Real Time Reconfiguration of Manufacturing Systems*

Sébastien Henry
Laboratoire d'Automatique de Grenoble
ENSIEG - Rue de la houille blanche
38402 Saint Martin d'Hères Cedex, France
sebastien.henry@lag.ensieg.inpg.fr

Eric Zamaï and Mireille Jacomino
Laboratoire d'Automatique de Grenoble
ENSIEG - Rue de la houille blanche
38402 Saint Martin d'Hères Cedex, France
eric.zamai@lag.ensieg.inpg.fr
mireille.jacomino@lag.ensieg.inpg.fr

Abstract – *Usually, in an industrial context, the design of discrete control law to drive manufacturing system is assume off line by several experts. This is mainly due to the lack of a generic method to model the controlled system abilities. Consequently, not only the design of all the control laws mobilizes many PLC program developers, but also, in case of unexpected resources failures, the reconfiguration process can be only considered from a manual point of view. So, to bring a solution in this field of research, first a methodological approach to model a controlled system is given. Second, based on such a resulting model, an automated discrete control law synthesis is proposed to provide a law compatible with one of the CEI 61131-3 languages.*

Keywords: Control law, Synthesis, Reconfiguration, Manufacturing systems, Programmable Logic Controllers.

1 Introduction

This paper deals with the dependability of automated systems and more particularly manufacturing systems. These systems are made of a supervision, monitoring and control (SM&C) system and a controlled system (see Figure 1) defined by the set of resources. In this paper, a resource will only be a manufacturing or handling machine. From a hierarchical and modular point of view, a manufacturing system can be a plant, a workshop, a flexible manufacturing cell or a machine. A product flow goes through all these systems. Depending on services offered by the controlled system and the customer's request, the control system applies control laws to act on the product flow.

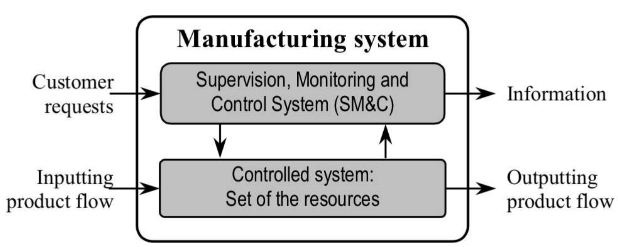

Figure 1. A manufacturing system

The specification of the decisional system in [4] in the context of the supervision, monitoring and control (SM&C) approaches is discussed, but this paper is mainly directed towards the controlled system modeling and the automated control law synthesis. Firstly in the context of manufacturing system design, the objective is to reduce the time required to create a control law in one of the CEI 61131-3 languages and to minimize the debug time by creating correct code initially. Secondly in the context of the dynamic reconfiguration, the reactivity of resource failures and the customer's request variation will be definitely improved.

The functions of a manufacturing system is discussed in section 2. Then section 3 describes the required initial data for the automated control law synthesis. And they are compared with data considered by the classical synthesis approaches. So in section 4, the aim of the paper is presented. In section 5, the controlled system is modeled to synthesize automatically the control laws, in section 6. The paper then concludes with section 7, where we summarize the paper and outline several research directions.

2 Functions of manufacturing system

The basic function (defined by the norm NF X50-100, [7]) of a manufacturing system is to increase the value of the product flow. It achieves its function by having an effect on the product flow, as in Figure 1. This effect is defined for a product by the difference between its input state and its output state which are data for the control law synthesis problem. The secondary function of a manufacturing system is to guarantee the performances. Indeed to assure a profit to the company, the added product flow value must be superior to the manufacturing cost. From a control law point of view, the manufacturing cost is the total of the costs generated by the infringement of the security and environmental constraints and the costs resulting from the values of the productivity and quality criteria. In other words, the control law must respect the security and environmental constraints and optimize the productivity and quality criteria.

* **0-7803-8566-7/04/$20.00 © 2004 IEEE.**

3 Scope

To achieve its functions, the manufacturing system must control the product flow evolutions. These are linked to the resource evolutions controlled by the control system according to interpreted or compiled control laws compatible with the CEI 61131-3 norm. Thus, the control laws of all the levels must respect the controlled system constraints (security and environmental) and the objectives (the product specifications, the values of the productivity and quality criteria) [6]. When an expert synthesizes manually a control law, he has in mind a picture of the resources evolutions and their possible effects on the product flow. In addition to the evolutions, this picture takes the controlled system constraints into account. As the expert, the control law synthesis method must have access to a controlled system model, the initial state of the product flow and the resources (PF&R) and a model of the objectives. For discrete event systems, the three synthesis approaches presented below are compared on the basis of the system model, the initial system state and the objectives.

