Internal Models and Negotiations in the Extended Scheduling Problem

Florin-Gheorghe Filip and Alexandru-Dan Donciulescu
Research Institute for Informatics
8-10 Averescu Avenue,
71316 Bucharest
ROMANIA

Abstract: This paper presents the incremental development of a DSS meant to support various management and control tasks in the continuous process industry. It refers the state-of-the-art in modelling for production scheduling and discusses related problems in this industrial milieu. In particular, the negotiation problem of the *external* flows from material suppliers/ to clients is proposed to solve and the extended scheduling problem is introduced.

Keywords: DSS, intelligent control, incremental design, modelling, process industry, scheduling, virtual manufacturing

Dr. **Florin-Gheorghe Filip** was born in Bucharest, Romania, in 1947. He graduated in Control Engineering and took his Dr.Engng. Sci. for his contributions to hierarchical control from the Polytechnical Institute of Bucharest in 1970 and 1981 respectively. In 1970 Dr. Filip joined the Research Institute for Informatics, Bucharest. He was the head of Advanced Decision Support Systems Research Laboratory. Since May 1991 Dr. Filip has been general director of the institute.
In 1974 he was an invited research fellow at Swedish universities (Chalmers TH Goteborg, Uppsala Universität, TH i Lund, KTH Stockholm). In 1995 he was trained in research management at FhG IITB Karlsruhe, Germany.
In December 1991, Dr. Filip was elected as a corresponding member of the Romanian Academy. He is a member of the IT Section of the Romanian Academy.
Dr. Filip is the author/co-author of over one hundred thirty technical papers published in contribution books and international scientific journals such as IFAC J. Automatica, Contr. Engng. Practice, Large Scale Systems, Computers in Industry, Modelling & Simulation, Syst. Anal. Model. and Simul.,etc. He is the co-author of two books: "Cybernetics, Automation and Informatics in the Chemical Industry" (1979), and "Hierarchical Real-time Systems" (1986).
His main current scientific interests include: modelling and simulation, hierarchical optimisation and control of large scale systems, decision support systems, integrated plant control in process industries and discrete part manufacturing, information systems in public administration and IT technology transfer.

Mr **Alexandru-Dan Donciulescu** graduated in Control Engineering from the Polytechnical Institute of Bucharest in 1972. He is preparing a doctoral thesis in Applied Informatics. Since 1972 he has been with the Research Institute for Informatics, where currently he is head of DSS Research Laboratory. His main fields of interest are in analysis and mathematical modelling of complex information systems; design and programming of DSS-like software products; DSS, including integration of expert systems; end-user interfaces design. Mr Alexandru-Dan Donciulescu is author/co-author of more than 60 papers presented at national and international workshops, symposia, congresses. He also authored/co-authored over 50 papers published in national and international journals. He is associate professor in computer programming and DSS.

1. Introduction

Various opinions were presented on the development approach recommended when building information systems: traditional life cycle (waterfall) methods are opposed new incremental (prototype-based) ones [Coad and Yourdon,1990]. In the particular case of **Decision Support Systems** (DSS), the migration from an initial application area their developers meant them for to new applications on client pressure [Keen,1980] is often experienced; the incremental (middle on) approach appears to be advisable.

This paper aims at illustrating this issue based on the authors' practical experience in developing and applying in industry the DISPATCHER family, a DSS line of application software for process industries, which was developed over a fifteen-year period, and upgraded several times with different technologies and methods [Filip,1995]. This was originally meant to support short-term production scheduling [Filip et al, 1985]. Then it was found out that it could also be used in tank sizing (design) problems, or in supporting applications such as order entry and material purchasing. This process and its development allow for a certain integration into the organization. The latter applications imply *negotiation* processes.

The paper is organized as follows. First, some trends in **information systems** (IS) development are reviewed with special emphasis on incremental design of a DSS. Then a typical application task of steadily evolving intelligent DSS in the particular domain of the continuous process industries is presented. The extended scheduling problem (ESP) is introduced next. ESP is solved by means of a *negotiation* algorithm to handle raw material inflows and/or product outflows .

