Customers – Suppliers relationship self organized control modelling using DEVS formalism

Lynda MEKAOUCHE, Fouzia OUNNAR, Patrick PUJO, Norbert GIAMBIASI
Laboratoire des Sciences de l’Information et des Systèmes (LSIS)
Université Paul Cézanne, Avenue Escadrille Normandie Niemen
13397 Marseille Cedex 20 – France
{lynda.mekaouche, fouzia.ounnar, patrick.pujo, norbert.giambiasi}@lsis.org

Abstract - The control of the relationship between partner companies concerns all the actions they develop together to achieve their common objectives and to react at the right time to any failure of one of the partners. A negotiation between the partners is thus required, and this approach involves the management and organization of each partner’s production. The control system of each actor partner must thus be adaptive enough to satisfy the requirements of production. To improve the customer - supplier relationship we propose a self organized control model in which the decision system manages the operation of a group of actors who are in a partnership. The proposal is based on the concept of Autonomous Control Entity (ACE). An ACE is the decision-making centre associated with the production system of the actor; it allows a supplier to become an intelligent production unit able to operate in self organization with other companies. We will present the global operation of the proposed control model then we will focus on the modelling of the ACE's modules using the DEVS formalism.

Keywords: Control, self organisation, customers-suppliers relationship, DEVS formalism.

I. INTRODUCTION

Today, the manufacturing companies evolve under the pressure of globalization, competition, reduced cycle times and increasing complexity. Consequently, much of them are obliged to improve their performance, to reduce deadlines and costs and to increase the diversity of the products as well as their qualities [GUI, 97]. Nevertheless, this solution remains insufficient because of the appearance, on the market, of more complex products. Having been aware of this situation, the companies decide to focus only on their basic trades, they thus apply the outsourcing to a part of their activities, which involves a new form of organization [OUN, 01]. Through, the outsourcing phenomenon, companies tend to gather for the realization of a joint project. Indeed, the company fits in a customers/suppliers network, forming thus a supply chain network, in order to optimize it by satisfying the customer. In order to make these companies cooperate along the supply chain, and seek together its optimization, we see throughout the last years the integration of a new approach in the industry world, which is the company partnership [CHA, 01]. Indeed, the latter pushes the companies to react and develop, together, actions to face a dysfunction or a disturbance of one of the partner. On this basis, we propose a new approach of customers/suppliers relationship control by considering that the whole of the entities (customers/suppliers) partners, communicating on the same medium of communication, negotiate to answer with the better way to the customers needs. In other words, to respond to the Calls For Proposal (CFPs) launched on the network by the customers, and to exploit the capacities of the suppliers with the better way. We have thus provided each supplier with a decision-making centre: Autonomous Control Entity (ACE), which allows him to self evaluate his performance in order to be able to take part to the negotiation. In the future, our objective is to validate the suggested approach by a distributed simulation which will be applied on a set of distinct entities. For that, a simulation is considered on each entity. This latter, will enable us to have an idea on the balance charge/capacity of the CFPs treated by the network partners, and on the number of CFPs launched, treated, refused, negotiated, etc... We will thus be able to install performance indicators of the suggested approach. In this paper, we present the modelling of the autonomous control entity by using the DEVS formalism [ZEI, 84].

Firstly, we describe the general operation of this decision-making centre (the ACE), then we describe the various modules of an ACE. Secondly, we present the DEVS formalism by introducing the bases concepts. The last part will be devoted to the description and the presentation of the DEVS model of each module composing the ACE. These models will be used as formal specifications in the system realization.

II. DESCRIPTION OF THE AUTONOMOUS CONTROL ENTITY GENERAL OPERATION

The suggested approach allows the increase of the autonomy of the network entities. For that, the entities must have the capacity to negotiate and communicate among them in order to achieve their common goal which is to ensure collectively the distribution of the orders coming from the various customers with respecting the interests of each one. When a customer launches a CFP, the suppliers enter into a negotiation step in order to seek the best answer. We thus come to the proposal of a self organized control model. The concept of self organization is subordinate, to use a
decentralized decision structure\(^1\), and to take into account the one behaviour of each component. With this approach there is no estimated organization. The self organization would be a decision-making in real time mode. Then, a common goal is necessary in order to have organization. This can be translated into coordination, co-operation and negotiation terms. Finally, the adopted solution to make the whole of these components operating will be obtained by emergence. For that, we have associated to each partner a decision-making centre named Autonomous Control Entity (ACE) which allows him to self evaluate his performance in order to be able to take part to the negotiation. Four modules composed this entity: Communication Module, Interaction Module, Optimization Module and Planning Module.