To pilot the manufacturing system, the supervised control [1] is based on the Ramadge and Wonham theory. In this approach, the supervised system is a set of resources with their local control law given by an expert. From the automata system model, which is not obtained with a modeling methodology, the supervisor limits the system behavior to the most permissive, which respects the objective specifications. They must also be expressed with the automata; Thus they cannot express the optimization criteria. This method then does not guarantee performance in productivity and quality criteria.

The automated control code synthesis [5] is based on the condition systems that are closely related to Interpreted Petri Nets. A component model representing a physical component is described with this formalism as well as the associated taskblocks. In a condition system, the output conditions result from marked places. For each output condition of the component model and from any mark, a taskblock defines all the control laws to reach the marking for which the output condition is true. Except for the low level, the set of control laws cannot be built because of its size. At the upper level for the coordination of the taskblock, the authors propose to use the supervisory control approach.

According to French [3], the general scheduling problem is to find a sequence of operations, carrying out one or several tasks, which is a feasible schedule, and optimal with respect to some performance criterion. An operation represents the use of a resource to produce a service. Contrary to two previous synthesis approaches, the scheduling is not a method with one algorithm and a controlled system model that is exactly defined with the modeled constraint point of view. Each scheduling problem defined by controlled system model and objectives (task(s) and criterion) has a particular solving algorithm. Thus in the context of the reconfiguration, if a disruption modifies the scheduling problem, then a new algorithm must be applied. This is deeply penalizing for the expected reactivity. However, the controlled system model is always formalized by a set of operations with the constraints on or between the operations like precedence, preemption, splitting, and very rarely condition(s) on the resource state, etc, and with the resource properties like disjunctive, renewing, etc.

4 Aim of paper

In comparison to other synthesis methods, scheduling methods are interesting from the point of view of the optimization criteria and the controlled system model with operations. In spite of the many works in the scheduling field, none of them has a very low-level controlled system view with so many operation constraints. First, to use the scheduling method to synthesize the control laws, the controlled system model must integrate all the constraints. Then, from data (controlled system model, initial PF&R state and the objectives), the automated control law synthesis is developed. As in the scheduling methods, the first synthesis stage generates a precedence graph. And the second synthesis stage translates this graph to one of the CEI 61131-3 languages, as in Figure 2.

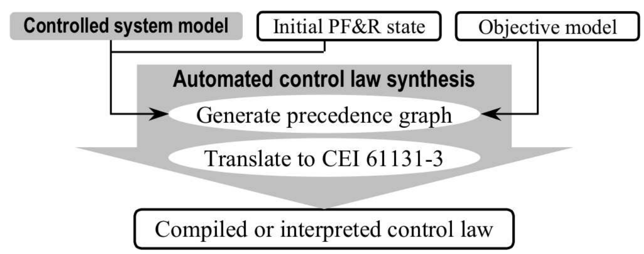

Figure 2. Aim of paper

5 Controlled system model

To achieve its basic function, the manufacturing system must control the product flow evolutions. But its control system does not pilot directly the product flow evolutions, it pilots the resource evolutions. Thus, we propose to represent the controlled system model by the set of the product flow evolutions, the resource evolutions and the links between the product flow and resource evolutions. With these links and from product flow evolutions respecting the objectives, we can deduce the necessary resource evolutions, and in this way we can automatically synthesize a control law. If the product flow and resource evolutions are modeled in a same operation, its model defines this link.

5.1 SAPHIR

This section presents an application example based on the loading system (see Figure 3) of the research platform Saphir of the Laboratoire d'Automatique de Grenoble, in France. This platform is dedicated to the assembly of camshafts. A rotating storage been made up of a tray with four places is used to receive until six different kinds of products. These products are identified by a weight identification system. Once a product has been identified, a central conveyor drives it to a sorting device. So, a robot takes the different products to assembly them. A shopworker is charged to fill the rotating storage and to empty the assembly station.

