A MODEL OF OPTIMAL AGGREGATION OF ACTIONS IN MANUFACTURING PROCESSES

Vasile CRĂCIUNEAN[1]

ABSTRACT: The key to mastering the complexity of large complex production processes is hierarchical modularization, i.e. group actions in cohesive blocks with minimal interaction. Everyone agrees that a production process must be decomposed into hierarchical blocks to be driven and optimized. A very important issue in the organization of production processes is how to aggregate the actions into blocks, which are the blocks that must result and what relationship should exist between these blocks. This is actually the aggregation and the disaggregation problem in complex production processes. This paper proposes a model for determining the optimal architecture of action blocks in designing complex production processes. The model is built by incrementing step by step the actions in a hierarchy of blocks beginning with elementary actions and reaching the complex action which is the production process.

KEY WORDS: action of aggregation, aggregation system of actions, aggregation network of actions, aggregation block of actions.

1 INTRODUCTION

In this paper we consider that the atomic unit of a production process is the action.

Actions and the resources on which the actions are produced are distributed for concurrent execution in a virtual network of companies working together. Actions are very inhomogeneous beginning with actions of design, procurement, logistics, manufacturing, quality control etc. There will be software actions that will store and dynamically process all data on the evolution of the production process.

Each component embeds in it a certain degree of intelligence. It will know how to check if it has provided the necessary context for starting, for example it will know to check if it has the necessary resources for the smooth conduct. If this conditions are not satisfied it will wait until these conditions are met. The wait will last until it gets a message about updating the input resources and then it will restart the process with context verification. The action will know how to decrement stock, for example, with consumed resources and increment the stock with the resource created by it and also sending the corresponding event. An action of aggregation also does a very important thing that is real-time update of developments related to various data such as start time, the actual length, waiting time, costs, etc.

[1] „Lucian Blaga" University of Sibiu, Victoriei Blvd, no. 10, Sibiu, 550024, Romania

E-mail: craciunean@sln.ro;

The fundamental principle with which we start our aggregate the actions into blocks model is maximum cohesion within each block and minimum coupling between the blocks. For maximum cohesion we will group several actions in a block if this block produces a single resource that is used outside of the block and all actions from this block directly or indirectly contribute to the achievement of this resource.

Starting from a hierarchy of elementary actions corresponding to the production process that we want to build, after a first assembly of these actions we get the initial network of aggregation. The actions of this aggregation network will be the starting blocks.

We consider that only a final product will be produced and therefore there will be an initial node represented by the action that has as output the final product.

2 PRELIMINARY NOTIONS AND NOTATIONS

Let X be a set. The family of subsets of X is denoted by $\mathcal{P}(X)$. The cardinality of X is denoted by |X|. The set of natural numbers, {0, 1, 2, ...} is denoted by N. The empty set is denoted by Ø. An alphabet is a finite nonempty set of abstract symbols. For an alphabet V we denote by V^* the set of all strings of symbols in V. A language over the V alphabet is a subset L of V^*.

We begin by defining the notion of action of aggregation that is underlying our model.

An action of aggregation (Crăciunean V.(2014)) a is a construct of the form:

$$a = (I,O,M) \quad (1)$$

where

I={(r_i,c_i,e_i) | i=1,2,...m ; r_i is a resource, c_i is the quantity of that resource, e_i is an event};

The resource r_i is called an input resource and represents the necessary resource in the c_i quantity for the action a, and e_i is called an input event.

O={(r_o,c_o,e_o) | r_o is a resource or a final product, c_o is the quantity of the resource, e_o is an event};

If r_o is the resource then it's called the output resource and c_o is the quantity of that resource, else r_o is the final output product. The event e_o is the output event.

M is a set of data and associated metadata such as duration of action, the minimum allowed stock, costs, technical data about the action, software components, etc.

We consider the set A of actions of aggregation involved in a production process, then we define the precedence actions by two functions, namely: a bottom-up function of precedence f: A → $\mathcal{P}$(A), defined as follows:

$$f(a) = \{b \in A \mid O_a \in I_b\},\ (\forall)\ a \in A \qquad (2)$$

and a top-down function of precedence g: A → $\mathcal{P}$(A), defined as follows:

$$g(b) = \{a \in A \mid O_a \in I_b\},\ (\forall)\ b \in A. \qquad (3)$$

The symbols from the A set, represent real elementary actions and therefore the semantic load of the elements of the vocabulary is well defined.

