Production Systems and Information Engineering
Volume 2 (2004), pp. 3-21

PRODUCTION PLANNING OF INDIVIDUAL MACHINE SYSTEMS: A RATE BASED APPROACH USING SIMILARITY

FERENC ERDÉLYI
Production Information Engineering Research Team (PIERT) of the Hungarian Academy of Sciences;
Department of Information Engineering, University of Miskolc
Hungary
erdelyi@ait.iit.uni-miskolc.hu

TIBOR TÓTH
Department of Information Engineering, University of Miskolc
Production Information Engineering Research Team (PIERT) of the Hungarian Academy of Sciences;
Hungary
toth@ait.iit.uni-miskolc.hu

[Received January 2004 and accepted March 2004]

Abstract. Production planning and scheduling is one of the most important technical activities of enterprises in the manufacturing and engineering industries. In this paper we define the concept of production rate at different hierarchy levels of production planning and scheduling. At the lowest level it appears as material removal rate (MRR). At the highest level it means the rate of activities of the current project. In the customer-oriented machine industry a project can be defined as a set of abstract engineering activities (e.g. mechanical design, part manufacturing, mechanical assembly, electrical and electronic design, electrical and electronic assembly, etc.) that must be completed by a specified due date. The main task of production planning is to distribute the necessary production load-fractions to the resources within the allowable time window in a quasi-optimal way. In project based production modeling the constrained optimum problems can be effectively solved by planning of the production rate as a function of time. In creating the planning models the similarity of projects can be advantageously utilized. The paper gives an overview of the theoretical background and summarizes the expectable benefits of the proposed approach. Some initial application experience obtained so far at a Hungarian factory will also be outlined.

Keywords: Aggregate production planning, project resource model, capacity model, constrained optimum problems, similarity, intensity/rate-based approach

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

1. INTRODUCTION

Production planning and scheduling is one of the most important activities for enterprises in the manufacturing and engineering industries. Production planning is carried out at several hierarchy levels in general. The task of *aggregate production planning* is to generate quantitative and scheduling data on production in the medium and long run (usually for three months and one year, respectively). The input data of production are: the orders based on market demands, specification of the products and technology processes, the internal and external (supply chain) capacities available, as well as stocks (*V.D. Hunt,* 1989). The output is the *aggregate production plan* and the *master schedule*.

In order to summarize the requirements for production planning we have to start from the goals of business policy of the firm in question. They can be as follows:

- ***Improved customer service.*** Nowadays this business goal is top priority. Keeping the market and attracting new customers is a precondition of realization of every other business goal. This goal can only be obtained by means of guaranteed quality of products, meeting deadlines and product specifications, as well as offering an advantageous price-level.
- ***Increasing revenue.*** This business goal, at the level of production planning and control, requires continuous improvement and control of the macro-parameters of the so-called *production triangle* (readiness for delivery, stocks level, capacity utilization: *D.Kiss*, 1988; *T.Tóth*, 1998; *D.Kiss* and *T.Tóth*, 1999).
- ***Lowering working capital.*** This goal can also be achieved by simultaneously and in a synchronized way improving the macro-parameters (production indices) of the production triangle mentioned previously. Meeting deadlines and minimizing the stocks level under reasonable constraints have a direct effect on working capital demand.
- ***Managing fixed assets.*** Utilization of the existing capacities invested in earlier is a fundamental condition to realize effective production capable of ensuring the profit expected. In case of well-proven products, successful accomplishment of the accepted external orders depends on, in most cases, the capacities available.
- ***Reducing operation costs.*** Under the conditions of the prices agreed and fixed in contracts the net profit can mainly be influenced by decreasing the operation costs and lead times, as well as by optimal utilization of the resources (machines, workers, materials).

It is easy to see that the concept of *competitive enterprise* can only be defined in a complex manner (*J.G. Monks,* 1987). The primary business goals can only be influenced through improving the secondary manager (or performance) indices. Effectiveness of production planning and control can only be ascertained after the results obtained in money, i.e. with a delay. Factual influence of the previously

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

made decisions related to scheduling of the production activities (i.e. concerning their quantitative and time-based distribution) can only be ensured by means of a smoothly operating activity-based controlling system.

2. AGGREGATE PRODUCTION PLANNING OF INDIVIDUAL MACHINES AND MACHINE SYSTEMS

The tasks of aggregate production planning are very different in mass production and in the one-of-a-kind production of complex individual machines and machine systems (e.g. production lines suitable for producing specialized products of large volume). In large series and mass production the most important viewpoint is to harmonize prediction of the market demand and utilization of the capacities available. In the case of production of the complex products mentioned above, production planning has to be subordinated to the interest of successful realization of the external orders obtained. Here the demand for flexibility of production is significantly greater than that of mass production and the deadlines are stricter. Production planning has to be dynamic and incremental. This means that aggregate production planning is controlled by not the start of planning periods but by the order-events appearing in changing dates. The new orders necessitate rearranging the work quantities previously allocated to production but this is also limited by the conditions determined by the works in progress.

In the case of the production of individual complex machines and machine systems the project-like approach becomes more advantageous. Project-like production planning is a typical production planning function of the machine manufacturers that make very complex products according to special orders. A complex machine of great value, that has been made to order and is to be assembled of numerous parts, requires a great variety of expertise. Therefore such a complex product is usually made as a special version of another similar machine made and sold in a previous period, i.e. the new machine to be made to order can be considered as a further developed and more or less modified version of a similar one previously made and sold in a successful way. The activities and processes of a project are based on experience of previous similar production activities and processes on the one hand, as well as on the unchanged and standardized engineering documentation and specific data of the new project including the new technical documents attached, on the other hand.

Aggregate production planning uses a high-level resource model of hierarchical structure. This model is different from firm to firm and is adapted to the usual practice (see Figure 1).

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

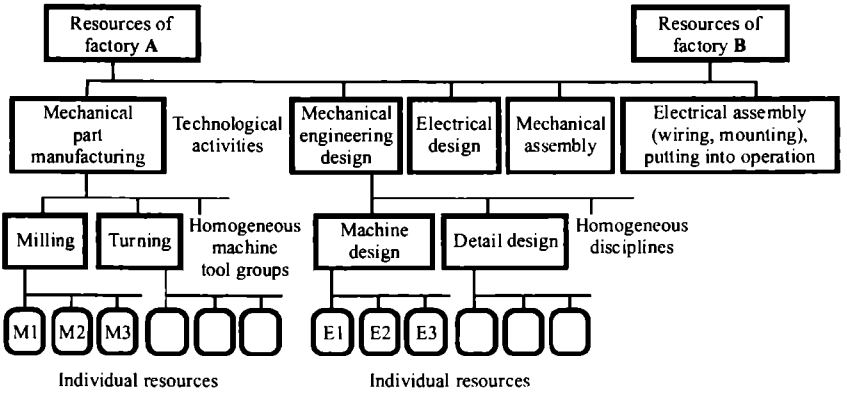


Figure 1: A characteristic resource model

A fundamental feature of the resource model is the available capacity of the given *resource class* depending on the *production calendar*. The resource model used by aggregate production planning is an abstract one and is connected with the high-level activities of production process. Every aggregate activity requires one or more resource(s) and this demand can be expressed in terms of generalized working hours required.

For example, let the list of aggregate production activities in a machine works be the following:

1. Mechanical engineering project work (engineering design)
2. Electrical project work (electrical design)
3. Part manufacturing
4. Component purchasing
5. Mechanical assembly (mounting)
6. Electrical assembly (wiring, mounting)
7. Putting into operation, testing and delivery.

To the activity type set $A=\{A_1, A_2, \cdots, A_p, \cdots, A_P\}$ a resource type set $R=\{R_1, R_2, \cdots, R_k, \cdots, R_K\}$ is allocated where $K \geq P$ The effective projecting models make it possible to utilize several resources by a given activity, too.

Available capacity is defined as the capability of resource class R_k for doing a certain work, available in the course of the given work-week (the unit of measurement is working hours/week) (*A. Kusiak,* 1984). In the aggregate planning models it is expedient to model the time by means of a series of discrete time intervals δt. In most cases the discrete time unit is one work-week. On the discrete time scale let t be the serial number of time interval, i.e.: $t=(1,2,\cdots,T)$. At the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

planning time horizon the so-called *relative time* is $\tau = t \cdot \delta t$ At the end of the time horizon used in modelling we have $\tau_s = T \cdot \delta t$. Considering this time horizon there is an internal resource capacity for every time unit according to the calendar: $c_k(t)$, $k = 1,2,\cdots,K$ The production scheduling model treats the available capacity, after it has been fixed, as a constraint. In mechanical engineering not only the internal capacities should be taken into account but the external capacities based on suppliers as well, in order to fill the external orders obtained. The external capacity $s_k(t)$ is more expensive in general and it is also constrained (*J.E. Buzacott,* et.al., 1993).

From the point of view of content, project work can be classified into three different types. They are as follows:

(1) ***Project work for tender***

This is the basic version of project work, which consists of the analysis of demand (or: interest) of the potential purchaser (or: customer), a feasibility study of the project and determination of the main data of the project. The deadline of the project previously accepted has to be determined on the basis of such a model, in which the activities and their work demand are only known at an estimated level.

