the content of the macro-step).

3.3.9 Use of the identification of the main entity types

To give an example, the following schema shows a part of a global grafcet (command of the workshop WS1) and the corresponding occurrences schema. The decomposition of the second partial grafcet (machine tool2) in occurrences of the entity types PG, ST, TR, AC and RE, and the identification of each occurrence are described. In this occurrences schema, the occurrences of the entity types are referenced by the abbreviation of the label of their type and by its identifier (or its content for the type receptivity). The occurrences of the relationship types are referenced by the label of their type and by the occurrences of linked entity types.

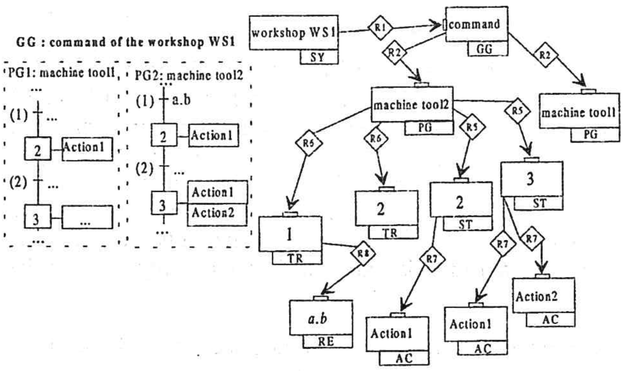

Figure 8. *Example of identification*

3.4 Simple relationship types

The definition of the simple relationship types enables to express the communication links between grafcet objects, that is the sharing of data (by variables) and the dependencies (explicit commands and inter-dependencies of state).

3.4.1 Intra- connected grafcet communication

The proposed meta-model is not yet complete. Even if each step and each transition are completely identified, we can not define what step follows what transition and what transition follows what step. The grafcet sequential structure is not described. The links representing the grafcet connectivity (expressed by the structure of the connected grafcets) have to be defined. These links characterise the links intra- connected grafcet. The defined relationship types (R11 and R12) are called *communication intra- connected grafcet* relationship types. They characterise the directed links step - transition. The cardinalities 0.N mean that a step can come before (can follow) 0 or several transitions (and the same for the transitions with regards to the steps).

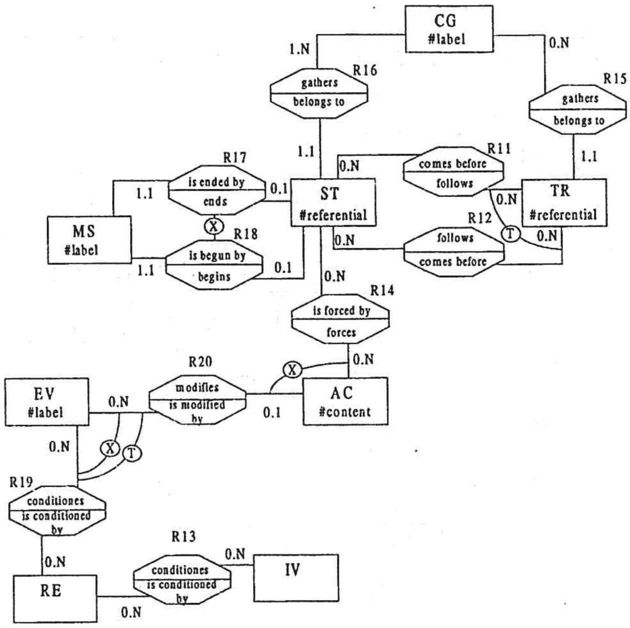

Figure 9. Communication between the entity types

Use of the relationship types R11 and R12

To give an example, the following schema (see Figure 10) shows parts of grafcets whose occurrences schemata are typical structures [BOU 92a].

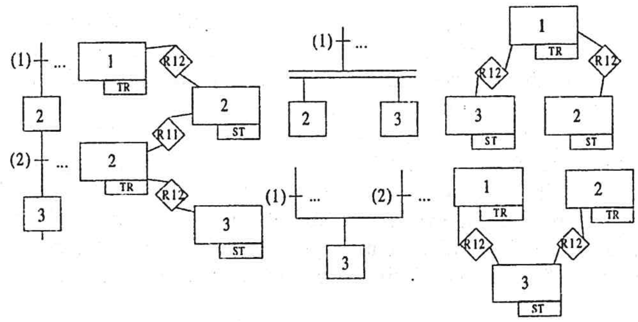

Figure 10. *Examples of typical structures*

These schemata describe occurrences of the entity types ST and TR, and occurrences of the relationship types R11 and R12. The relationship type R12 characterises the directed link transition - step and the relationship type R11 characterises the directed link step - transition. To describe the alternation step - transition that characterises the execution command, one only needs to use the relationship type R12 for a transition coming before a step and the relationship type R11 for a step coming before a transition. This linear structure is called *sequence*.