An aggregation system of actions(Crăciunean V.(2014)) is a construct of the form

$$S = (A, f, g, \{L_X \mid X \in \mathcal{P}(A)\}) \qquad (4)$$

where

A is a finite set of actions called action of aggregation;

f is a function f: A → $\mathcal{P}$(A) named bottom-up function of precedence;

g is a function g: A → $\mathcal{P}$(A) named top-down function of precedence;

L_X is a language over A, (∀) X∈$\mathcal{P}$(A). It is obvious that L_X can be an empty language for some X∈$\mathcal{P}$(A).

Based on the function f we define the application of aggregation φ as follows:

An application of aggregation (Crăciunean V. (2014)) is an application φ: $\mathcal{P}$(A) → $\mathcal{P}$(A) defined as follows:

$$\varphi(X) = \bigcup_{x \in X} f(x) \quad (\forall) X \in \mathcal{P}(A). \qquad (5)$$

In this conditions $\varphi^n(X)$ can be defined as:

$\varphi^0(X) = X \quad (\forall)\ X \in \mathcal{P}(A)$;

$\varphi^n(X) = \varphi(\varphi^{n-1}(X))$.

Similarly, we define the application of disaggregation ψ:

An application of disaggregation(Crăciunean V.(2014)) is an application ψ: $\mathcal{P}$(A) → $\mathcal{P}$(A) defined as follows:

$$\psi(X) = \bigcup_{x \in X} g(x) \quad (\forall) X \in \mathcal{P}(A). \qquad (6)$$

Similarly, $\psi^n(X)$ can be defined as:

$\psi^0(X) = X \quad (\forall)\ X \in \mathcal{P}(A)$;

$$\psi^n(X) = \psi(\psi^{n-1}(X)). \qquad (7)$$

3 AGGREGATION NETWORK OF ACTIONS

We now consider an aggregation system of actions:

$$S = (A, f, g, \{L_X \mid X \in \mathcal{P}(A)\}) \qquad (8)$$

as defined above, together with this applications:

$$\varphi: \mathcal{P}(A) \to \mathcal{P}(A)$$
$$\psi: \mathcal{P}(A) \to \mathcal{P}(A) \qquad (9)$$

defined as above.

Based on the application of aggregation φ: $\mathcal{P}$(A) → $\mathcal{P}$(A) we will use the following notations:

$\widehat{\varphi}^k(X) = \bigcup_{i=0}^{k} \varphi^i(X)$;

$$\widehat{\varphi}(X) = \bigcup_{i=0}^{\infty} \varphi^i(X). \qquad (10)$$

Also, we will use similar notations for the application of disaggregation: ψ: $\mathcal{P}$(A) → $\mathcal{P}$(A)

$\widehat{\psi}^k(X) = \bigcup_{i=0}^{k} \psi^i(X)$

$$\widehat{\psi}(X) = \bigcup_{i=0}^{\infty} \psi^i(X) \qquad (11)$$

We consider that a unique final product is manufactured and therefore there will be an initial action that has as output the final product. Suppose we want to manufacture a determined product p, of course we are talking about a product that can be made only with actions from A. Then there will be an action a∈A, a=(I,O,M) such that {p}=O. Under these conditions we can define an aggregation network of actions as follows:

Definition 3.1. An aggregation network of actions is a construct R=(X,φ,ψ)

where:

$X = \widehat{\psi}(\{a\})$;

φ and ψ are the restrictions of the applications φ and ψ defined above at the subset X=$\widehat{\psi}$({a}) of A.

The fundamental principle on which we go in partitioning the actions is maximum cohesion within each block and minimum coupling between the blocks.

For maximum cohesion we will group several actions in a block if:

i)The block produces a single resource that is used outside of it,

ii)All actions in the block directly or indirectly contribute to the realization of resource i)

iii)The only resource that uses a resource from the block and at the same time is used by another resource from the block is the resource from i).