2. Trends in Information Systems Development

Two of the various factors having a significant impact on modern information systems are presented as providing the background of the developments to be further described : a) the evolution of the enterprise model, and b) the *ontogeny* (development process) of IS.

The business processes tend to be ever more interconnected and consequently the virtualisation and extension of the enterprise become more and more apparent. A particularly important idea is expressed by J.Browne from CIMRU, on extending the current concept of manufacturing. His model justifies the need for a virtual extended enterprise "comprising the currently used description plus raw materials and parts [ingredients] suppliers and its customers to assure the utmost flexibility and customers response capabilities" [Williams et al, 1995]. As a matter of fact, this presupposes a tighter collaborative work and decision-making by the people within the "factory walls" with those outside the "factory walls".

Business process reengineering (BPR) and new enterprise paradigms such as "extended", "learning", "bionic", "liberated", "re-engineering", etc. [Norman,1994] require ever more integrated IS. **Computer integrated enterprise**, CIE and **computer integrated business**, CIB [Raulefs,1994] extend the support scope and help the integration into the enterprise. Those IS are apparently less technology driven but ever more based on mission and business perspective. Grando [1993] has predicted some of the significant transitions which the IS development was to go through such as a) strategy (aimed at maximising re-usability) and b) process as such (from *waterfall* approach to *incremental* design). Concerning the development process, Keen's remarks, made many years ago [1980], on the specific context of DSS are of a particular relevance to this paper. He noticed that the "*final system must emerge through an adaptive process of design and usage*". We invoke only two reasons why this process should go on a) the "*users concept of the task or decision situations will be shaped by the DSS, that stimulates learning and new insights, which in turn stimulates new uses and the need for new functions in the system; the unpredictability of DSS usage reflects learning, which can be exploited only if DSS evolves in response to it*" and b) the "*intended users have sufficient autonomy to handle the task a variety of ways...; that suggests the user shapes the DSS*".

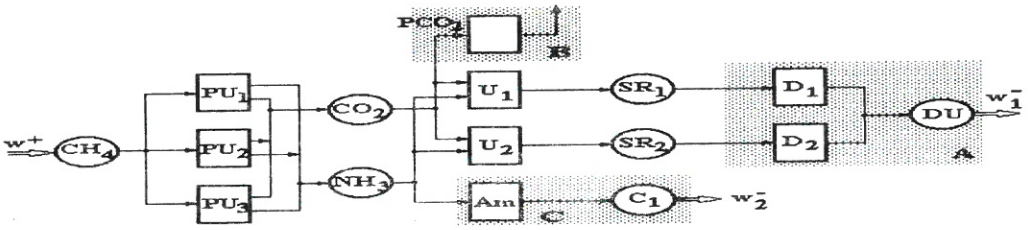

Figure 1. A Continuous Process Industrial Complex

3. Evolving DSS for Continuous Process Industries

3.1 Application Area Description

Consider the typical continuous process industrial system in Figure 1. This may be viewed as a set of **processing units** (PU) interconnected through buffer **stocking reservoirs** (SR). Each PU is normally fed from/feeds some up-/downstream. Process- to - process links are quite possible [Filip,1990]. A certain PU runs at a certain time in one (out of a finite number) of the **operating regimes** (OR) defined by a technological prescription normally characterised by specific delays and transfer ratios, and on I/O channels and some upper and lower physical limits of the main material flow (u_M, u_m). At the same time, the SRs are characterised by their physical stock limits (s_M, s_m). Raw- material, ingredient and utility inflows (f^+) feed one/several SR/PUs and product outflows (f^-) drain SRs (in most cases) and PUs. Such systems can be found in more continuous process industries (pulp and paper mills, fertilizer plants) and related fields (water sytems).

Based on the client learning process and the evolving usage several tasks have been sequentially defined in an industrial milieu such as a) **production scheduling** (PS), b) order entry, c) raw -material planning, d) SR sizing (an off-line design problem), and e) operator training through simulation.