A. Communication Module

It is used to communicate with other partners via an extranet. The latter allows the reception and the diffusion of the CFPs as well as the responses to CFPs. The communication is structured around the communication protocol for sending messages, known as "network of contract" (contract-net) [SMI, 80], [SMI, 83]. The principle consists in diffusing a CFP message about an order, then in comparing the received offers (responds) in order to attribute a contract to the entity which proposes the best offer.

B. Interaction Module

It ensures the comparison between the answers suggested by the network partners for a given call for proposal with respect to its entity response. The main functionalities of this module are summarized in two points:

1) The transmission of various information concerning the CFPs or the response to the CFPs received through the communication module towards the optimization module or vice versa.

2) The positioning of each entity with regard to the received offers (updated of the CFP): for each received new offer, the corresponding CFP is updated if the received offer is the first one or if it is better than the best offer already received.

C. Optimization Module

It allows the ACE to self evaluate its performance with respect to the received CFP, in order to estimate its own capacity to respond to this one [OUN, 04]. The performance evaluation is based on a multicriteria method. The selected multicriteria method corresponds to the Analytic Hierarchy Process (AHP) method [SAA, 80] [OUN, 99]. Pair-wise comparison are the central ‘ingredient’ of the AHP. In the case of a qualitative comparison, it is necessary to choose a scale of values to specify the degree of importance. The present study adopted the scale of 1 to 9 used in the AHP [SAA 80]. On the basis of qualitative or quantitative criteria, AHP method ensures to classify CFPs, according to the capacity of the entity to treat them. Among the quantitative criteria, appears operating time of the CFP. This data is obtained by the planning module. For that, the knowledge of the states of CFPs is necessary. The different CFPs received by the entity are placed in the planning module. A CFP can be in one of the following states:

1) Negotiated: characterizing the fact that we have no information about its assignment.
2) Engageable: characterizing the fact that an entity is the most successful on an order (the offer is better than the best of the received offers).
3) Pre-engaged: a call for proposal is pre-engaged if it is “engangeable” and selected as being one of the most priority of the CFP list. The entity appropriates temporarily a CFP. This CFP will be the following one that will take the state “Engaged” if there is no overbid.
4) Engaged: the entity appropriates definitively the “pre-engaged” CFP, on its schedule at its engaged date.
5) Refused: specify the fact that no proposition was made for this CFP.

D. Planning Module

It allows the calculation of the operating time for CFP using an analytic method applied to the entity production system. This method is based on the various states of the planning. A partner can be customer, supplier or both. One of the customers launches a CFP on the network. This latter will be provided with a number of information such as: name of the entity (customer), quantity wished by the customer, lead time of end of negotiation, delivery lead time, etc... All the ACEs which are connected to the network will receive this CFP. Once the CFP will be received by a given ACE via its communication module, this latter will transmit the received information to the interaction module. The interaction module will check the feasibility of the CFP in technical term and then transmit the characteristics’ CFP to the optimization module. The optimization module starts the application of the multicriteria method (AHP) in order to obtain a classification of all the CFPs received, according to the entity capacity to treat them. The application of this method requires a set of qualitative or quantitative criteria [GLA, 02]. Among the quantitative criteria, appears operating time for the CFP at the production system level. This data depends on the state of the planning and of the availability of the equipment. We propose to obtain this

---

\(^1\) The decision system managing the operation of a set of components is named decentralized when these components organize their own operation without the direction of any higher hierarchical level decision-making centre.
data by the execution of an analytic method at the level of the planning module. This latter calculates the operating time for the CFP by studying the various possible states of insertion of the CFP in the entity planning. This result will thus be transmitted to the optimization module in order to finish the application of the multicriteria method. Once the classification will be carried out, the optimization module calculates the local performance of the entity with respect to the CFP classified first by the multicriteria method. The interaction module compares then this performance with regard to the best actual performance and then send it on the network if it is the best one (see Fig.1).