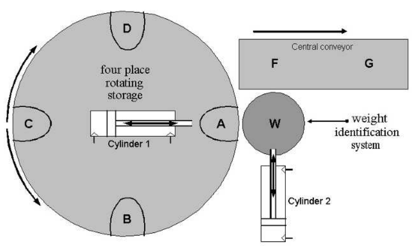

Figure 3. Loading system of SAPHIR

5.2 Preliminaries

From this instance, if we consider a control law that is able to put identified products on the central conveyor, two sets of the operations can be characterized. In the first set, the operations have an effect on the resource state and on the product flow state (i.e. "extend cylinder 1" that moves the product from A on the tray to W on the weight identification system). In the second set, the operations have only an effect on the resource state (i.e. "retract cylinder 1"). According to the set that an operation belongs to, the reason for its existence and for its position in the control law depends on the operation goal. The operations of the first set put the product flow in the expected state. But to run these operations, the resources must sometimes be in a particular state. The operations of the second set make it possible to put the resources in this state.

5.3 Operation model

First, the "extend cylinder 1" (EC1) operation model with an effect on the product flow (EC1 moves the product from A to W) is submitted. By a simple observation of all the possible behaviors when C1 extends, this model is refined, and finally the general operation model with its properties is proposed.

To choose the operations to reach the final product state given by the objectives, it is required to have information on the effect on the product. But to have this effect, EC1 must also modify the cylinder 1 (C1) state from the retracted position to the extended position. The effect on C1 defines the controlled resource and its evolution to have the expected effect on the product. Finally, the operation model must represent not only the effect on the product state but also the effect on the resource state running the operation. An effect is defined by the initial, intermediate and final states which respectively correspond to the states before, during and after the operation running. The effects on the resource and on the product is presented in the "Effects" column in Figure 4. The initial states of C1 and the product are the conditions for starting EC1. And the intermediate and final states are the result of the EC1 running.

The effect modeling does not take the security and environmental constraints into account. Indeed, this model does not guarantee collision between the cylinders will be avoided. For that, C1 can extend if cylinder 2 (C2) is retracted before and during the EC1 running. If we want to avoid product falls from the weighing post, other constraints must be respected: no product between A and W before the EC1 running, and no product in W before and during the EC1 running. The EC1 running ends accurately only if constraints on the other resources and the product flow are respected before (pre-constraints) and during (constraints) the EC1 running, as in the associated constraints in Figure 4.

To optimize the control law, the quantifiable criteria are necessary like the time cycle, the cost, etc. To assess these criteria, the operation modeling must give the operation features like the operation cost and duration. For reasons of concision, only the operation duration is taken into account in this paper. The EC1 model defining the effects, the constraints and the duration characterizes one behavior when C1 extends.

Extend Cylinder 1 (EC1)

Duration: 1 s

Effects on resource	Effects on product	Associated constraints on other resources	Associated constraints on product flow
C1 retracted pos.	P in A	C2 retracted pos.	No P bet. A and W No P in W
C1 intermediate pos.	P bet. A and W	C2 retracted pos.	No P in W
C1 extended pos.	P in W		

Figure 4. Partial "extend cylinder 1" operation

The controlled system model must represent everything the controlled system is capable of doing [2]. For example, C1 can also extend without any effect on the product. Contrary to the behavior with effect on the product, no product must be in A, but a product can be in W. So for the same evolution of the resource state (from the retracted position to the extended position), two behaviors exist depending on the initial state of the product flow (with or without a product in A). In this case, we can model either each of the behaviors with one operation or the two behaviors with only one operation. If we consider that an operation defines one behavior, then there are the EC1_1 operation without any effect on the product and the EC1_2 operation with an effect on the product. The information on the effect on C1 and on the associated constraints is modeled in EC1_1 and EC1_2. The modeled information is redundant which increases the number of operations. So, the two behaviors must be modeled in the same EC1 operation. To avoid information redundancy, the two behaviors must be followed by a basic behavior which will be completed with an extra behavior. Contrary to the effect on C1, if we observe the EC1 model in Figure 4, we cannot distinguish the associated constraints. So, the constraints associated with the effect on the resource must be separate from the constraints associated with the effect on the product. Finally, EC1 is modeled by a basic behavior, which defines the effect on C1 with the associated constraints, and by an extra behavior, which defines the effect on the product and the associated constraints. The EC1 has a third behavior. Indeed, C1 can also be extended with an effect on the product if the product is initially between A and W. Thus, the complete EC1 model has two extra behaviors, as in Figure 5. Generally speaking, the operation model can have several extra behaviors. A behavior will be obtained if the condition on the initial state (of the resource or the product) is true and the associated pre-constraints and constraints are respected. The pre-constraints and the constraints will be noted (pre-)constraints in this paper.