3.2 Production Scheduling: Problem Manipulation

The PS consists of two interrelated subtasks namely: a) **operation configuration** (OC), and b) **flow scheduling** (FS). OC is to choose (over a number of KF intervals in the proximate time horizon) the sequence of ORs for PUs; it is a qualitative problem. FS specifies (over the same time horizon [0,KF]) the feasible (possibly optimal) sequence of production rates of PUs for given sequences of vectors of inflows and outflows ($\mathbf{f}^+(k)$, $\mathbf{f}^-(k)$; k=0,KF). In FS there are several objectives (which sometimes contradict one another) : a) the sequence $\mathbf{f}^-(k)$ must be observed to meet client delivery plans, b) the sequence of stock vector values in SR should be under some economical values $\mathbf{s}_u$ to reduce dormant material value even though tanks allow higher values $\mathbf{s}_M$, c) safety stocks $\mathbf{s}_l$ are desirable to avoid draining-down SRs by downstream PUs, d) PUs production rate vector should be within some minimal consumption ranges ($\mathbf{u}_u$, $\mathbf{u}_l$) even though their actual physical ranges ($\mathbf{u}_M$, $\mathbf{u}_m$) are in fact wider, etc.

To handle the FS problem an optimal, discrete time tracking problem, with constrained variable parameters, was set and a fast and robust solving algorithm was proposed [Filip et al,1985]. It considered production rates as controls (**m**) and mapping stock values as states (**x**), characterised by adequate sequences of desired values (**md**(k), **xd**(k+1); k=0,KF) and limits (**mu**, **ml**, **xu**, **xl**). A refined model, together with several examples, taking into account various specific application characteristics (in particular a sparse structure with relatively constant parameters) is presented in Filip [1990] . This model is in fact the *conceptual* layer lying between the external (*representation)* layer (addressing the user), and the internal (*performance*) layer (improving the algorithm performance at a low cost). All the features are contained in DISPATCHER, a DSS family developed using different technologies and platforms, and implemented in various industries, over a fifteen-year period [Donciulescu and Filip, 1994]. To improve system intelligence, a *declarative model* of a *knowledgeable operator* (DMKO) was introduced to support model building, validation and experimentation, mainly addressing operator training applications.

In the original SP formulation, inflows and outflows are viewed as known, fixed sequences of disturbances (**w**(k), k=0,KF). If the solution restriction space is empty in SP, or for pure order entry / purchasing applications, those *external* flows should be considered as decision variables and represented accordingly.

4. Extended Schedule Method

Decision -maker's learning as well as some extremely unbalanced operation conditions generated by unfavourable combinations (UC) of initial stock values (**x**(0) together with poorly planned sequences of *external* flows ($\mathbf{w}^+(k)$ and $\mathbf{w}^-(k)$, k=0,KF), led to new application of the system such as a) *extended scheduling problem* (ESP) and b) order entry or material purchase planning. The basic idea is to consider some or all supplies and product consumers as *virtual extensions* of the manufacturing model, and to use SP as a tool to provide and evaluate a negotiate solution between the system itself and its virtual extensions. In the former case the material supply and/or product clients are considered by sheer necessity to prevent failures and to meet the order due dates to UC. In the latter case the optimisation model is merely used as a simulation tool to evaluate the possibility of certain orders being entered in parallel with an adequate planning of raw- materials' check-in. Naturally, the model horizon and the time interval are larger in case b (model parameters differ as well) although the model keeps the same.

4.1 Necessary Conditions

Any feasible (possibly optimal) solution of the SP asks that the following conditions hold for a given **operation condition** (OC) defined by an initial condition $\mathbf{x}_o$ and the disturbance sequences ($\mathbf{w}^+(k)$, $\mathbf{w}^-(k)$; k=0,KF).