(2) ***Detailed project work***

This is the principal version of project work including all the known phases of product design, technology process planning and production planning on the basis of the customer's order. The project must be included in the actual projects running in the same period. Scheduling of the project is to be carried out by taking into consideration the actual business goals and by fixing the constraints and the objective function.

(3) ***Redesign, replanning and rescheduling of projects***

This is a correcting and modifying version of project work. It is used when certain modification is needed because of an unexpected reason that has arisen in the course of parallel project execution. There can be many kinds of reasons, including:

- a change in the business processes,
- a change in the production policy,
- changes in the engineering specification,
- unexpected business events,
- unexpected events in the technology process,
- changes in the constraints and objective functions,
- changes in the uncertainty factors, etc.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

For project-like production planning, another key issue is what we consider to be the optimal production plan. As is known (see: *D.Kiss,* 1988, *T.Tóth,* 1998, *D.Kiss* and *T.Tóth,* 1999), in order to qualify as achieving the production goals three natural state variables (macro-parameters) are needed and they are also sufficient at the same time. These complex state variables can be considered the three nodes of an abstract triangle (the "Production Triangle" suggested by *D.Kiss,* 1988). They are as follows:

1. The *average utilization of resources*;
2. The *readiness for delivery*, i.e. the reciprocal value of the average lead time of the external orders;
3. The *average stock level* fixed in production.

These complex state variables, of course, are not independent of each other. Any of them can be improved to the detriment of the other two.

In the production planning of individual machine systems the alternative objective functions of a project scheduler suitable for optimisation appear as the special descriptions of the "Production Triangle" The objective functions are:

1. The weighted sum of the external capacities utilized;
2. The weighted sum of due-date tardiness of the projects;
3. The number of projects released at the same time.

In project-based production the successful realization of external orders is a very important and primary business goal that has to be supported by the utilization of external capacities as well. However, the maximal utilization of internal capacities is also expected. The most important characteristic of the *readiness for delivery* is to meet the due dates (terms of delivery) fixed in the contract. The deviation from the term of delivery either may be not allowed (hard constraint) or may be an objective to be minimized.
The task of the scheduler of project activities is to determine those production activities (both in quantity and in time) that meet all the constraints and minimize the objective function in the domain allowed. The first objective function of project work gives a good solution, typically, in the case of overloaded resources. If the works required by the actual order-book of the firm cannot load the resources in the planning period then the value of external capacity demand is equal to zero and there can be numerous scheduling solutions suitable for meeting the constraints. In many cases it is difficult to decide if improving the stocks level or improving the readiness for delivery should be the objective targeted at such a time. The conflict between the short term and long term goals makes the situation even more complicated. The philosophy of the schedulers used at present is, in general, that the constraints are the important ones; they have to be met by all means. There can also be several production plan solutions (schedules) meeting all the constraints. It is possible to select the most suitable of them on the basis of heuristic

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

considerations. Of course, an exact optimum is out of question here. The larger the number of permissible solutions, the more robust the optimum is, and the less sensitive it is to the changing circumstances.

3. THE PRODUCTION RATE AS A STATE VARIABLE

In the Department of Information Engineering at the University of Miskolc scientific investigations have been carried out for a long time related to the role of production rate type state variables at the different hierarchy levels of production management, from the well-known material removal rate (MRR) to the rates of the main production activities.

Production processes are typically integrating and cumulative processes, the output, i.e. the quantity of produced products, of which is continuously increasing in time. In the control of such processes the *process rate* (process intensity) is of great importance (*G.L.Ravignani,* 1977, *I. Detzky,* et al., 1989). If we consider production control as a closed control loop then the basic signal of control is the production rate. The rate of production processes can be measured in the measuring unit [working hours used/time unit] in the most general manner. At the level of operations the production rate depends on the *technological rate* that can be measured in measuring units [number of products/time unit] or [removed material volume/time unit]. In cutting technology processes where the finishing processes are of great importance, the measure of rate is [machined surface/time unit] and in case of chemical technology processes [processed mass (volume)/time unit].

For cutting the technological rate as a state variable in time can be defined as follows:

$$\int_0^t Q(\tau)d\tau = V(t), \tag{1}$$

or, in differential form:

$$Q(t) = dV(t)/dt\,, \tag{2}$$

where $Q(t)$ - the cutting rate changing in time;

$V(t)$ - the material volume removed until the time t.

Then

$$Q(t) = A(t)\cdot v_e(t), \tag{3}$$

where $A(t)$ - the momentary effective cross section of cutting

v_e - the feeding speed.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Eq. (3) can also be used in case of multiple-edged tools (see Figure 2).

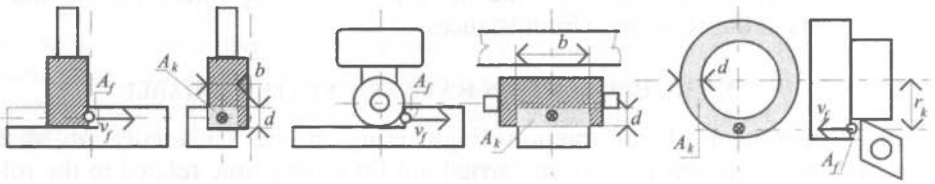


Figure 2: Interpretation of the momentary effective cross section in case of different machining methods.

In planning and production control the average rate $\overline{Q}$ is advantageously used for a given operation or operation element that makes it possible to estimate the primary time of cutting (the machining time) t_m

$$\overline{Q} = V / t_m . \tag{4}$$

Here V is the material volume removed in the given operation (or operation element: *T. Tóth,* 1997).

The rate of technological operations is the reciprocal value of the operation time; its measuring unit is [1/min]:

$$q_0 = \frac{1}{t_0} = \frac{1}{t_m + t_a}, \tag{5}$$

where q_0 - the rate of operation,

t_0 - the operation time,

t_a - the auxiliary time.

If in Eq. (5) $t_m >> t_a$ then t_0 can be approximated by t_m and the following relationship is valid:

$$q_0 = \frac{\overline{Q}}{V} . \tag{6}$$

Part manufacturing demands a consecutive series of operations in general; therefore the average rate of part manufacturing, referring to work pieces or series, is an aggregate production characteristic.

$$\overline{q}_p = \frac{n_p}{t_c} \text{ [pieces/min]}, \tag{7}$$

where n_p is the lot size and the cumulated time t_c can be calculated as follows:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$t_c = \sum t_{prep} + \sum t_0 + \sum t_w \tag{8}$$

Here t_{prep} is the preparation time and t_w is the time of the work piece spent in waiting. Summing has to be extended to all the operations of the series executed so far. The average rate of part manufacturing referring to work piece series plays a great role at shop floor level and in medium term scheduling where the equilibrium of demand rate and production rate is the condition of production stability.

The concept of production rate is the most abstract at high-level aggregate planning. At this level it is expedient to model the production rate by the rate of production activities. The rate of activities can change in the function of time not only from project to project but within any project as well.

Let the *i*-th activity of the actual running projects be A_i. Every activity has an earliest starting date and a latest completion due date (deadline). Let us denote these two dates with e_i and d_i, respectively. Both dates will be determined in the course of aggregate planning and they can be originated from the project deadline, as well as from the precedence of the activities. Any project means a defined product to be manufactured, the technology process planning of which gives that the project activity demands engagement $r_{i,k}$ [working hours] to the resource used by the activity, in a cumulated way. At preliminary planning for a bid this, of course, can only be based on engineering estimations, however after having carried out detailed process planning it can be calculated from the technology process plans in a well-established way. For a given activity one or more resource engagement(s) can also be allocated but this fact will have importance in the planning phase of the capacity-constrained production scheduling only.

We can give an implicit definition for *activity rate* in case of aggregate planning:

$$\sum_{t=e_i}^{d_i} q_{i,k}(t) \cdot \delta t = r_{i,k}, \quad i = 1,2,...,n \tag{9}$$

Hence, the activity rate $q_{i,k}(t)$, changing in time discretely, is the activity concerning the time unit demanded by the *i*-th project, which loads the *k*-th resource. We name the "stepped" function $q_{i,k}(t)$ the *profile of activity* (see: Figure 3). For the profile numerous constraints can be defined which must be taken into consideration in the course of production planning and scheduling.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

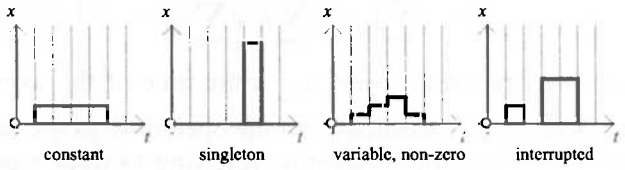


Figure 3. Characteristic activity profiles

The rate of a project activity can be constrained from below and from above:

$$q_{im} \le q_i(t) < q_{iM} \tag{10}$$

The upper constraint depends on the type of activity, which expresses that the rate of the given activity cannot exceed the maximum denoted by q_{iM} even if there were free capacity available for this purpose. (It is not possible to design or to assemble a machine with optionally great rate; there should be an adequate human expertise and free capacities must be also available.) The lower constraint can express the fact that if we have already started with a certain activity then a minimum expenditure is needed for it in every time interval. If $q_{im} = 0$ then the activity in question can be interrupted; otherwise it cannot be done. A correct modelling of the rate constraints is of fundamental importance for scheduling of projects because the model of the scheduler is obviously sensitive to the right boundaries.