Now we can define an aggregation block of actions in the following way:

Definition 3.2. Let R=(X,φ,ψ) be an aggregation network of actions. Then an aggregation block of actions with the head h is an aggregation network of actions Ɓ=(B,φ,ψ) where B⊆X has the following properties:

- i) h∈B;
- ii) If x∈B⇒x∈$\widehat{\psi}$({h});
- iii) If x∈B\{h} and x∈$\widehat{\psi}$({x}) ⇒ h∈$\widehat{\psi}$({x});
- iv) If x∈ B\{h} then φ(x) ⊂B.

φ and ψ are the restrictions of the applications φ and ψ on subset B of X.

Starting from a hierarchy of elementary actions corresponding to the production process that we want to build, after a first assembly of these actions we get the initial network of aggregation. The actions of this aggregation network will be the starting blocks.

Example 3.1. For the aggregation network of actions R=(X,φ,ψ) where:

X={a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8}

the application of disaggregation will be:

ψ({a_1})={a_2}; ψ({a_2})={a_3}; ψ({a_3})={a_4,a_5}; ψ({a_4})={a_6}; ψ({a_5})={a_6}; ψ({a_6})={a_3,a_7}; ψ({a_7})={a_2,a_8};

and the application of aggregation will be:

φ({a_2})={a_1,a_7}; φ({a_3})={a_2,a_6}; φ({a_4})={a_3}; φ({a_5})={a_3}; φ({a_6})={a_4,a_5}; φ({a_7})={a_6 }; φ({a_8})={a_7};

Therefore we have the blocks:

Ɓ(a_1)={a_1}; Ɓ(a_2)={a_2}; Ɓ(a_3)={a_3,a_4,a_5,a_6,a_7,a_8};

The following algorithm determines a family of aggregation blocks of actions of an aggregation network of actions.

Algorithm 3.1.

Input: The aggregation network of actions R=(X,φ,ψ).

Output: The family of blocks $Ɓ_j$, j=1,p that partitions the aggregation network of actions R.

Method: The algorithm is based on the conditions in the definition of aggregation block of actions.

```
P1.  APRB(R)
P2.      i←1; j←1; l←1; Y←∅
P3.      DO UNTIL(ψ (Y)=∅)
P4.      IF(j>1) THEN
P5.          l←min{m|a_m∈ ψ (Y) }
P6.          y_i←x_l
P7.      ELSE
P8.      y_i←a_i
P9.      ENDIF
P10      I_{j_1} ←{y_i}; Y←Y ∪ { y_i }
P11      k←1P12
P12.     DO UNTIL ( I_{j_{k+1}} ≠ I_{j_k} )
P13.     Z={x∈ψ ( I_{j_k} )| φ (x) ⊆ I_{j_k} };
P14.     I_{j_{k+1}} ← I_{j_k} ∪ Z; Y←Y ∪ Z;
P15          k←k+1
P16.     ENDDO
P17.     B_j← I_{j_k}
P18.      j←j+1; i←i+1
P19.     ENDDO
P20.     p←j-1
P21.     END
```

Lemma 3.1. Let R=(X, φ, ψ) be an aggregation network of actions. Then the algorithm 3.1. determines a unique partitioning of the aggregation network of actions in aggregation blocks of actions.

Demonstration. Let's see first that $Ɓ_1$,$Ɓ_2$,...,$Ɓ_p$ produced by the algorithm satisfies the conditions i) - iv). Let Ɓ $_h$ be such an aggregation block of actions.

Because

$I_{h_1} \subseteq I_{h_2} \subseteq \ldots \subseteq I_{h_n} \Rightarrow B_h = \bigcup_{k=1}^{n} I_{h_k}$

the first action included in I_h is either x_1, either an action y_i selected in the steps 4,5 and 6. This will be the head of the aggregation block of actions I_h. From the step P14)$\Rightarrow(\forall x\in I_h \Rightarrow x\in\hat{\psi}(h))$ and therefore condition (2) is true.