The *AND* divergence enables to define that sequential treatments (sequences) are done simultaneously. It is characterised by a transition coming before several steps ; the relationship type R12 is used to model this structure. The *OR* convergence is characterised by a step following several transitions. The relationship type R12 is used to model this structure.

Note:

The representations of the AND convergence and of the OR divergence are modelled by using the relationship type R11.

3.4.2 Inter- connected grafcets and inter- partial grafcets communication

By the same, let us define the inter- connected grafcets links which communication mechanisms have been studied in [LHO 93]. A first case describes the conditioning of a transition by a step belonging to an other connected grafcet (interpreted parallelism). This kind of communication can be modelled by a relationship type linking the entity type internal variable and the entity type receptivity.

The second case of inter- connected grafcets links is the forcing command. This important concept enables to define activation step priorities between partial grafcets of a same global grafcet. A forcing command has priority on any other evolution rule and imposes a given situation to the partial grafcet. This communication is modelled by a relationship type between the entity type action and the entity type step.

The relationship type R13 is called *inter- connected grafcets communication* relationship type and the relationship type R14 is called *inter- partial grafcets communication* relationship type. We can note that the forcing command links can be deduced from the occurrences of the relationship type R14 between the partial grafcets by using the established identification mechanisms.

3.4.3 Inter- global grafcets communication

The other relationship types modelling communication (R19 and R20) concern the external variables ; these variables can appear in the receptivities they condition (the variables are then inputs of the system) or are modified by the actions (the variables are then outputs of the system). These two communication ways are modelled through two relationship types. These relationship types are called *inter- global grafcets communication* relationship types. In fact they theoretically enable the communication between global grafcets through the modelled system.

3.4.4 Other structures

The other relationship types are defined to enable the construction of the structures of the connected grafcets and of the macro-steps. A connected grafcet gathers a set of steps and transitions (R15 and R16). A macro-step can be characterised by the ending step and the beginning step of its expansion (R17 and R18), a bijection exists between the macro-step and its expansion.

3.5 Improvement of the semantic of the meta-model through constraints

This chapter describes several examples of the use of constraints (on roles, on relationships and explicit ones) in order to complete the meta-model.

3.5.1 Non isolated transition

A transition can not be both shaft (no step following) and source (no step coming before). This rule is modelled by a totality constraint on roles between the entity type transition and the roles it is linked to in the two relationship types R11 and R12. By the same, a step can not be both the beginning step and the ending step of a same macro-step. In order to model this rule an exclusion constraint on relationships is defined between the two relationship types R17 and R18. An action can not both be an internal forcing action and modify an external variable. So an exclusion constraint on roles is added between the entity type action and the roles it is linked to in the two relationship types R14 and R20.

3.5.2 External variables partition

Each external variable is either an input or an output of the modelled system, but never both. This restriction can be modelled by two constraints on roles (exclusion and totality) between the entity type external variable and the roles it is linked to in the two relationship types R19 and R20. We can note that an occurrence of external variable is an output if it is linked to an occurrence of R20 and is an input if it is linked to an occurrence of R19.

3.5.3 Forcing and scope rule

The meta-model can be improved by defining the nature of the *inter- grafcets communications*. A *hierarchy of communication* and a *hierarchy identification* have been introduced. The explicit constraints are of a great interest to model the *scope rules* between the grafcet elements. The scope is defined as the ability of an object to manipulate another one. In the particular case of forcing, an action can only force steps belonging to the same global grafcet. This rule can be modelled through an explicit constraint:

$$\forall x \in AC, \forall y \in ST \bullet (R14(x, y)) \Rightarrow (\Re(R2, \Re(R5, \{y\})) = \Re(R2, \Re(R5, \Re(R7, \{x\}))))$$

Example of the constraint concerning forcing

The following example describes the forcing of a partial grafcet by another partial grafcet.