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{B}(k)\,\mathbf{u}(k) + \mathbf{w}+(k) - \mathbf{w}^-(k) \quad (1)$$

$$\mathbf{xl}(k) \leq \mathbf{x}(k) \leq \mathbf{xu}(k)\,; \quad k=1,KF+1 \quad (2)$$

$$\mathbf{ul}(k) \leq \mathbf{u}(k) \leq \mathbf{uu}(k)\,; \quad k=0,KF \quad (3)$$

Necessary conditions (NC) for feasible OC can be derived and tested during the problem validation phase. For example for the ith SR drained by a certain product outflow w^-_i the following necessary condition must hold as to prevent *draining down* situations:

$$x^*_i(k) \geq xl_i;\ k=1,KF+1 \quad (2.1)$$

```
While   $\exists (i,k) \left[x_i(k) \notin \left(xl_i, xu_i\right)\right]$ do
    For   i=1 to $n_i$
        if $\forall k \left[x_i(k) \in \left(xl, xu\right)\right]$ then repeat
          else case
            : $w_i^-$ is attached: go to 1
            : $w_i^+$ is attached: go to 2
            : no $w_i^-$, $w_i^+$: go to 3
            endcase
          1 negotiate with client a possible [larger] variation range $\left[\varepsilon_i^-, \varepsilon_i^+\right]$
            create new $PU_i$
                • card $\Re_i \leftarrow$ card $\Re_i + 1$
                • $b_{ia} \leftarrow -1$
                • $ul_{ia} \leftarrow w_i^- - \varepsilon$ ; $ud_a \leftarrow w_i^-$ ; $uu_a \leftarrow w_i^- + \varepsilon_i^+$
                • $w_i^- \leftarrow 0$
            go to 4
          2 negotiate with supplier ...
            go to 4
          3 negotiate with unit plant operators a further relaxation of limits
        endif
    4 repeat
    resume optimisation/simulation
repeat
```

Figure 2. Negotiation (co-operative) Procedure (simplified)

$$x_i^*(k) = \sum_{t=0}^{k-1}\left[\sum_{j\in\Im_i} b_{ij}uu_j(t) - \sum_{j\in\Re_i} b_{ij}ul_j(t) - w_i^-(t)\right] + x_i(0) \quad (4)$$

where x^*_i is the *most favourable* evolution of SR_i for a given OC.

$\mathfrak{I}_i$ and $\mathfrak{R}_i$ are input and output sets of *internal* flows of Sr_i.

Other three similar necessary conditions can be set to test possible overflow or draining situations for all the SRs connected to the *external* flow of a given OC. Of course, those necessary conditions are based on locally most favourable values of internal flows and might not be sufficient as they are mutually conflicting sometimes. However, they can allow for early detection of unfeasible OCs and for inceptive **corrective actions** (CA). A sample of some simplified rules of the production system used in the DISPATCHER family to support the problem validation phase, which must precede optimisation or simulation process, is given in the sequel.

R27

IF 27A (Test (2.1) fails for SR_i), AND

27B (there is a stocking margin:

$x_{m_i} < xl_i$), AND

27C ($x^*_i(k) \geq xm_i$),

THEN

Define a new lower limit of the stock in the model

$xl_i(t) \leftarrow x^*_i(k)$, for t = k.

R28

IF 27A, AND

28.B (there exist physical production rate margins on some upstream neighbour PUs:

$u_{M_i} > uu_j;\ j \in \mathfrak{I}'_i \subset \mathfrak{I}_i$),

THEN

cautiously increase the upper limits

$uu_j(t) \leftarrow 1/2\ (u_{M_i} - uu_j)$, for t=0,k-1, $j \in \mathfrak{I}'$.

4.2 Relaxation Algorithm

If NCs are not met (despite of all CAs) or feasible solutions are not reached (even though NCs hold), it will be features for automatic(internal) or manual (co-operative) *manipulation* of external flows to be described during the model experimentation phase. While the former one is normally used to provide a first action alternative, the latter is normally meant for supporting interactive negotiation with suppliers and/or product consumers, and allows for some co-operative work. Let us first consider the *automatic* manipulation. Basically, it consists in building an "internal model" of some external I/O (material/product) flow sequences. They have a priori been agreed on by suppliers/clients as being relaxable within specified margins. The procedure is said to be automatic because of its embedment into the "model building" component of the KBS part of the DSS (Filip, Roberts and Zhang, 1992). A sample rule is given below:

R30

IF 27A, AND

30B (w^-_i was declared as negotiable),

THEN

Define a new "virtual "PU_a characterised for each time k by:

$$ud_a(k) \leftarrow w^-_i(k)$$
$$uu_a(k) \leftarrow w^-_i(k) + \varepsilon^+_i$$
$$ul_a(k) \leftarrow w^-_i(k) - \varepsilon^-_i,$$

where ε^+_i and ε^-_i are permissible variations of the $w^-_i(k)$ stored in the system.