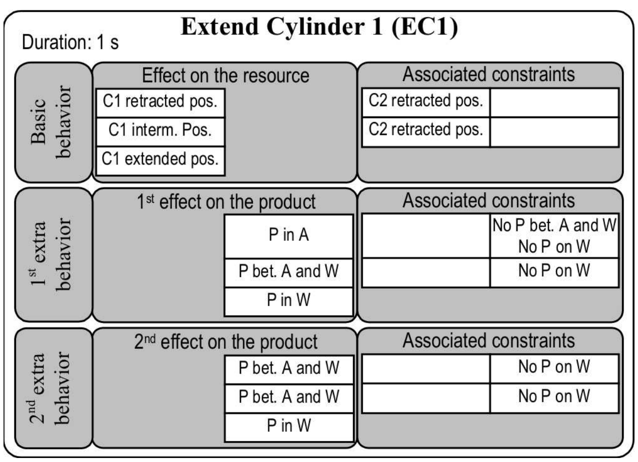

Figure 5. Complete “extend cylinder 1” operation

The EC1 running will have an effect on C1 and no effect on the product flow if the basic behavior can be obtained and if all the extra behavior conditions are false: no product in A and no product between A and W. The EC1 running will have only the first effect on the product, if the basic behavior and the first extra behavior can be obtained and if the second extra behavior condition is false (no product between A and W).

The negation of this condition is one of the first extra behavior pre-constraints on the product flow. So, this second extra behavior condition and this first extra behavior pre-constraint cannot be true at the same time. This condition and this pre-constraint are considered incompatible. The first and second extra behaviors cannot be obtained simultaneously. But a set of extra behaviors can be obtained simultaneously if both the set of the conditions and the pre-constraints are not incompatible, and if both the set of the intermediate product states and the constraints are not incompatible. For instance, rotating the tray a quarter turn clockwise is an operation offered by the tray resource. If there are four products on the four places, then this operation will have simultaneously four extra behaviors (each product will rotate a quarter turn).

To finish the operation modeling, we are now interested in the operation without any effect on the product flow. With the suggested operation model, these operations can be also modeled. They have only a basic behavior and they do not have any extra behavior.

In the preliminaries, we have highlighted two sets of operations: with or without effects on the product. This operation classification is not sufficiently accurate towards the product flow state. Some state variables of the product flow result from the product specifications like geometric forms, etc. And other state variables of the product flow result from resources and their position in the manufacturing system. According to the state variables modified by an operation running, we have identified three kinds of operations, presented in Figure 6.

			Operation **1**	Operation **2**	Operation **3**
State variables of	product flow	∈ product specifications	***are*** modified	***are not*** modified	***are not*** modified
		∉ product specifications	***can be*** modified	***are*** modified	***are not*** modified
	resources		***can be*** modified	***can be*** modified	***are*** modified

Figure 6. Three kinds of operations

5.4 Information on the specific states

When a product is located in F on the central conveyor, no operation offered by the loading system can evacuate this product. Nevertheless, the “No product in F” state is a pre-constraint and a constraint of the “extend C2” operation with an effect on the product. So an operation sequence to reach this state does not exist. To avoid searching for such sequence, information on this specific state must be represented in the controlled system model. The environment (sensors, other control systems, human operators) of the loading system provides this information. Thus, the controlled system model is the set of operations with a specific state list like “No product in F”.

6 Automated control law synthesis

An operation is an elementary brick. The assembly of two bricks creates a link between these bricks. This link is not modeled in the controlled system model, but it must be found by a synthesis algorithm according to the objectives. Finally, a control law is a particular assembly of the elementary bricks. The proposed controlled system model does not represent all the possible control laws. Then, from the initial PF&R state and the objectives, the global synthesis method builds the acceptable control laws. Afterwards, it searches for the optimal control law according to the criteria. But the set of the acceptable control laws is too large to be built. So, we propose an automated control law synthesis part by part with local optimizations: The first synthesis stage generates a precedence graph, the second synthesis stage translates this graph to one of the CEI 61131-3 languages.