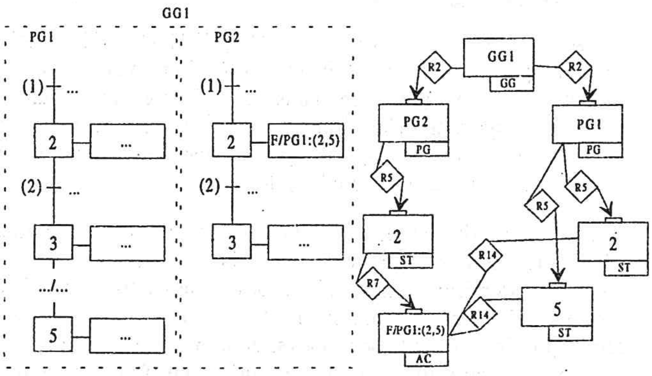

Figure 11. *Example of forcing*

The evaluation of the forcing condition leads to:

$\Re$(R2, ($\Re$(R5, {(ST, GG1.PG1.2)}))) = {(GG, GG1)} et

$\Re$(R2, ($\Re$(R5, ($\Re$(R7, {(AC, GG1.PG2.2.F/PG1: (2, 5))}))))) = {(GG, GG1)}

We can conclude that the occurrence of the relationship type R14 (forcing) between the occurrence of the entity type action F/PG1: (2,5) and the occurrence of the entity type step 2 of the partial grafcet 1 is allowed (it is the same for the occurrence of the entity type step 5 of the partial grafcet 1).

Note:

Every scope rule can be modelled with that kind of constraint. In the framework of a Language Inspired of the Grafcet meta-modelling [FAU 92], other cases have been studied and modelled [COU 93].

3.6 Comments on the resulting model

The resulting model enables to point out every Grafcet object. It also enables to rebuild the topological links between the different objects (links step - transition) and the hierarchical links coming from the communication between the different levels.

The Grafcet syntactical rules (for example the step - transition alternation) are checked during the construction ; the partition of the objects is ensured by the relative identification mechanisms. The set of the steps and the transitions builds a partition of a partial grafcet, the set of the partial grafcets builds a partition of a global grafcet, ...

The notion of partial grafcet has been preferred to the notion of connected grafcet because of a nowadays perception of the connected grafcets more topological than structuring described in the reference books [BOU 92a], [UTE 93]. If the definition of the connected grafcet notion is given with some more structuration, as the designer wishes it, it would modify the meta-model by introducing one more identification level. The steps, transitions and macro-steps would then have been identified with regards to the connected grafcet.

Moreover some additional attributes can be thought of in order to include some Grafcet extensions [BOU 92b] or in the case of the Languages Inspired of the Grafcet.

4 CONCLUSIONS

For several years, many works [LES 94], [LHO 94], [COU 97], for example, have been dedicated to the improvement of the models representation formalisms used during the design of production systems. The Grafcet is one of these formalisms. This paper is an attempt to give better definitions of its concepts by using meta-modelling techniques.

In fact this work enables to point out some lacks or ambiguities of the standards on which the work is based (existence of a transition identification, definition of the step activity variable, scope of the forcing commands, for example) and proposes some solutions to fill these lacks. Moreover this meta-model enables to ensure the consistency during any future evolution of the Model. Every new concept must be integrated to the existing meta-model.

Concerning the possible applications the proposed model can be the base for a data model for a grafcet editor or be used to design the structure of exchange files between softwares supporting the activities concerned by the data described in the meta-model, these applications come from the principles of the integration by means of data [FRA 93], [DID 95], [LAM 95].

Concerning the integrated engineering of production systems [CHA 96], the Grafcet meta-model can be linked to other formalisms (IDEF0, SA-RT, ...) [COU 96], [FAU 96] or to conceptual data models describing the materialisation of an application (materiel components, software elements,...). In the second case a link with a reference model of the command part, for example the one of BASE-PTA standard [AFN 96], can easily be defined through the entity types modelling behaviour.

An important work is still to be done, the following perspectives can be thought of:

- the taking into account of the Grafcet dynamic aspect (evolution rules)

The object modelling frameworks can then provide attractive solutions since they allow to describe from a conceptual data model the states of the entity types and the communications between these entity types.

- the taking into account of the external semantic brought by the designer during the building of a grafcet

It is usually admitted that the structure of a particular grafcet brings information on the application structure (working ways, command flexibility, ...) that are not included in the Grafcet formalism. The modelling of these underlying concepts implying structuration ways would be of a great improvement.

Finally in the aim of a better dissemination of the Grafcet, we think it would be of a great interest if a meta-model of the formalism was given in the future standards, what would avoid the different interpretations brought by incomplete definitions.