Set $w^-_i(k) \leftarrow 0$.

In the *manual* (co-operative) case, the same basic idea is maintained, but the **decision - maker** (DM) can dynamically allocate variation ranges to the new variables, based on his knowledge and on feedback from negotiation partners. The algorithm is given in Figure 2.

This cannot be an automatic procedure (even though for simplicity it has been described somehow algorithmically): it implies real-time interactions with suppliers and clients. Those may be viewed as virtual parts of the plant extended beyond its "four walls" (Figure 3). At the same time the procedure is a co-operative one and assumes the "horizontal" (at the same decision level) communication and empowerment of middle level decision-makers to solve "crisis" situations, working as a "virtual team".

The procedure shown in Figure 2 was started when some technical conditions of the production plant could not be satisfied, for example stocks cannot be within permissible limits for a given set of initial conditions and sequences of I/O material flows.

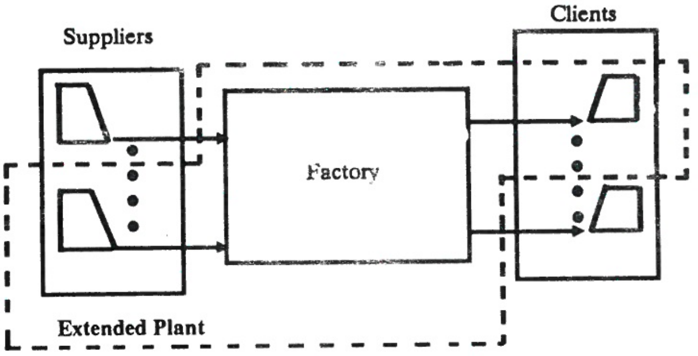

Figure 3. Extension of the Factory

One can expect that the procedure is initiated not only by the "production" people but also by their "clients" (representing either further processing plants or product distributors) to better respond to market trends.

5. A Numerical Example

Consider the system in Figure 1, which is a 2-line urea section of a fertilizer complex. The methane supplies with given quantities w^+ to three plants producing ammonium, collected in the NH_3 tank. CO_2 also results, which can be further used in the urea processing, or in other sections of the complex; the unused CO_2 will be evacuated in the atmosphere. The B area checks on maintaining an empty dummy tank CO_2 (automatically created by the system). U1 and U2 process NH_3 to produce urea. Also NH_3 is requested by other sections of the complex, in fixed quantities w^+_2; The plant Am and the area C model these consumers. SR_1 and SR_2 contain urea of the same quality, produced on the two lines. The requested urea quantities are w^-_1. The area A in Figure 1 models a mechanism to deliver urea between the two lines. A 5 time intervals horizon is considered. The recipes for the plants are simplified to the components shown in Figure 1; the numerical values are given in Table 3. Tables 1,2, and 4 show the operational data used in the example. Empty initial stocks are initially considered for all tanks in the experiments ($x_0 .= 0$).

When running the optimisation with the data in Tables 1-3 and part 1 in Table 4, no solution is found. The reason for this is that the needed urea w^-_1 and ammonium w^-_2 ask for as much NH_3 as no one of PU_1, PU_2 and PU_3. would allow. To solve this problem, any of the w^- may be selected to be negotiated. Table 4, part 2 gives the NH_3 possible deliveries (w^-_2 are negotiated) when w^-_1 original values are preserved. When the deliveries w^-_1 and w^-_2 are both forced to low values , as for example in Table 4, part 3, the original w^+ imposed flow (first line in Table 4) causes the SR tanks' overloading. If no other urea deliveries are expected, the w^+ flow is to be reduced, and w^+ turns to be negotiated (the resulting values are given in part 3 of Table 4).