Let the "time window" of the i-th activity be the following:

$$\Delta t_i = d_i - e_i \tag{11}$$

The minimum number of weeks during which the activity can be executed is as follows:

$$1 < \frac{r_i}{q_{iM}} < \Delta t_i\,, \tag{12}$$

from which we obtain:

$$\frac{r_i}{\Delta t_i} \le q_{iM} \tag{13}$$

The inequality (13) means a lower constraint for the maximum of the activity rate, i.e. if (13) is not satisfied the project deadline cannot be met. Another constraint for q_{iM} can come about from technological features of the competent resource of the activity. For each activity a minimum time interval for completion of the activity can be determined according to experience. This can depend both on the project type and the utilized resources at the same time and it can be given for the project

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

planner in a two-dimensional table (p_{ik}). Obviously, the relation $p_{ik} \cdot \delta t \le \Delta t_i$ has to be performed; otherwise the project deadline cannot be met.

On the basis of the aforementioned considerations, the constraint

$$\frac{r_i}{p_{ik} \cdot \delta t} \ge q_{iM} \tag{14}$$

is also to be satisfied. Therefore q_{iM} has to be kept within bounds:

$$\frac{r_i}{\Delta t_i} \le q_{iM} \le \frac{r_i}{p_{ik} \cdot \delta t}. \tag{15}$$

The relative value of the maximum rate allowed is as follows:

$$a_i = q_{iM} / r_i \tag{16}$$

Modelling of the loading profile of project activities is a problem treatable in a more complex way.

The profile can be modelled with a graph or a conventional *Gantt*-diagram in a rough way only, because these graphical tools only concentrate on the time conditions and partial deadlines. As regards the resource demand of the project, only a constant or periodically constant rate can be modelled.

Demonstration of the loading profiles with a set of time-functions is better but there exists the danger of not-easy-to-survey (see Figure 4).

For the rate-based modelling of activities the difficulties can be summarized in the following way:

The definition of activities is subjective; it depends on the opinion and experience of the experts (designers, planners).

- The characteristic profile of activities can be very different. Part manufacturing can be interrupted at a highly aggregate level if the time unit is a week. This means that the profile can be of zero rate value even if the execution phase has already started. Electrical design, however, cannot usually be interrupted. It is worth completing the smaller design works without interruption if we have already started to deal with them.
- Formulation of the requirements for the profile of activities can be neither complicated nor very rigid for a production planning expert. Every complication can disturb the planning manager. A more serious problem is that there are mathematical consequences of the constraints related to the profile form both for the model and for the solver software as well.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

- Any experimental information can be useful but it can also be a useless constraint conserving a bad practice.

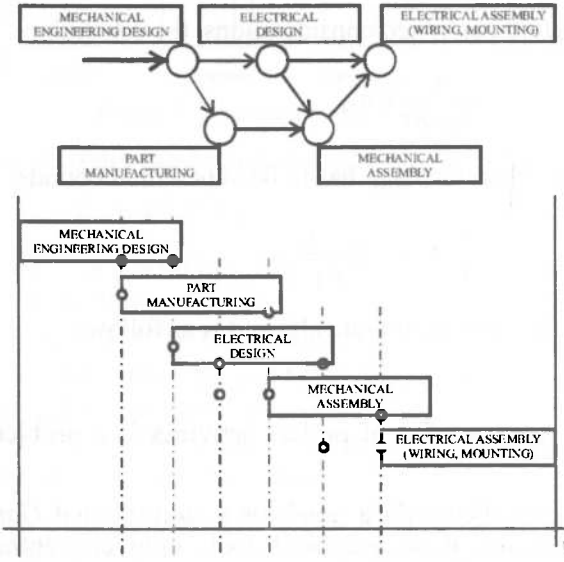


Figure 4: Demonstration of the loading profiles in graphical ways

There are precedence constraints between the activities. On the one hand, they originate from the technology process itself; on the other hand, they can be deduced from business goals, considerations between the projects. Precedence, for instance, can be described by a directed acyclic graph $D=(N,A)$. The simple or special precedence $A_i \rightarrow A_j$ means that activity A_j can only start if A_i has ended. It can also be interpreted as a more general precedence $A_i \overset{p}{\rightarrow} A_j$, which means the activity A_j can only start if A_i was completed to p %. In aggregate production planning the latter is typical. For modelling precedence a binary variable can be allocated to every activity fraction, $x_i(j) \Rightarrow z_i(j)$, which shows whether the rate is allowed in the given time interval. There is a "stepped" function $z_i(t)$ for activity A_i, which separates the interval $\Delta t_i = d_i - e_i$ into two sections. One of the sections is allowed for the activity, the other is not (*T. Kiss,* 2003).

4. RATE BASED MODEL FOR AGGREGATE PRODUCTION PLANNING

The first basic task of the aggregate production planning is to choose those production goals that are to be achieved in the planning period. Planning is carried

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

out on the basis of market predictions, the orders of customers and the capacities available, taking into consideration the specifications and quantitative data of the products to be manufactured. The second problem of production planning is to schedule the chosen high-level production activities in time and in a quantitative manner. In project-like production planning this can be done by choosing those specific production loads (i.e. discrete production rates) that appear on the resources in the chosen planning horizon. These tasks can also be solved in several ways and the task of computerized production planning applications is to support this solving process. Aggregate production planning models are conducive to constrained discrete optimum problems in general, the solving of which is supported by the results of Operations Research.

Considering the fact that there are effective computer solvers suitable for solving linear programming problems, it is worth investigating those models of the aggregate production planning which can be solved by these solvers. The problem has been investigated by a research consortium consisting of five Hungarian partners for the last two years, they are as follows: the Computer and Automation Research Institute of the Hungarian Academy of Sciences (CARI-HAS), Budapest University of Technology and Economics, the University of Miskolc and two firms from the competitive sphere. Several models of the joint research work show promising results (*F. Erdélyi,* et al. 2002).

Let us introduce the relative production rate changing on a discrete time scale:

$$x_i(t) = q_i(i)/r_i\,, \tag{17}$$

which means the loading fraction of the activity in the t-th time interval. It is obvious that

$$\sum_{t=e_i}^{d_i} x_i(t) = 1\,. \tag{18}$$

The relative production rate of project level can have a value between the limits 0 and 1. For the sake of simplifying the model let us assume that every activity can be interrupted therefore any value of $x_i(t)$ can also be equal to zero.

Let the goal of business policy be the maximal utilization of internal resources. In this case the rate of utilization of external capacities is to be minimized so that the objective function of the project scheduler is to minimize the utilization of external capacities. Hence, the objective function is:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$\sum_k \left[w_k \sum_t y_k(t) \right] \Rightarrow min, \tag{19}$$

where

$$y_k(t) = \max\left[0, \left(\sum_i q_{i,k}(t)\right) - c_k(t)\right] \tag{20}$$

In Eq. (20) $y_k(t)$ is the rate of external capacity used in the t-th time interval. In Eq. (19) w_i is the weighting factor expressing the properties of the resource in question.

The task of the project-based production scheduler is to determine those relative production rate fractions $x_i(t)$ and external demands $y_k(t)$ which meet all the constraints related to times, capacities and sequences, as well as to minimize the objective functions (*N. Duffie,* 2002).

The constraints are as follows:

$$x_i(t) = 0 \text{ if } 1 \le t \le e_i \text{ and } d_i \le t \le T \tag{21}$$

Constraint (21) means that the activity has to be completed in the given time window.

$$\sum_{t=e_i}^{d_i} x_i(t) = 1 \tag{22}$$

Constraint (22) expresses that every activity has to be carried out entirely (this is the same as equation (18)).

$$\sum_{i,k} r_{i,k} \cdot x_{i,k}(t) \le c_k(t) + y_k(t) \quad 1 \le t \le T \tag{23}$$

Constraint (23) means that all the demands are covered by the internal and external capacities.

$$y_k(t) \le b_k(t). \tag{24}$$

Constraint (24) shows that the external capacity is also limited.

$$x_i(t) \le a_i \cdot z_i(t). \tag{25}$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Constraint (25) expresses that the rate of activity cannot be greater than the allowed and it can only be different from zero in that interval where the precedence control condition allows.

If there is no solution of the planning task with the given data then it is the task of the production engineer to intervene interactively in the computer aided planning process. It can be expedient to slacken certain constraint(s) or to define a new production planning task by changing the demands of the project.

The aforementioned strategy of project work results in a solution most typically in resource-overloaded cases. If the task is not resource-overloaded then the value of the objective function is obviously zero and there can be numerous solutions for meeting the constraints. At that time the task of the project scheduler is to suggest those solutions from the possible and allowed solutions considering which profile of rate changing is the most suitable for meeting the requirements of the production goal.