The various messages circulating on the network can be summarised as follow: CFP, RCFP, LCFP (Local Call for Proposal which is a proposed response proposed by a given partner), ERCFP (Entity Response to a Call For Proposal which is a proposed response to a CFP launched by a given partner).

III. MODELING OF AUTONOMOUS CONTROL ENTITY USING DEVS FORMALISM

In this paper, we present the modelling of the assignment protocol of the orders described above in an ACE. For that, we describe the various models of the component modules of this ACE by using the DEVS formalism, developed for modelling and simulation of discrete events dynamic systems [ZEI, 84]. Introduced by Zeigler, DEVS (Discrete Event System Specification) allows the specification of the discrete event models. DEVS enables the development of robust model representation, based on atomic models concepts and on higher-level models representation closed under coupling. This enables a hierarchical modelling, where atomic models are considered as black boxes. DEVS atomic model specification is structured as follow:

$$\text{DEVS} = (X_M, Y_M, \delta_M, \delta_{s0}, \delta_{o0}, \delta_{in}, \lambda, \tau_a)$$

Where:

- $X_M$: The set of input ports through which external events are received.
- $Y_M$: The set of output ports through which external events are sent.
- $S$: The set of the sequential states.
- $\delta_{s0}$: the external transition function which specifies how the system changes state when an input $(x)$ is received during the time “$e$” where $e \leq \tau_a$, then, the system passes from the current “$s$” state to a new one “$s’$” by applying the function $\delta_{s0}(s, e, x)$.
- $\delta_{o0}$: The internal transition function which specifies to which next state the system will transit after the time given by the time advance function has elapsed. The latter can generate an external output just before it takes place.
- $\lambda$: The output function, it generates an output event.
- $\tau_a(s)$: For a given state, $s$, $\tau_a(s)$ represents the time interval during which the model will remain in the state “$s$” if no external event occurs.

Coupled DEVS model describes the system as a network of coupled components, the latter can being of atomic DEVS models or coupled DEVS models. A coupled DEVS model is structured as follow:

$$CM= (X,Y,D,[M_i|d \in D],EIC, EOC, IC)$$

The definition of $X$ and $Y$ is identical to that of $X_M$ and $Y_M$ of a atomic model. The inputs and outputs are made up of ports, each port can take values, and each port has its own field of values.

$$X = \{p \in \text{Input Ports} \mid p \in \text{Input Ports} \}, y \in X \}$$

$$Y = \{p \in \text{Output Ports} \mid p \in \text{Output Ports} \}, y \in Y \}$$

$D$: DEVS component set

$M_d$: The DEVS model for each $d \in D$, the latter can be either atomic DEVS or coupled DEVS.

$EIC$: The external input coupling which connects the input ports of the coupled model to one or more of the input ports of the components.

$EOC$: The external output coupling which connects output ports of components to output ports of the coupled model.

$IC$: The internal coupling which connects output ports of components to input ports of other components.

In a coupled model an output port of a model $M_i$ can be connected to the input of another model $M_d$ but not directly to its input.

A. Formal description of “Communication Module”

-Input ports: $X_c = \{CFP, LCFP2, RCFP, ERCFP1, LRCFP\}$, where:

- $CFP = \{CFP\}$: it indicates the arrival of a given CFP, from the network. The latter is represented by:

$\text{Num}_i \in \mathbb{N}$: number of the CFPs.

$\text{Entity}$: name of the entity defined on the set of string.

$\text{CFP.Type}_i \in \{\text{feasible, unfeasible}\}$: it describes the feasibility of CFP, in technical term.

$\text{CFP.state}_i \in \{\text{negotiated, engageable, pre-engaged, engaged}\}$: it describes the state of the CFP.

$\text{CFP.cond}_i \in \{\text{OK, not OK}\}$: describes the availability of the execution conditions.

$Q_i \in \mathbb{R}$: CFP quantity expected by the customer.