Let $x\in B\setminus\{h\}$ be an action and $x\in\hat{\psi}(\{x\})$ then the following string of actions exists:

$C=[x=x_1,x_2,\ldots,x_n x=x_{n+1}]$ so that $x_i\in\varphi(x_{i+1})\ \forall i=\overline{1,n}$ and $h\notin C$. Let x_{i_0} be the first action from C included in step P14) in the I_h block. Because x_{i_0} has been included $\Rightarrow \varphi(x_{i_0})\subseteq I_{j_k}$. But from the condition $x_{i_0}\in C\Rightarrow \exists\ x_{i_j}\in\varphi(x_{i_0})$ and $x_{i_j}\notin I_{h_k}\setminus\{h\}$because x_{i_0} was the first action from the cycle that we have included in step P14 in $I_h \Rightarrow h\in\hat{\psi}(\{x\})$ and therefore condition (3). Also from step P14 an action is included in $I_{h_{k+1}}$ with the condition that $\varphi(x)\subseteq I_{h_k}$. Therefore condition (iv) is true.

Let us note now that there are no two blocks of aggregation which meet i) -iv) having different heads and concurrent actions.

Indeed let h1 and h2 be the two heads and $x\in B_{h_1}\cap B_{h_2}\Rightarrow\exists\ x\in\hat{\psi}(h_1)$ and $x\in\hat{\psi}(h_2)\Rightarrow\exists\ y_1,y_2$ so that $y_2\in B_{h_2}$şi $y_1\notin B_{h_2}$ so that $y_1\in\varphi(y_2)\Rightarrow\varphi(y_2)\not\subset B_{h_2}$ is in contradiction with the condition 4). Let B_h be an aggregation block produced by the above algorithm B_h' an aggregation block with the same initial action h. From the above it follows that $B_h\subseteq B_h'$. Let us show that $B_h'=B_h$ and we demonstrate the equivalence of the algorithm with the definition. Suppose that $I_h\subset I_h' \Rightarrow \exists\ x\in I_h'\setminus I_h \Rightarrow \exists$ an action x so that $\varphi(x)\subseteq I_h'$ because the number actions is finite and $\varphi(x)\subseteq I_h$. Because I_h' is produced by the alorithm $\Rightarrow\exists$ n so that $I_h' = I_{h_n}' = I_{h_{n+1}}'$. But

$I_{j_{k+1}} \leftarrow I_{j_k} \bigcup \{x\in\psi(I_{j_k}) | \varphi(x)\subseteq I_{j_k}\}$;

contradiction $\Rightarrow B_h=B_h'$. The condition of maximality arises from the fact that $B_h=\bigcup_{k=1}^{n} I_{h_k}$ i.e. B_h is the union of all blocks of aggregation meeting the conditions i) -iv).

Let us prove now that $(B_1,\ldots,B_p)$ forms a partition for X.

Suppose there $\exists x\in X$ so that $x\notin\bigcup_{k=1}^{p} B_k\Rightarrow \psi(Y)\neq\emptyset$ in step P3$\Rightarrow$ the algorithm is not finished. Therefore $B_h=\bigcup_{k=1}^{n} I_{h_k}$.

Let us proof that $B_j\bigcap B_k=\emptyset$.

It is enough to show that there are no two distinct aggregation blocks with the same head h.

We assume by contradiction that B_h and B_h', so that $B_h\setminus B_h'\neq\emptyset\Rightarrow I_h'\subset I_h\bigcup I_h'$. Because $B_h\bigcup B_h'$ satisfies the conditions i)-iv) $\Rightarrow B_h$' is not the maximum with the properties i)-iv), contradiction!. So $B_h= B_h$'.

From the proof of the theorem results, in addition, that the partitioning of an aggregation network into blocks of aggregation is unique.

Definition 3.3. Let $R=(X,\varphi,\psi)$ be an aggregation network of actions and the partion of blocks of actions determined by the algorithm 3.1. It is called a derived aggregation network of actions of the aggregation network of actions R, an aggregation network of actions F(R) defined as follows:

$F(R)=(F_1(X),F_2(\varphi),F_3(\psi))$ where

$F_1(X)$ contains one action for each aggregation block of R determined according to algorithm 3.1.

$F_1(X)=\{B(h_i)|\ h_i\ i=1,p$ are all the aggregation block heads determined by the algorithm 3.1.}. Therefore each block $B(h_i)$ is the aggregation block with h_i head.