5. Principles of Similarity Based Production Planning

Important tools of the aggregate production planning are those estimating procedures that estimate the probable structure of activities and the utilization of resources on the basis of the similarity of the products. Under such circumstances the modular structure of these products, the principles of Group Technology (GT) and the similarity-based estimations can have an important role.

Machine manufacturers meet the task of aggregate production planning in the period of tender when obtaining the order is an outstanding business goal. If the production plans of the product meeting the requirements of customer are not available then inserting the project into the running tasks requires careful aggregate planning that includes planning alternatives of "What would happen then if..." type as well. Here the most important things are a well-established delivery deadline and a reliable estimation of the probable capacity overloading (*T. Tóth,* 1999).

In the course of the realization of a project two different hierarchies have to be taken into consideration:

- the structural hierarchy of the product (complex machine) constituting the base of the project in question;
- the technological hierarchy realized in the manufacturing process.

Structural hierarchy reflects the physical reality of the product, as well as subordination of the main machine units adequate to the major functions. We assume that a product can be dissected into four hierarchy levels at the very most:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

(1) the complex (complete) machine
(2) a machine unit
(3) an assembly unit
(4) a part group.

We hold natural that those projects can only be compared to each other that belong to the same structural hierarchy level.

In the hierarchy of the production process we allow two levels, namely

(1) the level of aggregate activities of a complete project;
(2) the level of operations of the production activities.

The first step of similarity-based production planning is to allocate the project to be planned to a product hierarchy level. After this, at the given hierarchy level, we select the similarity projects from the projects previously completed. This is an algorithm consisting of several steps. We make a list including the operations executed in the projects, the utilization of capacities, and the times for planning, manufacturing and assembly. The operation set obtained in this way can also be supplemented with several specific operations if needed for realization of the new project and if they have not appeared in any similar project so far. So we obtain a possible set of operations. We allocate the operation times occurring already in the completed projects to these operation sets in a primary table. The similarity based selection, after all, will be executed by means of a secondary table that qualifies the similar projects on the basis of the occurrence of operations and the operation times within defined tolerances. On the basis of activities of the projects selected in this way we can get a fairly good estimation for the production time requirements of the project activities planned.

6. CONCLUSIONS

Aggregate production planning is an important and difficult task of firms making individual machines and complex equipment. The concept of project-based production planning and scheduling makes it possible to treat the production activities and the engagement of capacities together. A production scheduling model can be based on the rate of project activities. The model of project activities is significantly different from the scheduling model of shop floor control. The high-level activities of the projects can be interrupted and can be planned at a changing rate. The activities can also use several different resources. The profile of rates in time can be influenced by additional constraints.

We have applied the rate-based model of aggregate production planning in R&D works carried out within the framework of a consortium. The model proved to be

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

successful in case of different products and production profiles as well. The experimental computerized applications are being tested at present. The advantages offered by the rate-based aggregate production scheduler are as follows:

- The number of occasions when the deadline is over-run decreases.
- The lead times of projects decreases.
- Utilization of capacities increases and will be more balanced.
- The use of overtime and external capacities decreases.
- The bottle-necks can be recognized and can be treated in a better way than earlier.
- The number of works in process decreases.

As a result of the advantageous effects listed above, the set of external orders can also change advantageously.

In addition, an important benefit is increasing the co-ordination of engineering functions and improving the integration of the chief engineer department and shop floor levels. Alternative solutions of modelling of the production processes increase efficiency of management decisions. On the basis of experience a reengineering process of greater scale can be realized for improving the working process of the production planning organization.

A disadvantage in application of the method is that the quantity of data required is greater than that of the standard approaches. Rate/intensity has a differential character and therefore it is fairly sensitive to errors of calculation and parameterization. The rate constraints allowable must be estimated on the basis of experience. Present engineering and management practice prefers the operations of constant rate as compared to the operations of variable rate. In the case of constant intensity the time of operation to be expected can be estimated in an easy way. During project planning the model allows zero rates, i.e. interrupting the activity. This fact can give trouble to the experts.

ACKNOWLEDGEMENTS

The research was partially supported by the Hungarian Academy of Sciences (HAS) within the framework of Production Information Engineering Research Team (PIERT) established at the Department of Information Engineering of the University of Miskolc (Grant No. MTA-TKI 06108). The results are also connected with the project entitled "Digital Factories, Production Networks" (National Research and Development Program founded by the Ministry of Education of Hungary; Grant No.: 2/040/2001, project leader: *László Monostori* DSc). The authors would like to express their thanks for the financial support.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

REFERENCES

[1] ASKIN, G. A., STANDRIDGE, C. R. (1993), *Modeling and Analysis of Manufacturing Systems*. John Wiley and Sons Inc., New York.

[2] BUZACOTT, J. E., SHANTHIKUMAR, J. G. (1993), *Stochastic Models of Manufacturing Systems*. Prentice Hall Inc., New Jersey.

[3] DETZKY, I. FRIDRIK, L., TÓTH, T. (1989), *On a New Approach to Computerized Optimization of Cutting Conditionns*. Proc. of the 2nd World Basque Congress. Bilbao, V.1. pp. 129-141.

[4] DUFFIE, N. FALU, I. (2002), *Control Theoretic Analysis of a Closed Loop PPC System*. Annals of the CIRP V 51/1 2002. pp. 379-382.

[5] ERDÉLYI, F. TÓTH, T., SOMLÓ, J., KOVÁCS, A., KÁDÁR, B., MÁRKUS, A. VÁNCZA, J. (2002), *Production management: taking up the challenge of integration*. 3rd Conference on Mechanical Engineering. Budapest, 2002. pp. 705-709.

[6] GOLDRATT, E. M. (1994), *Theory of Constraints*. North River Press, New York.

[7] HUNT, V. D. (1989), *Computer Integrated Manufacturing Handbook*. Chapman and Hall Ltd. New York.

[8] KISS, D. (1988), *An Information Model for the Production Planning and Control of Enterprises*. Symposium of "Organization and Computational Technology Application", Nyiregyháza (in Hungarian).

[9] KISS, D. AND TÓTH, T. (1999), *The methods of theoretical approach in Production Planning and Control*. In: Information Systems for Enterprise Management in Hungary (Hetyei, J., Ed.), ComputerBooks, Budapest, pp. 59-94. (in Hungarian).

[10] KISS, T. (2003), *A Branch and Cut Approach for scheduling projects with variable intensity activities*. 6th Workshop on Models and Algorithms for Planning and Scheduling Problems. Aussois, France, 2003. pp. 160-172.

[11] KRAJEWSKI J., RITZMAN B. (1996) *Operation Management*. (Strategy and analysis) Addison-Wesley Publishing Co.

[12] KUSIAK, A., DORF, R. C. (1994), *Handbook of Design, Manufacturing and Automation*. John Wiley & Sons Inc., New York.

[13] MONKS, J. G. (1987), *Operations Management: Theory and Problems*. McGraw Hill Book Company. New York.

[14] PERKINS, J. R., KUMAR, P. R. (1989), *Stable, Distributed Real-Time Scheduling of Flexible Manufacturing Systems*. IEEE Trans.on Aut. Cont. V. 34, N.2, pp. 139-148.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[15] RAVIGNANI, G. L., TIPNIS, V. A., FRIEDMAN, M. Y. (1977), *Cutting Rate Tool Life Function (R-T-F)*. General Theory and Application. Annals of the CIRP V. 25/1. pp. 295-301.

[16] SOMLÓ, J. (2001), *Hybrid Dynamical Approach makes FMS Scheduling more effective*. Periodica Polytechnica. V. 45. No. 2. 2001. Budapest. pp. 175-200.

[17] STARBEK, M., GRUM, J. (2000), *Operation lead time control*. Robotics and Computer Integrated Manufacturing. 2000. N. 16. pp. 443-450.

[18] TÓTH, T. (1998), *Design and Planning Principles, Models and Methods in Computer Integrated Manufacturing*. University of Miskolc Press (in Hungarian).

[19] TÓTH, T. (1999), *New Principles and Methods in the Computerized Integration of Process Planning and Production Control*. Publications of the University of Miskolc. Series C, Mechanical Engineering. Vol.49. pp.173-187.

[20] TÓTH, T., ERDÉLYI, F. (1997), *The Role of Otimization and Robustness in Planning and Control of Discrete Manufacturing Processes*. Proc. of the 2nd World Congress on Intelligent Manufacturing Processes and Systems. Springer Verlag, Budapest, 1997. pp. 205-210.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 23-36

PROJECT-ORIENTED APPROACH TO PRODUCTION PLANNING AND SCHEDULING IN MAKE-TO-ORDER MANUFACTURING

PÉTER EGRI, ANDRÁS KOVÁCS, ANDRÁS MÁRKUS, JÓZSEF VÁNCZA
Computer and Automation Research Institute
Hungarian Academy of Sciences
H-1111 Budapest, Kende u 13-17
HUNGARY
{egri,akovacs,markus,vancza}@sztaki.hu

[Received December 2004 and accepted January 2005]

Abstract. In this paper, we present a unified framework for production planning and scheduling in make-to-order manufacturing. Both planning and scheduling problems are captured as resource-constrained project scheduling problems. The actual models and solution techniques are different at the two levels. Hence, we also suggest an aggregation method that provides the connection between planning and scheduling. The viability of the approach is demonstrated by some experimental results on large-scale industrial problem instances.