5 THANKS

The authors of this paper have been for several years members of the AFCET workgroup GRAFCET. The described work has been presented to the workgroup and we thank its members for their comments and advices.

Bibliography

[AFC 83] : AFCET, Logic Systems Group. *Interprétations algébriques et algorithmiques du Grafcet*. Le Nouvel Automatisme Review. September 1983.

[AFN 95] : French standard NF C 03-190. *Diagramme fonctionnel GRAFCET pour la description des systèmes logiques de commande*. September 1995.

[AFN 96] : *Représentation des systèmes de contrôle et de commande des systèmes automatisés*. AFNOR standard NF Z 68-901. 1996.

[AND 94] : C. ANDRE, D. GAFFE. *Evénements et conditions en GRAFCET*. Automatic Control Production Systems Review, AFCET/HERMES, vol. 28 n° 4. p. 331-352. 1994.

[BAS 89] : *BASE-PTA V 3.1*. Reference document, fascicles 0 to 4, Centre Coopératif de Génie Automatique (CCGA). January 1989.

[BIE 96] : E. BIEREL, O. DOUCHIN, P. LHOSTE. *Le Grafcet : du jeu à la réalité*. Proceedings of the AFCET'96 Congress : Reactive systems modelling. Brest, France. p. 123-130. March 1996.

[BON 94] : E. BON-BIEREL. *Méta-Modèles du Grafcet*. Research Report, Automated Production DEA, Nancy I University. September 1994.

[BOU 92a] : N. BOUTEILLE, P. BRARD, G. COLOMBARI, N. COTAINA, D. RICHET. *Le GRAFCET*. Cépaduès

Publishers, collection automatisation et production. March 1992.

[BOU 92b] : N. BOUTEILLE. *L'extension du modèle GRAFCET à un plus large domaine est-elle nécessaire ?*. Proceedings of the GRAFCET'92 Congress : Theory and applications - Grafcet. p. 3-12. Paris, France. March 1992.

[CHA 96] : V. CHAPURLAT, F. PRUNET. *Un modèle pour la spécification, la conception et la validation des systèmes de contrôle/commande répartis : le modèle ACSY-R*. European Journal of Automation, AFCET/HERMES, vol. 30 n° 1. p. 25-62. 1996.

[CHE 76] : P. CHEN. *The entity-relationship model : towards a unified view of data*. ACM transactions on database systems, vol. 1 n° 1. March 1976.

[COU 93] : F. COUFFIN. *De la spécification à l'exploitation des systèmes d'information pour le génie automatique*. Research Report, Automated Production DEA, Ecole Normale Supérieure de Cachan. September 1993.

[COU 96] : F. COUFFIN, J.-J. DUMERY, J.-M. FAURE, J.-P. FRACHET. *La méta-modélisation pour l'intégration des outils-méthodes : application au GEMMA et au Grafcet*. Proceedings of the AFCET'96 Congress : Reactive systems modelling. Brest, France. p. 381-388. March 28-29, 1996.

[DEN 89] : C. DENIS. *Le méta-modèle de représentation des méthodologies de conception des systèmes d'information du projet GEMME*. Proceedings of the INFORSID'89 Congress. p. 95-118. Nancy, France. May 23-26, 1989.

[DEN 92] : B. DENIS, J.-J. LESAGE, G. TIMON. *Toward a theory of integrated modelling*. Proceedings of the IFAC Workshop on CIM in process and manufacturing industries. Espoo, Finland. November 1992.

[DID 95] : M. M. DIDIC, F. COUFFIN, E. HOLLER, S. LAMPERIERE, F. NEUSCHELER, J. ROGIER, M. DE VRIES. *Open Engineering and Operational Environment for CIMOSA*. Computers in Industry, vol. 27 n° 2. p. 167-178. October 1995.

[FAU 92] : J.-M. FAURE. *Quelques difficultés à transposer des spécifications GRAFCET dans un langage d'automate "inspiré du GRAFCET"*. Proceedings of the GRAFCET'92 Congress : Theory and applications - Grafcet. p. 197-205. Paris, France. March 1992.

[FAU 96] : J.-M. FAURE, A. BASSAND, F. COUFFIN, J.-P. FRACHET. *Intégration des formalismes CIMOSA et GRAFCET pour la modélisation des aspects fonctionnel et dynamique des systèmes de production*. Proceedings of the 5th International Congress of Industrial Engineering. p. 19-25, tome III. Grenoble, France. April 2-4, 1996.