Table 1. Operational Data for Plants

Plant	Min	Desired	Max
PU_1	0	90	100
PU_2	0	90	100
PU_3	0	80	80
PCO_2	0	0	500
U_1	0	120	150
U_2	0	60	80
Am	0	160	200
D_1	0	0	500
D_2	0	0	500

Table 2. Operational Data for Tanks

Tank	Min	Desired	Max
CH_4	0	0	100
CO_2	0	0	10
NH_3	0	100	500
SR_1	0	100	500
SR_2	0	50	500
C_1	0	0	10
DU	0	0	10

6. Conclusions

Our negotiation solution assumed that each *external* flow was related to a single SR. In fact in the real DSS the external flow various attributes such as destination/source (SR or PU or both), constrained value (instantaneous, integral or both of them), shareability (one or several sources/destinations) allow for a larger modelling diversity to be automatically treated by a simple rule based production system [Filip, Roberts and Zhang, 1992].

Table 3. Plants - Tanks Interconnection

Plant/ Tank	PU_1	PU_2	PU_3	PCO_2	U_1	U_2	Am	D_1	D_2
CH_4	-0.9	-1.3	-1.4	0	0	0	0	0	0
CO_2	1.3	1.4	1.4	-1	-0.6	-0.5	0	0	0
NH_3	1	1	1	0	-0.7	-0.7	-1	0	0
SR_1	0	0	0	0	1	0	0	-1	0
SR_2	0	0	0	0	0	1	0	0	-1
C_1	0	0	0	0	0	0	1	0	0
DU	0	0	0	0	0	0	0	1	1

Table 4. Inflows and Outflows

	Time1	Time2	Time3	Time4	Time5
Part 1 Original values					
w^+	360	360	300	360	360
w^-_1	0	0	500	220	180
w^-_2	200	200	200	200	200
Part 2 w^-_2 negotiated					
w^+	360	360	300	360	360
w^-_1	0	0	500	220	180
w^-_2	79	142	122	121	135
Part 3 w^+ negotiated to cope with low outflows					
w^+	167	94	68	91	150
w^-_1	0	0	10	0	0
w^-_2	10	10	20	10	0