Keywords: Production planning, scheduling, aggregation

1. INTRODUCTION

Production planning and *scheduling* (PPS) match future production load and capacities by determining the flow of materials and the use of resources, over various horizons and on different levels of detail. Albeit planning and scheduling problems have their own, specific timescale, resource and activity model granularity as well as optimization criteria, the two levels of PPS are strongly interdependent. On the one hand, planning guarantees on the long term the observance of high level temporal and resource capacity constraints and thus sets the goals as well as the resource and temporal constraints for scheduling. On the other hand, scheduling is responsible for unfolding a production plan into executable schedules; i.e., to detailed resource assignments and operation sequences. No scheduling strategy can improve much on an inadequate plan, whereas a bad scheduling strategy that wastes resources may inhibit the fulfillment of a good plan. All this makes PPS extremely complex and hard to solve. At the same time, PPS calls for efficient decision support methods and intuitive, flexible models with fast, reliable solution techniques that scale-up well even to large problem instances. Hence, even if production planning and scheduling problems

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

are solved in a superior-inferior hierarchy, they have to be treated in an integrated manner.

In this paper we review the main goals of PPS and introduce a model of an integrated planner and scheduler system. After having proposed solution principles, we describe an implemented prototype system and the lessons of its experimental use.

2. PROBLEM STATEMENT AND OBJECTIVES

Production planning (PP) is responsible for making the aggregate plan of using production resources (workforce, equipment) and material to meet customer orders. Typically, the plan covers a wide time horizon of several months. Besides giving the date of completing each customer order, production planning determines the capacity and material requirements of production over time. For instance, when materials should arrive, which activities should be outsourced to subcontractors, when to increase or decrease the workforce level. Due to the strong interrelations among such decisions, these questions must be addressed simultaneously. As a consequence, the two strongly coupled, but traditionally separated function of the production planning Material Requirements/Manufacturing Resource Planning (MRP, MRP II) and Capacity Requirements Planning (CRP) – must be handled in an integrated way.

According to the traditional approach, production planning in make to order production systems is still based on *lead time estimations* and on MRP logic [14]. In this approach, customer orders are segmented with milestones, and the time needed to reach the next milestone is estimated by production lead times. Estimates are based on *past experience* rather than on the actual production load. The MRP system determines the timing of these milestones backward from the due dates of the customer orders. At this stage of the planning process the *actual production capacities* are considered only implicitly, i.e., through the lead times, which again are based on historic data. In the subsequent stage, when the timing of production activities has already been set, production capacities are allocated to the specific customer orders. If in a certain period of time, the in-house capacity is not sufficient to meet demands, decisions are made as whether to extend the capacity of the scarce resource or to involve subcontractors.

Scheduling is responsible for making detailed, executable schedules that achieve the goals set by production plans. Hence, scheduling has to assign finite capacity resources to production operations as well as to determine their order of execution. So as to guarantee that all shop orders can be executed in time and that load on resources never exceeds available capacity, scheduling has to unify the resource and temporal aspects of production at the most detailed level of aggregation. Beyond satisfying various temporal and resource capacity constraints, the solution

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

should approach optimality with respect to some optimization criterion. Close-to-optimal solution of scheduling problems requires expressive and flexible models and efficient, customized solution methods.

In our opinion, the main requirements towards the models and the solution methods of an integrated PPS system are as follows:

- Representation methods at both levels should be able to capture relevant temporal as well as resource capacity constraints of production.
- The results should be optimal or close-to-optimal according to various objectives, and robust to cope with unexpected disorders.
- Production plans should be unfoldable into executable schedules. Hence, planning must also handle precedence relations that ensue from complex product structures (e.g., assemblies) and technological routings.
- However, resource assignment problems with finite capacities and precedence constraints are in general extremely hard to solve. Planning must apply aggregation so that typical instances of planning problems could be solved in a tractable way.
- The solution methods applied at both levels have to be efficient enough to support interactive decision making.
- Both planning and scheduling should use the same master data readily available in *de facto* standard production information systems: product and technology related data (e.g., bills of materials (BOMs), routings), resource calendars, and order data.
- The actual status of production and open orders should be handled on both levels.

3. THE MODEL OF THE INTEGRATED PLANNER AND SCHEDULER

In what follows we propose an integrated PPS system that was designed to meet the above target requirements. The overall framework of the system and its connections to other main modules is presented on Fig. 1 below. Note that the proposed PPS system provides a bridge between *de facto* standard Enterprise Resource Planning (ERP) and Manufacturing Execution (MES) systems. In the overall framework the role of simulation is to validate production schedules and test their sensitivity towards factors that are included neither in the planner nor in the scheduler model.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

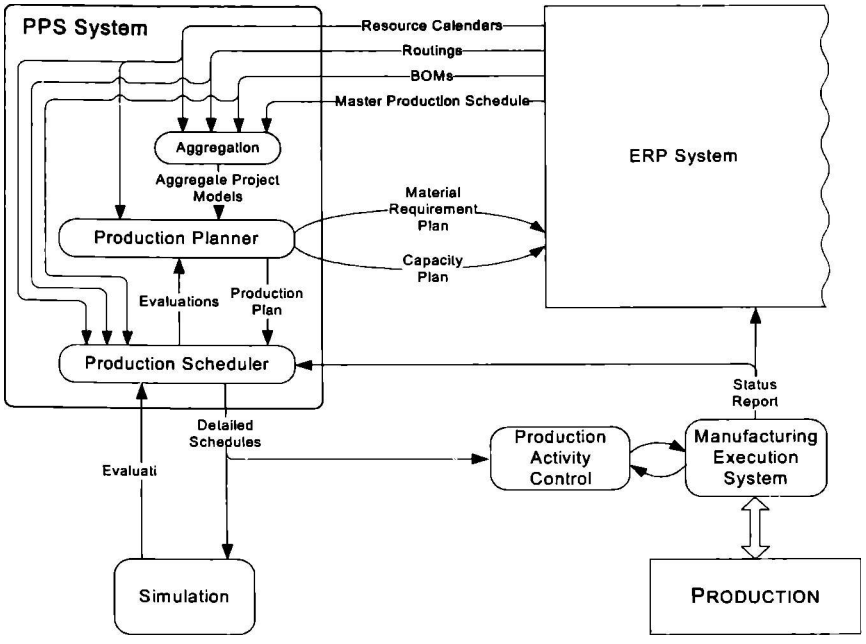


Figure 1: Structure of the PPS framework

3.1. Production planner

The challenge of production planning is the timing of the activities in medium term, over a typically 3-6 months long time horizon with a time unit of one week. The generated plans must comply with the project deadlines, obtain effective utilization of the resources with finite capacity, keep stock levels low, and on the whole, minimize the cost of the production. In our production planner the MRP logic and the production lead times are replaced by a *resource-constrained project scheduling* model. The timing of the competing customer orders is determined with taking into consideration other orders and the production capacity available. In particular, excessive use of certain resources in a time period is possible only if there is no way to avoid this but violating some customer order deadlines. All decisions are made with regard to the actual set of orders and production load. At the same time, important questions – such as the date when certain material should be available, or when additional resources are needed – get answered too.

In the planning problem, *projects* consist of various *activities* needed to complete an order. Usually, some ordering of the activities is to be followed, but many of them may overlap in time, especially in case of large, complex projects. Each activity may call for the use of a number of different resources. The *resources* are typically either machine or human resources (or both, in a coupled way) that will

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

be shared by the activities of different projects. The resources may be distributed, geographically dispersed and may even belong to different organizations.

Each product order is considered a *project*. A project has a *time window* set by the negotiated earliest starting time and deadline. Activities of the same project are linked by *precedence constraints*. An activity may require the execution of a given amount of work on one or more resources. However, the *intensity* of executing an activity may vary over time; the activity can even be pre-empted.

Activities are *aggregates*: they represent groups of manufacturing, assembly, etc. *operations*, some of which are executed simultaneously, some sequentially, and others independently of each other. This leads to a model in which not the durations but only the work amounts of activities are fixed *a priori*. Activities are defined in the course of an aggregation process (see Sect. 3.3) that uses product, production technology and resource availability information. Summing up, the planner works with the following input data (see also Fig. 1):

- specification of customer orders as given in the master production schedule;
- Bill of Materials (BOMs) of products;
- routings (sequence, processing time and resource requirements) of operations; and
- detailed calendar of available resource (machine, workforce) capacities.

The production planner produces the following outputs:

- medium term production plan, which assigns operations to weeks of the planning horizon;
- medium term capacity plan, which specifies the resource requirements of each week on the planning horizon; and
- medium term material requirement plan, which specifies for each week the requirements for raw materials and other components.

3.2. The scheduler

The ultimate goal of scheduling is to unfold the medium term production plan into an *executable* detailed schedule. The scheduler has to determine the order of the operations and the resource allocations with respect to the technological, temporal and capacity constraints. Our short-term scheduler performs finite capacity scheduling with respect to detailed technological and capacity constraints. The scheduling horizon is as long as the time unit of the planner (i.e., one week), while the scheduling time unit is 0.1 hour.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The set of *operations* to be scheduled are determined by disaggregating the activities that fall into a given time unit in the medium-term production plan. If an activity covers several weeks, then its operations are distributed in this period proportional to the activity's intensities. Typically, schedules are generated for the next few weeks only.