The action $B(h_i) = (I_i,O_i,M_i)$ where

$I_i = (\bigcup_{a\in B(h_i)} I_a)\bigcup(\bigcup_{a\in B(h_i)\setminus\{h_i\}} O_a)$,

$O_i = O_{h_i}$ and M_i are the data and the metadata of the block of actions obtained through union of data and metadata of the included actions.

$F_2(\varphi)$ is an application of aggregation Φ defined as follows:

$\Phi : \mathcal{P}(F_1(X)) \rightarrow \mathcal{P}(F_1(X))$

$\Phi(\{B_{h_j}\})=\{B_{h_i}|\ h_i\in\varphi(B_{h_j})\}$;

and

$\Phi(\{B_{h_1},B_{h_2},\ldots,B_{h_k}\})= \bigcup_{j=1}^{k}\Phi(\{B_{h_j}\})$

$F_3(\psi)$ is an application of disaggregation Ψ defined as follows:

$\Psi : \mathcal{P}(F_1(X)) \rightarrow \mathcal{P}(F_1(X))$

$\Psi(\{B_{h_j}\})=\{B_{h_i}|\ h_i\in\psi(B_{h_j})\}$;

and

$\Psi(\{B_{h_1},B_{h_2},\ldots,B_{h_k}\}) = \bigcup_{j=1}^{k}\Psi(\{B_{h_j}\})$

The aggregation network of actions F(R) has one action for each aggregation block of actions. The beginning action of F(R) is the action corresponding to the aggregation block of actions that contains the start action of R.

The aggregation block J belongs to $\psi(I)$ if and only if $I\neq J$ and an action a exists from I so that the head of J is in the set $\psi(I)$.

The derived aggregation network of actions of F(R) is $F(F(R))=F^2(R)$. Since the number of actions from R is finite $\Rightarrow \exists$ n so that $F^n(R)=F^{n+1}(R)$.

Definition 3.4. It's called a limit of the aggregation network of actions R, the derived aggregation network $F^n(R)$ with the property that $F^n(R)=F^{n+1}(R)$. We denote this limit by R_n. From the uniqueness of partitioning the aggregation network in aggregation blocks it follows that R exists and is unique.

Definition 3.5. Let $R=(X,\varphi,\psi)$ be an aggregation network of actions, then if R_n has a single action we say that the aggregation network of actions R is reducible, and if not it is irreducible.

Example 3.2. For the aggregation network of actions $R=(X,\varphi,\psi)$ from the example 3.1. we have:

Level 1.

$F(R)=(F_1(X),F_2(\varphi),F_3(\psi))$ where :

$X_1= F_1(X)=\{ \mathcal{B}(a_1), \mathcal{B}(a_2), \mathcal{B}(a_3)\}$

$\mathcal{B}(a_1)=\{a_1\};\mathcal{B}(a_2)=\{a_2\};\mathcal{B}(a_3)=\{a_3,a_4,a_5,a_6,a_7,a_8\}$

$\Phi_1= F_2(\varphi)$

$\Phi_1(\{\mathcal{B}(a_1)\})=\{ \mathcal{B}(a_2)\}$

$\Phi_1(\{\mathcal{B}(a_2)\})=\{ \mathcal{B}(a_3)\}$

$\Phi_1(\{\mathcal{B}(a_3)\})=\{ \mathcal{B}(a_2)\}$

$\Psi_1= F_3(\psi)$

$\Psi_1(\{\mathcal{B}(a_2)\})=\{ \mathcal{B}(a_1), \mathcal{B}(a_3)\}$

$\Psi_1(\{\mathcal{B}(a_3)\})=\{ \mathcal{B}(a_2)\}$

Level 2.

$F^2(R)=(F_1^2(X), F_2^2(\varphi), F_3^2(\psi))$ where:

$F_1^2(X)=\{\mathcal{B}^2(a_1), \mathcal{B}^2(a_2)\}$

$\mathcal{B}^2(a_1)=\{ \{\mathcal{B}(a_1)\}\}$

$\mathcal{B}^2(a_2)=\{ \mathcal{B}(a_2), \mathcal{B}(a_3)\}$

$\Phi_2= F_2^2(\varphi)$

$\Phi_2(\{\mathcal{B}^2(a_1)\})= \mathcal{B}^2(a_2)$

$\Psi_2=F_3^2(\psi)$

$\Psi_2(\{\mathcal{B}^2(a_2)\})= \mathcal{B}^2(a_1)$

and finally:

Level 3.