[FRA 87] : J.-P. FRACHET. *Une introduction au Génie Automatique : faisabilité d'une chaîne intégrée d'outils CAO pour la conception et l'exploitation des machines automatiques industrielles*. Ph. D., Nancy I University. July 1987.

[FRA 93] : J.-P. FRACHET, J.-M. FAURE, F. COUFFIN. *L'intégration par les données en productique. Application au génie automatique*. Proceedings of the AFCET'93 Congress. p. 99-110, tome 12. Versailles, France. June 8-10, 1993.

[FRO 93] : B. FROMENT. *Structural design with GRAFCET : towards a global method*. Automatic Control Production Systems Review, AFCET/HERMES, vol. 27 n° 1. p. 11-23. 1993.

[HAB 88] : H. HABRIAS. *Proposals for some extensions of the NIAM model*. Proceedings of the Information & Knowledge Engineering Congress. Paris, France. April 1988.

[IEC 91] : *Preparation of function charts for control systems*. International Standard, IEC 848. 1991.

[KLE 71] : S. KLEENE. *Logique Mathématique*. Armand Colin Publishers. 1971.

[LAM 94] : S. LAMPERIERE. *Formalisation d'un outil de modélisation entité relation à l'aide d'un métamodèle*. Research Report, Automated Production DEA, Ecole Normale Supérieure de Cachan. July 1994.

[LAM 95] : S. LAMPERIERE, F. COUFFIN, J.-P. FRACHET. *Méta-modélisation pour l'intégration par les données des activités d'ingénierie des systèmes industriels. Application à la réalisation d'un atelier logiciel d'aide à la conception des systèmes d'information*. Proceedings of the International industrial engineering Conference : Productivity in a world without borders. Montreal (Quebec), Canada. p. 221-230. October 1995.

[LES 94] : J.-J. LESAGE. *Contribution à la formalisation des modèles et méthodes des systèmes de production*. Habilitation à diriger des recherches, Nancy I University. February 1994.

[LHO 93] : P. LHOSTE, H. PANETTO, M. ROESCH. *GRAFCET : from syntax to semantics*. Automatic Control Production Systems Review, AFCET/HERMES, vol. 27 n° 1. p. 127-141. 1993.

[LHO 94] : P. LHOSTE. *Contribution au Génie Automatique : concepts, modèles, méthodes et outils*. Habilitation à diriger des recherches, Nancy I University. February 1994.

[MOR 91] : J. MOREJON, R. OUDRHIRI, M. de GAUDEMONT, P. NEGROS. *GraphOR : A Meta Design Tool*. Proceedings of the Entity-Relationship Approach Conference : The Core of Conceptual Modelling. p.43-56. H. Kangassalo Editor, Elsevier Science Publishers B.V. (North-Holland). 1991.

[MOR 94] : J. MOREJON. *MERISE Vers une modélisation orientée objet*. Les Editions d'Organisation, collection Ingénierie des systèmes d'information. 1994.

[PER 89] : H. PERRIEN. *Modèle de structure de données dite application pour un poste de travail de conception, de réalisation et d'exploitation d'équipements de production automatisée*. Ph. D., Université des Flandres Artois. July 1989.

[PIE 96] : L. PIETRAC, B. DENIS, J.-J. LESAGE. *Formalization of the design of control systems.* Proceedings of the WAC'96 Congress. Montpellier, France. May 1996.

[REV 95] : N. REVAULT, H. A. SAHRAOUI, G. BLAIN, J.-F. PERROT. *A MetaModeling technique : the METAGEN system.* Proceedings of the TOOLS EUROPE'95 Conference, TOOLS 16, Prentice Hall. p. 127-139. Versailles, France. March 1995.

[ROC 88] : A. ROCHFELD, J. MOREJON. *Merise : une méthode en mouvement.* Modèles et Bases de Données Review, AFCET, n°8. p. 37-56. February 1988.

[STE 93] : P. M. STEELE, A. B. ZASLAVSKY. *The Role of Meta Models in Federating System Modelling Techniques.* Proceedings of the 12th international conference Entity-Relationship Approach, Springer-Verlag. p. 315-326. Arlington, USA. December 1993.

[UTE 93] : UTE, Union Technique de l'Electricité. *Etablissement des diagrammes fonctionnels pour systèmes de commande. Diagramme fonctionnel «GRAFCET». Extension des concepts de base.* Publication UTE C 03-191. June 1993.