$\text{datecept}_i \in \mathbb{R}$: reception date of CFP,

$\text{OT}_i$: operating time of CFP, defined on the set of reals.

$\text{DE}_i \in \mathbb{R}$: date of beginning of the execution of the CFP.
B. Formal description of “Interaction Module”

This model is a coupled DEVS model composed of three atomic DEVS models (see Fig. 3). CFP/LCFP management submodule which ensures the transmission of the CFP received from the communication module towards the optimization module. It also ensures the transmission of the LCFP coming from the optimization module to the communication module. RCFP/ERCFP management submodule which allows the comparison of the RCFP launched on the network towards its response concerning the same CFP (ERCFP) or vice versa. RLCFP management submodule allows the reception of the different responses proposed by the partners’ network concerning a given LCFP launched by the entity. Then it transmits the latest response received concerning this LCFP after the achievement of its time of end of negotiation. The coupled model of the interaction module is composed of six input ports: ERCFP, RCFP1, CFP1, LCFP1, RLCFP1, reinit and five output ports: ERCFP1, OC, OS, CFP2, LCFP2.

1) CFP/LCFP management submodule:
- Input ports: \( X_{CFP,LCFP} = \{CFP1, LCFP1, reinit\} \), \( CFP1 = \{CFP_i\} \): it indicates the arrival of a given CFP, from the communication module.
- \( LCFP1 = \{LCFP_i\} \): It indicates the arrival of a given LCFP, from the optimization module.
- \( Reinit = \{on\} \): allows the reinitialization of the system.
- - State variables: \( S = \{phase, T, L, L1\} \), where:
  - \( Phase = \{\text{wait, CFP\_recept, LCFP\_recept, } RCFP\_recept, ERCFP\_recept, RLCFP\_recept\} \), \( \sigma \in R^+ \): is the life time of the current state.
  - \( T \): defined the instance of class object CFP frame.
  - \( TR \): defined the instance of class object RCFP frame.
- -Output ports: \( Y = \{CFP1, LCFP3, RCFP1, ERCFP2, RLCFP1\} \) where:
  - \( CFP = \{CFP_i\} \), \( RCFP = \{RCFP_i\} \), \( LCFP = \{LCFP_i\} \), \( ERCFP = \{ERCFP_i\} \), \( RLCFP = \{RLCFP_i\} \).

The corresponding DEVS model is defined in Fig. 2

![Fig. 2 DEVS model of communication module](image)

Fig. 2 DEVS model of communication module

transmission of the CFP received from the communication module towards the optimization module. It also ensures the transmission of the LCFP coming from the optimization module to the communication module. RCFP/ERCFP management submodule which allows the comparison of the RCFP launched on the network towards its response concerning the same CFP (ERCFP) or vice versa. RLCFP management submodule allows the reception of the different responses proposed by the partners’ network concerning a given LCFP launched by the entity. Then it transmits the latest response received concerning this LCFP after the achievement of its time of end of negotiation. The coupled model of the interaction module is composed of six input ports: ERCFP, RCFP1, CFP1, LCFP1, RLCFP1, reinit and five output ports: ERCFP1, OC, OS, CFP2, LCFP2.

![Fig. 3 DEVS model of interaction module](image)

Fig. 3 DEVS model of interaction module
2) RCFP/ERCFP management submodule:
-Input ports: $X_{RCFP/ERCFP} = \{RCFP1, ERCFP, reinit\}$, where:
  $RCFP1 = \{RCFP\}$,
  $ERCFP = \{ERCFP\}$.
  reinit: [on].
- State variables: $S = \{phase, T, L2\}$, where:
  $Phase = \{wait, seek_RCFP, seek_ERCFP, insertion_end1, insertion_end2, compare, crush\}$,
  $\sigma \in R^+$: is the life time of the current state.
  TR: defined the instance of class object RCFP frame.
  L2: list where are stocked RCFPs.
  For these variables are applied following functions:
  “seek(L,TR.num)” : seek the element TR in the list L.
  “Ins(L,last,TR)” : inserts the element TR at the last of the list L.
  “Remp(L,pos,TR)” : replaces the element being in the position “pos” in the list L by element TR.
- Output ports: $Y_{RCFP/ERCFP} = \{ECFP1, OC, OS\}$, where:
  $ECFP = \{ECFP\}$,
  $OC = \{CFP\}$: It represents an order of change of a given CFP state.
  $OS = \{CFP\}$: it indicates an order of suppression of a given CFP.
  The corresponding DEVS model is defined in Fig. 5.