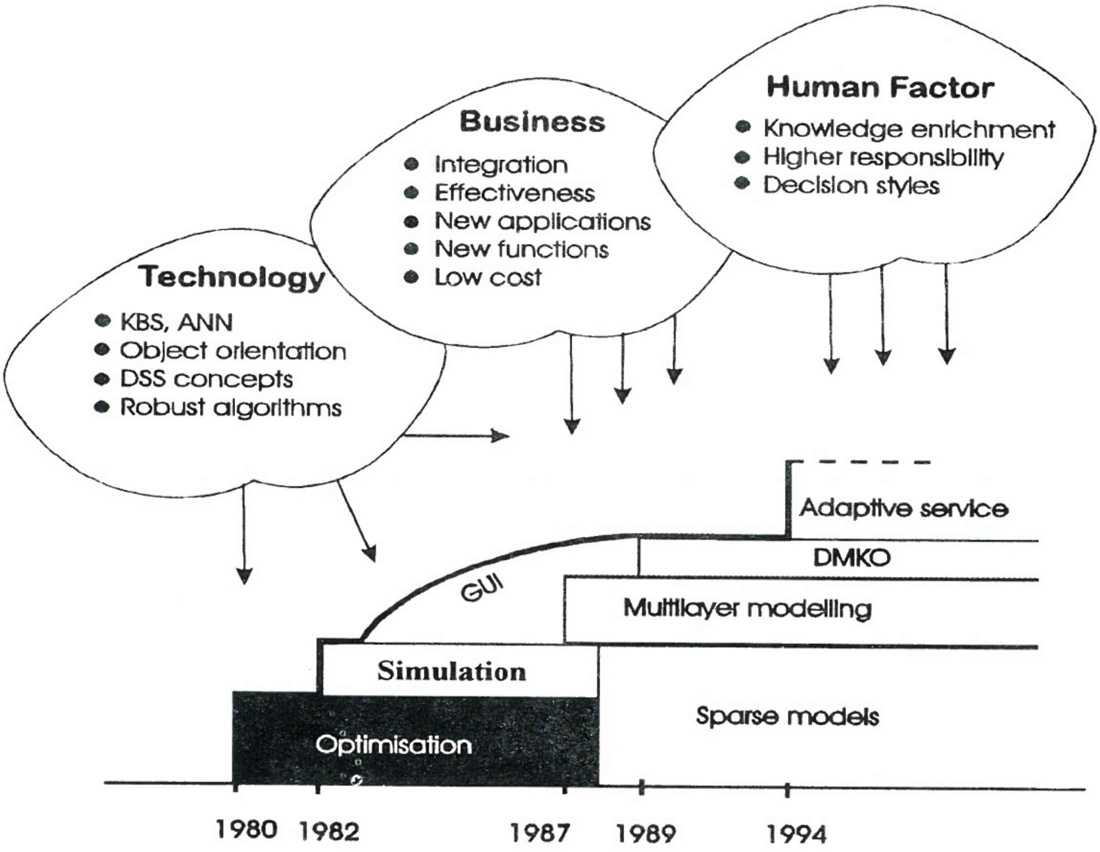

Figure 4. DISPATCHER Family Incremental Development (from [Filip,1995])

The system re-usability and adaptive character enabled various usages and allowed for a certain integration within the enterprise. Figure 4 in [Filip,1995], synthetically pictures the system incremental development with a view to adapting it to the greater influence of three groups of factors namely a) technologies, b) business perspective and c) human factor.

Other open problems the authors are concerned with are time aggregation, material blending problems and the development of an even more intelligent interface likely to be sensitive and automatically adaptive to user's degree of *domain* and *tool* knowledge.

REFERENCES

COAD, P. and YOURDON, E., **Object Oriented Design**, YOURDON PRESS, Prentice- Hall, Englewood Cliffs, N.J., USA, 1990.

DONCIULESCU, A.D. and FILIP, F.G., **Intelligent DSS in Production Control for Process Industry**, in N.Koussulas (Ed.) Proceedings ASI'94, 1994, pp. 181-187.

FILIP, F.G., **Towards More Humanized Real-time DSS**, Proc. BASYS'95, IEEE/ECLA/IFIP International Conference on Architectures and Design Methods for Balanced Automation Systems, Vitória, Brazil, 24-26 July, 1995.

FILIP, F.G., **DSS in Process Co-ordination Part I: Modelling Issues**, STUDIES AND RESEARCHES IN COMPUTERS AND INFORMATICS, Vol.1, No.3, 1990, pp.77-110.

FILIP, F.G., **Systems Analysis and Expert Systems Techniques for Operative Decision Making**, in Achim Sydow (Ed.) Computational Systems Analysis. Topics and Trends, ELSEVIER SCIENCE PUBLISHERS, Amsterdam, 1992, pp.285-305.

FILIP, F.G., ROBERTS, P.D. and ZHANG, J., **Combined Numerical / Knowledge - Based Hierarchical Control; Part III Process Scheduling and Co-ordination**, STUDIES IN INFORMATICS AND CONTROL,Vol. 1, No.4, 1992, pp. 267-274.

FILIP, F.G., DONCIULESCU, A.D., GASPAR, R., ORASANU, L. and MURATCEA, M., **Multilevel Optimization Algorithms in Computer Aided Production Control in the Process Industries,** COMPUTERS IN INDUSTRY, Vol. 6, No. 1, 1985, pp. 47-57.

GRANADO, J., **Business Technology and Information Technology**, Annex VI to Documentation for the WG on TBB, V2.0 Brussels, 1994.

KEEN, P.W.C., **Adaptive Design for DSS**, DATABASE, 1980, pp. 15-24.

NORMAN, D., **Emerging Models of the Enterprise**. TBP CEC Conf., Brussels, 1994, p.91.

RAULEFS, P., **The Virtual Factory**, in K. Brunnstein and E. Raumbold (Eds.) IFIP Trans. A52, 13th World Computer Congress'94, Vol. 2, ELSEVIER SCIENCE PUBLISHERS, Amsterdam, 1994, pp. 18-38.

WILLIAMS, TH. J., **Minutes**, 11th Workshop Meeting, IFAC/IFIP. Task Force on Architecture for Enterprise Integration, 1995.