6.1 Initial state and Objectives

Before defining the initial state and the objectives, two details must be clarified. Firstly, the loading system does not have any operation 1. So in the example presented, there is not any State Variable of the Product flow belonging to the product Specifications, noted in this paper SVP∈S. Secondly, the operator can add or remove the products in C. He does not inform the control system about his actions. And the product presence (or absence) in the tray is known when the product arrives in A. So in C, the value of the position variable is unknown. Now the initial state can be defined. For the state variables of the product flow not belonging to the product specifications (SVP∉S), the values are: no product in A, W, F and between all the points; unknown in B, C and D; non-identified for a product in B, C and D. For the state variables of the resources (SVR), the values are: C1 in the retracted position, C2 in the retracted position, The tray in the indexed position with a null speed.

As soon as possible, an identified product must be put in F on the central conveyor. The final state of this objective defines only the values of some SVP∉S: no product between W and F, between F and G; product in F, identified for the product in F. The optimized criterion is the time cycle.

6.2 Synthesis algorithm

The first synthesis stage (generate a precedence graph) is split into two algorithm steps: generate a graph for one product and generate a graph for several products. To reduce the complexity of the first algorithm step, the first sequence with only operations 1 is defined to modify the only SVP∈S from the initial state to the final state. As operations 2 cannot modify SVP∈S (see Figure 6), they can be added at the first sequence to respect the conditions and the (pre-)constraints on SVP∉S to run the operations 1 without modifying SVP∈S. And finally as operations 3 cannot modify SVP∈S and SVP∉S (see Figure 6), they can be added at the second sequence to respect the conditions and the (pre-)constraints on SVR to run operations 1 and 2 without modifying SVP∈S and SVP∉S. For reasons of concision, only the building sequences of operations 2 are presented. The sequences of operations 1 and 3 are built with the same method.

Before building the sequences of operations 2, we must know where to integrate them in the first sequence. To run the i operation, the conditions and the (pre-)constraints on SVP∉S must be respected. If the SVP∉S values after the i-1 operation respect them, then no operation 2 must be added. Otherwise a sequence of operations 2 are added between the i-1 operation and the i operation. To find the sequence of operations 2, an automata is built and the best way from initial state to marked state is sought according to the optimized criterion. An automata state is characterized by the values of all SVP∉S. And the event associated with the arcs is an operation 2. The SVP∉S value after i-1 operation defines the initial automata state. From this state, all the operations 2 able to run are sought. Then, for each operation 2 found, an automata state after operation 2 is calculated from the operation model. The same technique is applied at each new state, except if the new state is a marked state which respects the conditions and the (pre-)constraints on SVP∉S of the i operation. For the loading system, there are not any operations 1. To find the only operation 2 sequence, the automaton built has fifteen states, twenty-five arcs and two marked states. Part of it is presented in Figure 7. Only the modified SVP∉S for this part are written in the states. From this automaton, the TTC, EC1, IP and EC2 operation 2 sequence is found (see Figure 8).

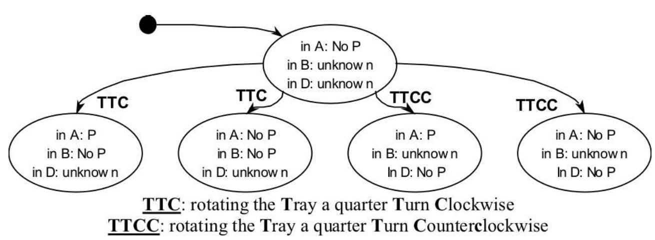

Figure 7. Partial automata of operations 2

The insertion of the sequences of operations 3 in the second sequence is based on the same method as the addition of the sequences of operations 2 in the first sequence. The only exception is for a sequence of operations 3: it is added before the j operation in the second sequence, but it is not necessarily added after the j-1 operation. Indeed operations 3 do not have an effect on the product; They can be run at the same time as operations 1 or 2. To minimize the time cycle, we search for the earliest time when each operation 3 can be run. A precedence constraint is added before operation 3, as in the RC1 operation 3 in Figure 8. At the end of the first algorithm step, the precedence graph for one product is built, like the grey area in Figure 8.