3) RLCFP management submodule:
-Input ports: $X_{RLCFP} = \{RLCFP1, reinit\}$, where:
  $RLCFP1 = \{RLCFP\}$: it indicates the arrival of a given response to a LCFP, coming from the communication module which is proposed by a given network partner.
  reinit: [on]: allows the reinitialization of the lists.
- State variables: $S = \{phase, TR, L3\}$, where:
  $Phase = \{wait, seek_RLCFP, suppression, pause, crush\}$,
  $\sigma \in R^+$: is the life time of the current state.
  TR: defined the instance of class object RCFP frame.
  L3: list where are stocked RLCFPs.
  For these variables are applied following functions:
  “seek(L,TR.num)”,”ins(L,last,TR)”,”remp (L,pos,TR)”.
  “Sup(L, TR)” : remove the element TR from the list L.
  “Getmin(L,TEN)” : function which returns the smallest date of end of negotiation.
  The corresponding DEVS model is defined in Fig. 6.

![Fig. 5 DEVS model of RCFP/ERCFP management submodule](image1)

![Fig. 6 DEVS model of RLCFP management submodule](image2)

C. Formal description of “Optimization Module”

This model is a coupled DEVS model composed of four atomic DEVS models (see Fig 7).
CFP/OS management submodule which allows the transmission of the CFP received from the communication module to the application_AHP submodule and OS (Order of Suppression) to the planning module. LCFP/OC management submodule which ensures the management of LCFPs coming from the planning module and their transmission to the interaction module. This sub module also ensures the management of Order of Change (OC) for a given CFP state. Indeed, the reception of an OC induced its transmission to the planning module. Application_AHP submodule allows the application of the AHP method on receiving CFPs list coming from the CFP/OS management submodule, with an aim of obtaining a classification of these CFPs. Calcul_performance submodule allows the calculation of the entity performance with respect to the CFP, classified first by AHP method. In the calculation of this performance it calls upon a mathematical formula based on a set of indicators applied by calcul_perf function.

The coupled model of the optimization module is composed of seven input ports: CFPvalued, CFPdate, OS, CFP2, LCFP, OC, reinit and five output ports: LCFP1, CFP3, OS1, OC1, ERCFP.

![Fig 7 DEVS model of optimization module](image3)
1) LCFP/OC management submodule:
- Input ports: \(X_{\text{LCFP/OC}} = \{LCFP, OC\}\), where:
  \(LCFP = \{\text{LCFP}\}\): it indicates the arrival of a Local CFP coming from the planning module.
  \(OC = \{\text{CFP}\}\): It represents an order of change of a given CFP, state. This order is launched by the interaction module.
- State variables: \(S = \{\text{phase}, \sigma, T\}\) where:
  \(\text{Phase} = \{\text{wait}, \text{CFP\_recept}, \text{OC\_recept}\}\), \(\sigma \in \mathbb{R}^+\): it is the life time of the current state.
  \(T\): defined the instance of class object CFP frame.
- Output ports: \(Y_{\text{LCFP/OC}} = \{\text{LCFP1}, OC1\}\), where:
  \(\text{LCFP1} = \{\text{LCFP}\}\): send the LCFP to the interaction module.
  \(OC1 = \{\text{CFP}\}\): send the OC on the state of a given CFP, to the planning module.

The corresponding DEVS model is defined in Fig. 8.