Most operations are non-preemptive but breakable, i.e., the workpieces cannot be unmounted from and re-mounted to the machines only after completing the operations. However, the on-going operations can be interrupted, e.g., during the weekends, and continued later on without any extra cost. We model also non-breakable operations (like heat treatment) that must not be broken due to technological reasons.

There are both individual (e.g., machine tools) and group *resources* (homogeneous machine groups, assembly stations, various pools of qualified workforce). Resource availability – that may vary shift-by-shift – is given by the detailed resource calendar. Resource and time requirements, as well as the sequence of operations are described in the *routings*. In the routing, each operation requires a given combination of resources. E.g., a turning operation might require a turning centre and a machinist during the entire length of its processing. In our scheduling model, operations have also specific *processing*, *setup,* and *transportation* times. We assume that transportation and setup are performed before the operation, but while the first needs the workpiece only, setup requires solely the resources of the operation.

The optimization *objective* of the scheduler is to minimize the maximal tardiness with respect to the due dates set by the production plan.

We took the constraint-based approach [1] to model the above scheduling problem. In the constraint model, *variables* are the start times of the operations. The variables are linked by several types of *constraints*. Temporal constraints are the *precedences* between the operations (as given e.g., in the routing, or implied by the BOM of complex products), the *durations* and the *time windows* (earliest start, latest finish times) of the operations. Performing setup and transportation may call for further time constraints, depending on the actual situation. *Resource constraints* prescribe that the resource requirements of the particular operations should be satisfied by limited resource capacities. Therefore, the solution of this problem is an assignment of start times to operations such that all temporal and resource capacity constraints are observed.

Summing up, the scheduler works with medium term production plans, detailed resource calendars, as well as BOMs and routings (see also Fig. 1). In return, it generates detailed predictive production schedules that satisfy all the technological and resource constraints, and approach optimality with respect to the actual optimization criteria.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

3.3. Aggregation: the connection between planning and scheduling

Although planning and scheduling models are built by using the same source of master data, the models are different at the two levels. On the one hand the size of the problem, on the other hand the uncertainty of the information related to future events suggest that the production planner should work with an *aggregate model* that covers only the most important temporal and resource constraints of the problem. However, since the production plan must be executable also on the job-shop level, aggregation – the creation of the medium term problem – is a very subtle task [2].

Traditionally aggregation involves the grouping of the operations which belong to the same project and require the same resource into an aggregate activity. This simple approach can be easily understood by human experts, but it can result in very complicated temporal relationships among the activities of an order [5]. These relationships can be expressed by generalized precedence constraints, intensity-curves, overlapping and similar conditions, but they cannot be generated automatically, since the enterprise information systems usually do not contain the necessary data. Consequently, this modeling policy requires the involvement of human experts. Moreover, this approach cannot guarantee executable plans.

In [13] we have proposed a novel method for constructing aggregate models for production planning departing from the detailed technological routings, BOMs and resource calendars. We note that in case of make-to-order manufacturing – when the ordered items are chosen from a catalog all this information is already available at planning time.

Orders are considered to be independent from each other. An order is modeled by a so called *project tree* – a rooted tree whose vertices with several children denote assembly operations, while those with a single child represent either machining operations or joining a purchased part to the workpiece. The execution of the project over time advances from the leaves towards the root that stands for the finishing operation of the final product. Edges represent strict precedence relations, i.e., the sons of an operation must all be completed before the operation itself could be started.

During aggregation, connected vertices of the project tree are contracted into components that define the activities of the planning model. This *partitioning* of the project tree is called the *aggregate model* of the project. If two operations of the project tree that are connected by a precedence constraint are inserted into the same activity, then this constraint is omitted from the aggregate model. Otherwise, a precedence constraint is posted between the two aggregate activities. Note that the precedence graph of the activities will also form a tree. The resource requirements of an activity are the sums of the processing and setup times of the contained

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

operations per each resource required. Fig. 2 shows two alternative aggregations of the same project tree.

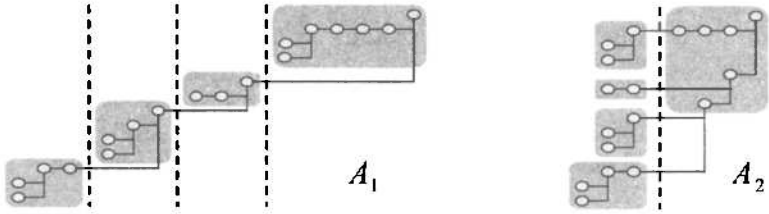


Figure 2: Alternative aggregations of the same project tree into activities

Aggregation has various effects both on planning and scheduling.

- Merging operations of the project tree into larger activities decreases the computational complexity of the planning problem.
- On contrary, too large activities can hardly be unfolded into feasible schedules. Therefore, it is reasonable to set a limit to the size of the aggregate activities. We have proven that the best compromise is setting the size limit of activities to the length of the aggregate time unit (one week, in our case). If an operation with a longer processing time hurts this condition, then this operation constitutes a single activity.
- Though the planning problem is usually considered as a relaxation of the detailed scheduling problem, some extra constraints may be introduced during aggregation. In any case, a precedence constraint in the aggregate model states that the connected activities have to be executed in the given order, *in distinct time units*. Hence, a precedence constraint implies a time unit change between finishing the preceded and starting the preceding activity. Therefore, the lead time of a project using a given activity model cannot be less than the its height. Consider the alternative activity models of the same project at Fig. 2: the two models have different cardinalities (4 vs. 5) and different depths (4 vs. 2). Clearly, *A2* provides a more appropriate aggregation of the same project tree, although it has more activities than *A1*.

Based on the above analysis, the criteria for aggregation are as follows:

- The total resource demand of an activity should not exceed the internal capacity limit per each time unit.
- The height of the aggregate model should be minimal (so that it can contain as many parallel branches as possible).
- The number of activities should be as small as possible (to reduce the problem size).

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Since the last two requirements are in conflict, an acceptable trade-off must be found between them.

4. SOLUTION TECHNIQUES AND ALGORITHMS

4.1. Solving the planning problem

When solving the planning problem, our primary objective is the *minimal extra capacity* usage. In this way, the planner attempts to keep the works allotted for a medium-term horizon within the factory. There is also a secondary objective: in order to minimize inventory costs, the level of *work-in-process* (WIP) should be minimal. We note that classical optimization criteria, like *project duration*, *maximum tardiness* or *weighted tardiness* fit also in the proposed framework.

To formalize the planning problem we have used a *resource-constrained project scheduling* model [12], whose detailed analysis has shown, that generally it is NP-hard. However, the analysis resulted also in a linear program re-formulation with cutting planes. The solution method uses them in a custom-tailored efficient *branch-and-cut search* that finds optimal solution [8]. The proposed algorithm is *any-time*: it generates a series of solutions with better and better objective values, thus a feasible solution can be generated quickly and then it can refined to converge towards the optimal one.

4.2. Aggregation

The generation of optimal aggregate project models corresponds to partitioning the project tree into sub-trees that represent activities of the project. The thoughput time of the activities should not exceed the limit of one week, while the height and the cardinality of the partitioning should be as small as possible. In [9] we have suggested polynomial time algorithms for solving such problems.

In order to check whether an activity fits into a week's capacity profile, one has to estimate the throughput time of the set of operations that constitute the activity. However, at the time of activity formation this throughput time can hardly be computed. Firstly, the set of activities competing for the limited resources is not known at this phase. Secondly, determining the minimal throughput time of a single activity is an NP-hard resource-constrained project scheduling problem in itself. Hence, we elaborated various *heuristic functions* for estimating the throughput time of an activity. In production environments where both resource constraints and complex precedence relations should be accounted for, we applied a priority-rule based scheduler that worked with the "greatest rank positional weight*" (GRPW*) rule [4].

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

4.3. Detailed constraint-based scheduling

As discussed above, we have modeled the scheduling problem as a *constraint optimization problem* [1]. The model uses variables (e.g., start time of operations), possible values (domains) of variables, constraints (resource and temporal) and an optimization objective. When solving this constraint problem, we are looking for those values of all the variables (in their corresponding domains) that satisfy all the constraints and are the best according to the given criteria. Typically, the so-called *maximum-type* objective functions, such as the *makespan* (the maximum of the end times), the maximum tardiness or the peak resource usage can be minimized efficiently by constraint-based techniques.

Solution techniques in constraint programming rely on an effective combination of inference and search. A foundational inference method is *constraint propagation*: it removes inconsistent values from the domains of the variables, i.e., values that provably cannot constitute a part of a solution. Propagation is executed every time the domain of some variable changes. Since the propagation machinery is incomplete, the solution has to be found by a search process. During search, new, artificial constraints are introduced that divide the original problem into separate alternatives. Search decisions and propagation are interwoven so that propagation can reduce the search space as soon as possible. If the constraint model becomes inconsistent in one of the alternatives, then work continues with the other ones, and – in the last resort – the system backtracks.