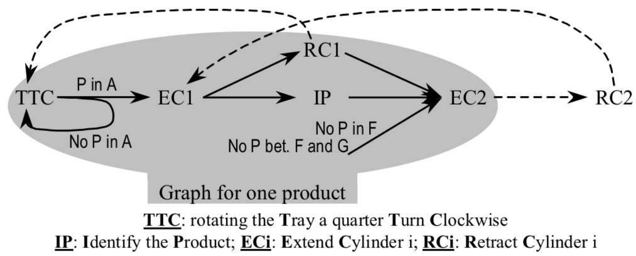

Figure 8. The precedence graphs

To build the graph for several products, the second algorithm step considers the possibility of applying again the previously defined precedence graph for a second product. To run the operations of the second graph at the earliest, operations 3 and the precedence constraints are added between the operations of the first and the second graphs. For instance, the TTC operation for the second product can be run at the earliest after the RC1 operation for the first product. But the EC1 operation for the second product cannot be run after the last operation (EC2) of the first graph because the EC1 pre-constraints are not respected. The RC2 operation 3 must be added between EC2 for the first product and EC1 for the second product. (see dotted line in Figure 8). To end the second algorithm step, the two graphs are merged; For instance the TTC operations for the first and the second products are merged and so on (see Figure 8).

For space reasons, the second synthesis stage (translate the precedence graph for several products to SFC) cannot be submitted in detail in this paper. However, in Figure 9, we show that the translation of the precedence graph to SFC is feasible. Each operation is associated with an action of one stage. And the SFC control structure is defined by the precedence constraints.

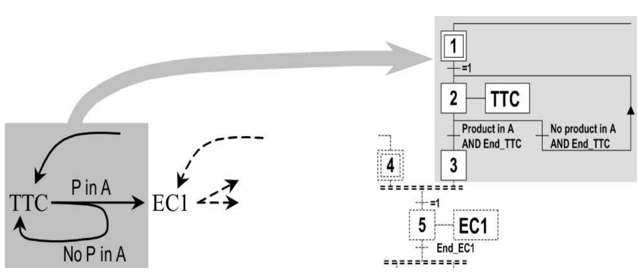

Figure 9. Translate to CEI 61131-3

7 Conclusion and Future Works

In this paper, problem of controlled system modeling and the automated control law synthesis is dealt. First, each resource is described by all the offered operations. The structure of a proposed operation model is generic. This model is composed of the basic behavior and of no or several extra behaviors. A behavior is defined by an effect and the associated (pre-)constraints. All the operations with information on the specific states define the model of the controlled system. Then, from the initial state of the resources and the product flow, the first synthesis stage defined a precedence graph according to the objectives. Finally, the second synthesis stage translates the precedence graph to one of the IEC 61131-3 languages.

Future works will first focus on a the validation of our synthesis approach in an industrial context inside a PLC programmers department. Second, based on such a validation, the synthesis algorithm will be integrated in PLC to test the dynamic reconfiguration abilities.

References

[1] F. Charbonnier, H. Alla, and R. David, “The supervised control of discrete-event dynamic systems”, *IEEE Transactions on Control Systems technology,* Vol 7, No 2, pp. 175-187, March 1999.

[2] A. De Bonneval, M. Combacau, and M. Courvoisier, “One the use of real-time diagnosis outcomes for recovery purposes in F.M.S.”, *Revue Européenne Diagnostic et Sûreté de Fonctionnement,* Vol 4, No 2, 1994.

[3] S. French, *Sequencing and Scheduling, New York:* Halsted Press, 1982.

[4] S. Henry, E. Zamaï, and M. Jacomino, “Decisional Requirements for Supervision, Monitoring and Control Structures” Proc. IEEE Int. Conf. on Computational Engineering in Systems Applications, France, June 2003.

[5] L.E. Holloway, X. Guan, and al, “Automated Synthesis and Composition of Taskbloks for Control of Manufacturing Systems”, *IEEE Trans. On Systems, Man and Cybernetics, Part B,* Vol 30, No 5, October 2000.

[6] H. Mendez, E. Zamai, and B. Descotes-Genon, “Quality, Productivity, Security and Ecological Criteria for Synthesis of monitoring laws”, *Quality Engineering*, To be published, 2004.

[7] http://www.value-eng.org/catalog_monographs.php