![Fig. 8 DEVS model of LCFP/OC management submodule](image)

2) CFP/OS management submodule:
- Input ports: \(X_{\text{CFP/OS}} = \{\text{CFP2, CFPdate, OS, remain, reinit}\}\), where:
  \(\text{CFP2} = \{\text{CFP}\}\): indicates the arrival of a CFP, from the interaction module.
  \(\text{CFPdate} = \{\text{CFP}\}\): receives a CFP, coming from the planning module on which a calculation of operating time was carried out.
  \(\text{OS} = \{\text{CFP}\}\): it receives a suppression order coming from the interaction module concerning a given CFP.
  \(\text{remain} = \{\text{listeCFPs}\}\): allows the reception of the result of AHP method applied at the Application_AHP submodule. This result is represented by a list comprising all the CFPs received classified according to the capacity of the entity to treat them except the CFP classified first.
  \(\text{reinit} = \{\text{on}\}\): allows the reinitialisation of the system.
- State variables: \(S = \{\text{phase}, \sigma, T, L8\}\) where:
  \(\text{Phase} = \{\text{wait}, \text{App\_AHP}, \text{supp\_recept}\}\), \(\sigma \in \mathbb{R}^+\): defined the life time of the current state.
  \(T\): defined the instance of class object CFP frame.
  \(L8\): list where the CFPs which are classified by AHP method are stocked.
- Output ports: \(Y_{\text{CFP/OS}} = \{\text{bestCFP, remain}\}\), where:
  \(\text{bestCFP} = \{\text{CFP}\}\): send the CFP classified first to the planning module.
  \(\text{remain} = \{\text{listeCFPs}\}\): sends the list of the CFPs classified except the CFP classified first to the CFP/OS management submodule.

Fig. 9 presents the corresponding DEVS model.

![Fig. 9 DEVS model of CFP/OS management submodule](image)

3) Application_AHP submodule:
- Input ports: \(X_{\text{Ap\_AHP}} = \{\text{CFPs}\}\), where:
  \(\text{CFPs} = \{\text{listeCFPs}\}\): allows the reception of a CFPs list coming from the management CFP/OS submodule, intended to be classified.
- State variables: \(S = \{\text{phase}, \sigma, T, L5\}\) where:
  \(\text{Phase} = \{\text{wait}, \text{App\_AHP}, \text{supp\_recept}\}\), \(\sigma \in \mathbb{R}^+\): defined the life time of the current state.
  \(T\): defined the instance of class object CFP frame.
  \(L5\): list where are stocked CFPs launched by the network partners intended to be classified by AHP method.
- Output ports: \(Y_{\text{Ap\_AHP}} = \{\text{bestCFP, remain}\}\), where:
  \(\text{bestCFP} = \{\text{CFP}\}\): send the CFP, to the planning module with an aim of calculating its operating time.
  \(\text{remain} = \{\text{listeCFPs}\}\): sends the list of the CFPs to the Application_AHP submodule, in order to classify them.

OS1 = \{\text{CFP}\}\): send the suppression order about a given CFP to the planning module.

Fig. 10 presents the corresponding DEVS model.

![Fig. 10 DEVS model of Application_AHP submodule](image)
4) Calcul_performance submodule:
- Input ports: \( X_{CFP} = \{ \text{CFPrevalued}, \text{bestCFP}, \text{reinit} \} \), where:
  \( \text{CFPrevalued} = \{ \text{CFP} \} \): it indicates the arrival of revalued CFP, coming from the planning module.
  \( \text{bestCFP} = \{ \text{CFP} \} \): allows the reception of the CFP, on which the entity is better. This CFP is sent by the Application_AHP submodule.
  \( \text{reinit} \{ \text{on} \} \): allows the reinitialization of the lists.
- State variables: \( S = \{ \text{phase}, \sigma, T, TR, L6, L7, result \} \), where:
  \( \text{Phase} = \{ \text{wait, seek_CFP, seek_RCFP, Insertion_end, Insertion_end1, Insertion_end2, cruch} \} \).
  \( \sigma \in \mathbb{R}^+ \): defined the life time of the current state.
  \( T \): defined the instance of class object CFP frame.
  \( TR \): defined the instance of class object RCFP frame.
- L6: list where the responses for a different CFPs proposed by the entity (ERCFP) are stocked.
- L7: list where the revaluate CFPs are stocked.
- For these variables are applied following functions:
  - \( \text{seek}(L7,T,\text{num}) \).
  - \( \text{ins}(L\text{last},T) \).
  - \( \text{remp}(L\text{pos},TR) \).
  - \( \text{calcul_perf}(T) \): function allowing the calculation of the performance of a given CFP stocked in \( T \) by the application of a formula based on a set of indicators.
  These functions return a vector made up of the following fields: \( \{ \text{num, entity, pi, decrepti, TENi} \} \).
- Output ports: \( Y_{CFP} = \{ \text{ERCFP} \} \), where:
  \( \text{ERCFP} = \{ \text{ERCFP} \} \).
- Fig. 11 presents the corresponding DEVS model.