$F^3(R)=(F_1^3(X), F_2^3(\varphi), F_3^3(\psi))$ is reduced to a single action:

$F_1^3(X)=\{ \mathcal{B}^2(a_1)\}$

And therefore the aggregation network of action is reducible.

Example 3.3. We consider the aggregation network of actions $R=(X,\varphi,\psi)$, where we have:

$X=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}\}$ and

$\psi(\{a_1\})=\{ a_2\}$; $\psi(\{a_2\})=\{ a_3,a_4\}$; $\psi(\{a_3\})=\{a_5\}$; $\psi(\{a_4\})=\{ a_5,a_{11}\}$; $\psi(\{a_5\})=\{a_6\}$; $\psi(\{a_6\})=\{a_7,a_8,a_9\}$; $\psi(\{a_7\})=\{a_{10}\}$; $\psi(\{a_8\})=\{a_{10}\}$; $\psi(\{a_9\})=\{a_{10}\}$; $\psi(\{a_{10}\})=\{a_{11}\}$; $\psi(\{a_{11}\})=\{a_{12}\}$; $\psi(\{a_{12}\})=\{ a_{11},a_{13}\}$; $\psi(\{a_{13}\})=\{a_6\}$;

and

$\varphi(\{a_2\})=\{a_1\}$; $\varphi(\{a_3\})=\{a_2\}$; $\varphi(\{a_4\})=\{ a_2\}$; $\varphi(\{a_5\})=\{ a_3,a_4\}$; $\varphi(\{a_6\})=\{ a_5,a_{13}\}$; $\varphi(\{a_7\})=\{ a_6\}$; $\varphi(\{a_8\})=\{ a_6\}$; $\varphi(\{a_9\})=\{a_6\}$; $\varphi(\{a_{10}\})=\{ a_7,a_8,a_9\}$; $\varphi(\{a_{11}\})=\{a_4,a_{10},a_{12}\}$; $\varphi(\{a_{12}\})=\{a_{11}\}$; $\varphi(\{a_{13}\})=\{a_{12}\}$;

In this case we have the blocks:

$\mathcal{B}(a_1)=\{a_1,a_2,a_3,a_4,a_5\}$; $\mathcal{B}(a_6)=\{a_6,a_7,a_8,a_9,a_{10}\}$; $\mathcal{B}(a_{11})=\{a_{11},a_{12},a_{13}\}$;

Level 1.

$F(R)=(F_1(X),F_2(\varphi),F_3(\psi))$ where :

$X_1= F_1(X)=\{ \mathcal{B}(a_1), \mathcal{B}(a_6), \mathcal{B}(a_{11}) \}$

$\Phi_1= F_2(\varphi)$

$\Phi_1(\{\mathcal{B}(a_1)\})=\{ \mathcal{B}(a_6), \mathcal{B}(a_{11})\}$

$\Phi_1(\{\mathcal{B}(a_6)\})=\{ \mathcal{B}(a_{11}) \}$

$\Phi_1(\{\mathcal{B}(a_{11})\})=\{ \mathcal{B}(a_6)\}$

$\Psi_1= F_3(\psi)$

$\Psi_1(\{\mathcal{B}(a_6)\})=\{ \mathcal{B}(a_1), \mathcal{B}(a_{11})\}$

$\Psi_1(\{\mathcal{B}(a_{11})\})=\{ \mathcal{B}(a_1), \mathcal{B}(a_6)\}$

We get 3 blocks at level 1 that cannot aggregate as defined in 3.2. So the aggregation network is irreducible. Of course the three blocks can be aggregated into a single block of aggregation of actions taken without the restrictions of definition 3.2.

It therefore starts from the initial aggregation network R which has the original elementary actions of X. We call them blocks of aggregation of level 0. The algorithm 3.1. is applied and we get the aggregation network F(R) with the aggregation blocks of level 1. In general, the aggregation blocks of the aggregation network $F^n(R)$ are called aggregation blocks of level n.

4 CONCLUSION

The model provides real support for the leadership, management and optimization of complex production processes.