Constraint-based scheduling applies both the generic propagation mechanisms of constraint programming and domain specific propagators that fit the actual temporal and resource constraints of the scheduling problem. Specifically, temporal constraints between operations can be propagated by versions of the standard, so-called *arc-B-consistency* algorithm [11]. For instance, if a precedence constraint prescribes that operation A must be executed before operation B, then the earliest start time of B should be at least the earliest start time plus the duration of A. Once the time window of an operation is reduced, propagation tries to narrow the time windows of all the other ones that are linked to this operation by precedence constraints.

For propagating *resource constraints*, we apply the widely used method of *edge finding* [3]. Given a particular resource and a set of operations requiring this resource, edge finding tries to deduce which operations must be (or cannot be) scheduled first (or last) in this set. The algorithm investigates time windows where the total demand of operations exceeds the capacity of the resource. The conclusions drawn are of two types: new precedence constraints are posted between some operation, and the time windows of some operations are tightened. Note that the application of edge finding may prompt the further call of temporal propagators and *vice versa*.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Even though constraint propagation can help prune the search space significantly, scheduling of discrete resources is still a very hard problem to solve to optimality. Hence, we have embedded constraint propagation into a search process that produces a sequence of better and better solutions that converge to the optimal schedule. The solution method is – like that of the medium term planner – *any-time*, thus it can be used interactively. Some further methods that we applied to increase the efficiency of these solution techniques are described in detail in [10].

5. IMPLEMENTATION AND INDUSTRIAL APPLICATION

The above PPS framework has been developed in the course of the "Digital Factory, Production Networks" NKFP project. During this work, a prototype PPS systems – the so called Proterv-II has also been implemented that supports production and capacity planning on medium term and detailed job-shop schedule on short term. Proterv-II has graphical user interface that facilitates its use as a *decision support system* (DSS) at both levels of the decision hierarchy. To implement the suggested algorithms, we have used professional constraint solver and optimization software [6].

Experiments with the prototype system have been carried out on real-world industrial data. Typical projects consisted of 20 to 500 discrete manufacturing operations, with processing times in the range of 0.5 to 120 hours. Operations required both machine and human resources. The project trees were generated from standard BOM and routing databases. The aggregate project models were generated from these trees consisted of 1 to 10 activities. The resource pool contained ca. 100 individual and 50 group resources. The horizon of the planning problem was set to 15-30 weeks. Under the above initial conditions, our approach was capable to generate optimal production plans even for several hundreds orders. The first feasible solution of the studied planning and scheduling problems could be created within a few seconds, and if one looked for the optimal plan, larger runtime had to be set.

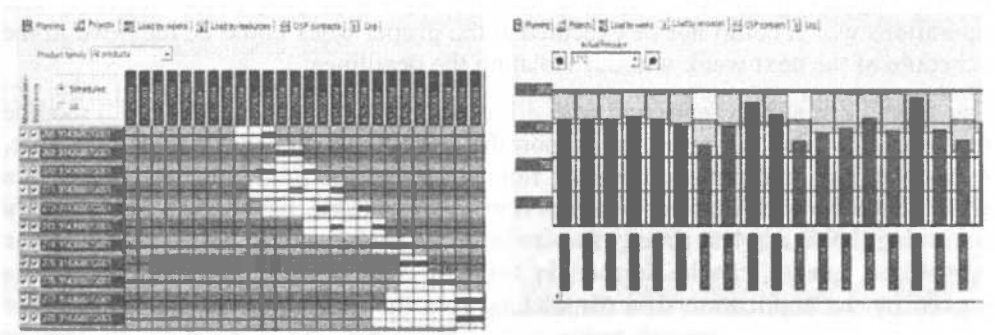


Figure 3: Medium term plan and load of a resource

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Fig. 3 presents a fragment of a production and a capacity plan. In the production plan the white areas show the allowed time windows of the projects and dark fields indicate weeks when one or more activities of the projects have to be performed. Since the minimal WIP level was a criterion, the projects start as close to their due dates as possible without violating them. The breaks in the production are caused by resource shortages. The capacity plan shows the load of a certain resource through the planning horizon. It looks really to be an overloaded factory: the internal capacities are completely exploited and in several weeks the use of extra capacities is also required.

In Fig. 4 a segment of a short time schedule is presented. The schedule contains the operations of the activities – connected parts of the project trees – distributed in the week, while the resource view shows which operations have to be performed on a certain resource or resource group.

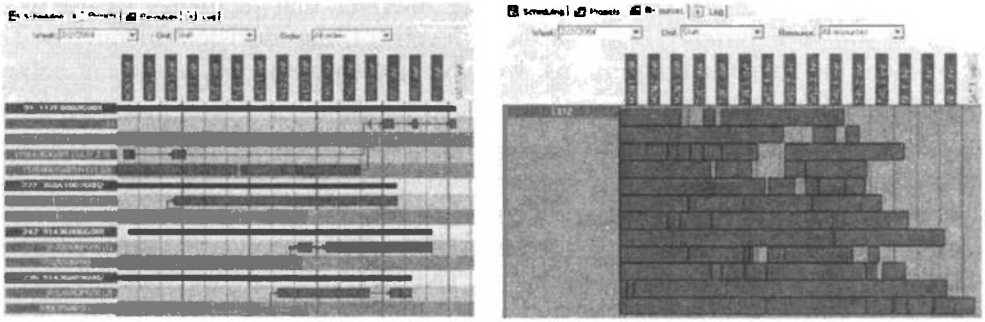


Figure 4: Short term schedule and load of a machine group

During the execution of the produced schedules various disorders (e.g., machine breakdowns or distortions come from model uncertainties) may occur. These situations have been analyzed and evaluated by discrete event simulations [7]. The simulation experiments have shown that in many cases the medium term plans are robust enough to remain feasible despite numerous unexpected events. The operations which could not be executed at the proper week could be included in the schedule of the next week without violating the deadlines.

Our model can also be enriched with a Production Activity Control (PAC) module (see Fig. 1), whose purpose is to support the execution of schedules under dynamic, ever-changing conditions at the shop floor. In a realistic production environment a rigid schedule with fixed operation starting times can hardly be executed. By removing the fixed start times of operations and keeping only the sequence of the operations, queues can be formed in front of each resource. Operations can be picked by the application of a dispatching rule that keeps the queues all the time consistent with the original precedence constraints. As a reaction to smaller changes in the environment, the PAC module should perform synchronization – a

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

re-scheduling based on the operation queues with respect to the newly emerged constraints – by the application of simple and fast modifications that cause only minimal perturbation to the original schedule. However, shop foremen should have the final word in deciding on schedule execution: they have the responsibility to postpone or remove some operations from the queues and to ask for a global rescheduling in case of major disorders.

6. CONCLUSIONS

In this paper we have presented an overview of the main concepts and solution methods of a hierarchical production planner and scheduler system. Novel feature of our approach is that it takes a project-oriented approach for solving the planning problem. Hence, decisions on the time of making the customer orders are combined with decisions concerning the load on resources. Further on, these decisions are made with regard to the actual demand and available production capacities. This approach results in better due-date observance and executable production plans.

We have also analyzed the role of aggregation that links models of production planning and scheduling. The proposed aggregation method enables PPS to work on common product, resource and production technology data on both levels of the decision hierarchy. Our experiences confirm also that proper aggregation is a major prerequisite for generating production plans that can really be refined to executable schedules.

ACKNOWLEDGEMENTS

This work has been supported by the NKFP grants No. 2/040/2001, 2/010/2004 and the OTKA grant No. T046509. The authors would like to thank Ferenc Erdélyi, Tamás Kis and László Monostori for their help and support.

REFERENCES

[1] BAPTISTE, PH., LE PAPE, C., NUIJTEN, W.: *Constraint-Based Scheduling.* Kluwer Academic Publishers, 2001.

[2] BITRAN, G.R. TIRUPATI, D.: *Hierarchical Production Planning.* In: Graves, S.C. Rinnooy Kan A.H.G. Zipkin, P.H. (eds), Logistics of Production and Inventory, North Holland, 1993, pp. 523-568.

[3] CARLIER, J., PINSON, E.: *A practical use of Jackson's pre-emptive schedule for solving the job-shop problem.* Annals of Operations Research, 26, 1990, pp. 269-287.

[4] DEMEULEMEESTER E.L., HERROELEN, W.S.: *Project Scheduling: A Research Handbook.* Kluwer Academic Publishers, 2002.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[5] HACKMAN, S.T., LEACHMAN, R.C.: *An Aggregate Model of Project-Oriented Production*. IEEE Transactions on Systems, Man, and Cybernetics, 19, 2, 1989, pp. 220-231.

[6] *Ilog Scheduler 5.1 Users Manual*. 2001.

[7] KÁDÁR B., PFEIFFER A., MONOSTORI L.: *Discrete Event Simulation for Supporting Production Planning and Scheduling Decisions in Digital Factories*. Proceedings of the 37th CIRP International Seminar on Manufacturing Systems, 2004, pp. 441-478.