D. Formal description of “Planning Module”

The model is a coupled DEVS model composed of tow atomic DEVS models (see Fig 12).
- Operating time management submodule which allows the calculation of the operating time of a given CFP. In the calculation of this time it calls upon the function “calcul_date” based on the study of the possibility of insertion of this CFP at the planning level, and the reception of a removing order (OS) on a given CFP. Thus, it allows the revaluation of CFPs located downstream from the removed CFP. In the revaluation, it uses the function “revaluate” which returns the list of CFPs whose date have been revalued. LCFP management submodule which launches a LCFPs.
- Application_AHP submodule which launches a LCFPs.
- Fig. 12 DEVS model of Planning module

1) Operating time management submodule:
- Input ports: \( X_{OT} = \{ \text{CFP3, OC1, OS1, Init, reinit} \} \), where:
  \( \text{CFP3} = \{ \text{CFP} \} \): allows the reception of a CFP coming from the optimization module intended to be inserted in planning.
  \( \text{OC1} = \{ \text{CFP} \} \): indicates the arrival of an order of change on a state of a given CFP.
  \( \text{OS1} = \{ \text{CFP} \} \): Indicates the arrival of a removing order of a given CFP from the planning.
  \( \text{Init} = \{ \text{ListeCFPs} \} \): allows the initialization of CFPs list of CFPs being in planning.
  \( \text{reinit} \{ \text{on} \} \): allows the reinitialization of the lists.
- State variables: \( S = \{ \text{phase}, \sigma, T, L9, date, listerep, i} \) where:
  \( \text{Phase} = \{ \text{wait_init, wait, calcul_date, change, suppression, diffusion} \} \).
  \( \sigma \in \mathbb{R}^+ \): it is the life time of the current state.
  \( T \): defined the instance of class object CFP frame.
  \( L9 \): basic list where are stocked all the CFPs of the planning.
  \( \text{Date} \): defined the instance of the class object CFP frame where is stocked the CFP, after the calculation of its operating date.
  \( \text{Listerep} \): list where are stocked CFPs which underwent a revaluation after a removing order of a given CFP.
- To these variables are associated the following operations:
  - \( \text{calcul_date}(T,L9) \): function which allows the insertion of a given CFP stocked in \( T \) in planning and returns as a result the same CFP with a modification at the level of its operating date.
  - \( \text{Revaluate}(L10,T) \): function which allows the revaluation of CFPs located downstream from the removed CFP. This function returns as a result a list of the revalued CFPs.
- Output ports: \( Y_{OT} = \{ \text{CFPdate, CFPrevalued} \} \), where:
  \( \text{CFPdate} = \{ \text{CFP} \} \): sends the CFP, to the optimization module, after calculating its operating date.
  \( \text{CFPrevalued} = \{ \text{CFP} \} \): send the CFP, on which the operating date was revalued to the optimization module.
- Fig. 13, presents the corresponding DEVS model.
The goal of our study is to improve control of customers - suppliers relationship. Firstly, we have treated the evolution of the customers - suppliers relationship which led the appearance of externalisation phenomenon which thus gave rise to what is called supply chain network. Thereafter the problem concerning the complexity of obtaining the best response to a call for proposal launched on this type of network was raised. This led us to present the proposed approach of self organized control of customers - suppliers relationship. The approach is based on the association, in network, of potential suppliers with an aim of ensuring collectively the assignment of orders coming from the different customers with respecting the interests of each partner. To do that, a decision-making centre is associated to each partner. This latter allows the entity (supplier) to self evaluate its performance with regards to receive calls for proposals. The decision-making centre was modelled using the DEVS formalism. Further research will focus on a distributed simulation in order to validate the whole of the approach.

REFERENCES