Aggregation and distribution of actions in this model enables accurate calculation of costs, time and many other parameters that depend on the development of basic actions.

Another feature of the model is that it reflects a true and fair view of the complexity of the production systems. We introduce measures of production systems as dependent functions proportional to the number of actions, the number of levels, input flows in each aggregation block, etc. We can therefore compare the production processes to each other.

Let's see what happens with the actions dependence. Generally if an action is not dependent on any other action is said to be independent. An action is responsible if other actions depend on it. Any incident in such an action may cause a chain of incidents in actions dependent on it. Generally in a manufacturing process there is no completely independent actions or actions totally free of responsibility.

The dependence of an action a to an aggregation block B is determined from the number of actions from block B on which the action a relies. It seems clear that our division into blocks of aggregation is optimal since any action a from the outside of block B depends on one action from block B and this is the head of the aggregation block h. At first glance the responsibility seems uncontrolled, a closer analysis also shows us that one action of each block is responsible for other actions outside the block and it is also the head of the aggregation block. All actions except the head of a block are only responsible for actions from the block.

An essential gain of partitioning into blocks is the control of critical actions of the process. This is a very important issue especially in Just in Time production where the stock is very important. At the base of the Just in Time method stays the principle of minimizing or eliminating stockpiles of raw materials, components, parts and work in progress and thus reduce the overall costs of these stocks, regardless of production volume. Our model provides necessary information about resources that is required to have stock in order not to create downtime in the production chain. This will be the subject of other future papers.

5 REFERENCES

►Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, (2001), *Compilers: Principles, Techniques, and Tools*, Addison Wesley

►Calude Ch. S., Păun Gh. (2001), *Computing with Cells and Atoms an introduction to quantum, DNA and membrane computing*, Taylor & Francis.

►Crăciunean V.(2014), *A Linguistic Model Of Aggregation In Virtual Manufacturing Processes* - Academic Journal Of Manufacturing Engineering, Vol. 12, Issue 3/2014.

►Crăciunean V. (2014), *Proiectarea Translatoarelor*, Sibiu, Editura Universității Lucian Blaga Sibiu, ISBN 978-606-12-0842-5

►Crăciunean, V. Fabian R. Popa E. M. (2007), *Total Fuzzy Grammar Driven Architecture for Intelligent Software Sytems* – WSEAS Transactions on Computers, Issue 2, Volume 6, pag. 248-253, ISSN 1109-2750.

►Crăciunean, V. Fabian R. Popa E. M. (2007) *Algorithm for Optimal Decomposition of Software Applications*– WSEAS Transactions on Computers, Issue 2, Volume 6, pag. 242-247, ISSN 1109-2750.

►Crăciunean, V. Aron C.E. Fabian R. Hunyadi I.D. (2007), *Multi Level Recursive Specifications for Context Free Grammars* - Proceedings of the 11th WSEAS International Conference on Computers, Agios Nikolaos, Crete Island, Greece, pag. 275-280, ISSN 1790-5117, ISBN 978-960-8457-95-9.

►Crăciunean, V. Fabian R. Hunyadi I.D. Popa E. M. (2007), *Fix point internal hierarchy specification for context free grammars* - Proceedings of the 11th WSEAS International Conference on COMPUTERS, Agios Nikolaos, Crete Island, Greece, pag. 247-251, ISSN 1790-5117, ISBN 978-960-8457-95-9.

►Dassow J. Păun Gh. (1999), *On the power of membrane computing*, J. Universal Computer Sci., 5, 2, 33-49.

►Marvin Scheaffer (1973), *A Mathematical Theory of Global Program Optimization*, Prentice-Hall.

►Matthias Dehmer, Abbe Mowshowitz, and Frank Emmert-Streib, (2013), *Advances in Network Complexity* Wiley-VCH Verlag GmbH & Co. KGaA,Boschstr. 12, 69469 Weinheim, Germany.

►Mowshowitz A. (1994), *Virtual Organization: A vision of Management in the Information Age*, The Information.

►Păun Gh. (1988), *Mecanisme generative ale proceselor economice*, Ed Tehnică, Bucureşti.

►Salomaa, A. (1973), *Formal Languages*, New York, Academic Press.