[8] KIS T.: *A Branch-and-Cut Algorithm for Scheduling Projects with Variable-Intensity Activities*. Mathematical Programming, 2005 february.

[9] KOVÁCS A., KIS T.: *Partitioning of Trees for Minimizing Height and Cardinality*. Information Processing Letters, 89, 4, 2004, pp. 181-185.

[10] KOVÁCS A., VÁNCZA J.: *Completable partial solutions in constraint programming and constraint-based scheduling*. In Proc. of the 10th International Conference on Principles and Practice of Constraint Programming (Springer LNCS 3258), 2004, pp. 332-346.

[11] LHOMME, O.: *Consistency techniques for numeric CSPs*. In Proc. of IJCAI'93 - the 13th International Joint Conference on Artificial Intelligence, 1993, pp. 232-238.

[12] MÁRKUS A., VÁNCZA J., KIS T., KOVÁCS A.: *Project Scheduling Approach to Production Planning*. CIRP Annals Manufacturing Technology, 52, 1, 2003, pp. 359-362.

[13] VÁNCZA J., KIS T., KOVÁCS A.: *Aggregation – The Key to Integrating Production Planning and Scheduling*. CIRP Annals - Manufacturing Technology, 53, 1, 2004, pp. 377-380.

[14] VOLLMANN, T. E., BERRY W.L., WHYBARK D.C.: *Manufacturing Planning and Control Systems*. McGraw-Hill, 1997.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 37-51

APPROXIMATE NEAREST NEIGHBOR SEARCH FOR LABELLED TREES

LÁSZLÓ KOVÁCS
Department of Information Technology, University of Miskolc
kovacs@iit.uni-miskolc.hu

TIBOR RÉPÁSI
Department of Information Technology, University of Miskolc
repasi@iit.uni-miskolc.hu

ERIKA BAKSA-VARGA
Department of Information Technology, University of Miskolc
iitev@uni-miskolc.hu

[Received October 2004 and accepted January 2005]

Abstract. In many scientific areas there is a frequent need to extract a common pattern from multiple data. In most cases, however, an approximate but low cost solution is preferred to a high cost exact match. To establish a fast search engine an efficient heuristic method should be implemented. Our investigation is devoted to the approximate nearest neighbor search (ANN) for unordered labeled trees. The proposed modified best-first algorithm provides a $O((N_q+N_b)\cdot M + K\cdot N_q\cdot N_b/M)$ cost function with simple implementation details. According to our test results, realized with smaller trees where the brute-force algorithm could be tested, the yielded results are a good approximation of the global optimum values.

Keywords: tree matching, approximate nearest neighbor search

1. INTRODUCTION

In many scientific areas there is a frequent need to extract a common pattern from multiple data. The most common structure of the data is the hierarchy or tree. The task is to determine the set of sub-trees having the best matching with the pattern. Some important application areas for sub-tree matching are

pattern recognition, where the objects (for example pictures) are described by a tree structure.

molecular biology, where the real topology of RNA and of other molecules is a tree. From the topological similarities it is often possible to infer similarities in the function of the molecules.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

natural language processing, computational linguistics where dictionary definitions are stored in a lexical database. The definitions are represented syntactically as trees.

programming languages, where one of the main metadata structures is the tree structure. The parse trees, the operation trees or syntax trees are often used in the algorithms.

information systems, where the most common new storage format is the XML tree. The XML is seeing increased use and promises to fuel even more applications in the future. An XML document can be modeled as a tree. Each node in this tree corresponds to an element in the document. Each edge represents inclusion of the element corresponding to the child node under the element corresponding to the parent node in the XML file.

In the applications mentioned above the nodes of a tree are characterized by one or more attributes. Such attributes can be the type of the molecule (a node corresponds to a molecule) or the type of the operation (a node corresponds to an operation). The description vector of the nodes is called the label of the nodes. In some areas, not only the node types but also the order of nodes is important. In this case, the children are assigned to an ordering number in the scope of the parent. XML documents, for instance have an ordered and labeled tree structure. In our investigation we are focusing on unordered labeled tree structures. A recent workshop report from Yale suggested that more research should be undertaken to improve the heuristic search using algorithms designed to meet the demand made by increasingly large tree datasets [1].

2. RELATED WORKS

In this section we review the different researched approaches to comparing trees, as well as the algorithms developed so far to solve these problems. **P. Bille** published an extensive survey [2] on comparing trees with exact searching methods. As a conclusion from his work it turned out that all of the unordered versions of the problems in general are NP-hard. Indeed, the tree edit distance and alignment distance problems are even MAX SNP-hard. However, using special constraints polynomial time algorithms are available, just like for the ordered versions of the problems. These are all based on the classic technique of dynamic programming.

The *general ordered tree edit distance*, also called tree-to-tree correction problem was also fully reviewed in Technical Report 95-372 [3]. The problem was introduced by **Tai** [4] as a generalization of the string edit distance problem. His algorithm, which solved the problem without recursion, has its time and space complexity in $O(nmd^2d'^2)$, where n and m are the maximum number of children from any node in each of the trees, while d and d' are the maximum depth of the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

trees. **Zhang and Shasha** [5] numbered the trees using postorder traversal instead of preorder, so the algorithm's space complexity is $O(nm)$, while its time complexity is $O(nmdd')$. **Klein** [6] solved the problem in $O(n^3 logn)$ time and $O(nm)$ space. In his paper he proved that the algorithm can be extended to unrooted ordered trees within the same time and space bounds. **Chen** [7] applied fast matrix multiplication to solve the problem. The *unordered* version of the problem is NP-complete even for binary trees with a label alphabet of size 2. It was shown in [8] that under special restrictions polynomial time algorithms exist.

There are other variants of the edit distance problem as well. One of them is the *unit cost edit distance*, where unit cost is defined as the number of edit operations required. In [9] the ordered version of the problem is considered and an algorithm with $O(u^2 min\{n,m\} min\{l,l'\})$ time need is introduced, where l and l' are the number of leaves of the trees. The algorithm uses techniques from **Ukkonen** [10], and **Landau** and **Vishkin** [11]. The recursive solution of **Selkow** [12] used the basic operations, but insertions and deletions were restricted to leaf nodes only, which made the algorithm very simple and therefore its time complexity is $O(nm)$. This is therefore sometimes referred to as the *1-degree edit distance*. **Chawathe** [13] utilizes the same restrictions, but in cases when external memory is needed to calculate the edit distance.

Tree inclusion, a special case of edit distance, is the problem to decide if tree T_1 can be included in T_2. T_1 is included in T_2 if there is a sequence of delete operations performed on T_2 which make T_2 isomorphic to T_1. For the *ordered* tree inclusion problem **Kilpeläinen and Mannila** [14] presented the first polynomial time algorithm using $O(nm)$ time and space. A more space efficient version of this was given in [15] using $O(nd')$ space. Later **Richter** [16] and **Chen** [17] developed more complex algorithms. In [14], [18] it is shown that the *unordered* tree inclusion problem is NP-complete. In spite of this an algorithm using $O(mni2^{2i})$ time exists.

Torsello and Hancock [19] prove, that a tree t' can be generated from a tree t with a sequence of node removal operations if and only if t' is an obtainable subtree of the directed association graph. Consequently the minimum cost edited tree isomorphism between two trees is a maximum common consistent subtree of the two directed association graphs if the node removal cost is uniform, and this result can also be extended to non-uniform cost. The background for this lies in [20], where the relationship between graph edit distance and the size of the maximum common subgraph is shown, and also their computational equivalence is demonstrated. This is an important observation since it has been established by **Barrow** and **Burstall** [21] that the maximum common subgraph problem may be transformed into a maximum clique problem using a derived structure referred to as the association graph. **Pelillo et al.** [22], for instance, transform the tree

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

isomorphism problem into a single max clique problem, a technique already used for the generic graph isomorphism problem. To obtain a maximal tree match, i.e. a maximal solution to the max clique problem, they use relaxation labeling. **Wang et al.** [23] considers the *largest approximately common subtree* problem for ordered, labeled trees using the edit distance to measure the dissimilarity of two trees. They present a dynamic programming algorithm, which runs as fast as the fastest known algorithm for computing the edit distance of trees.

This problem was investigated for *unordered* trees by **Khanna, Motwani and Yao** [24]. They created an algorithm for trees of bounded degree with performance ratio $O(nloglogn/log^2n)$ and then extended this to trees of unbounded degree with at most poly-log labels, obtaining a ratio of $O(n(loglogn)^2/log^2n)$. **Akutsu and Halldórsson** [25] also considers the approximation of the *largest common subtree* (and its special variation, the largest common edge subgraph) and *largest common point set* problems for unordered trees (and for ordered trees as a special case), and a general search algorithm is presented which approximates both problems within a factor of $O(n/logn)$. For trees of bounded degree an improved algorithm is developed which approximates the largest common subtree within a factor of $O(n/log^2n)$. A large amount of work has been performed for comparing unordered trees based on various distance measures, especially on edit distance as the most commonly used distance measure. **Shasha et al.** [26], however, proposed a new approach, called Atree-Grep. They addressed the *approximate nearest neighbor search* problem for unordered labeled trees. Their algorithm, called 'pathfix', consists of two phases. First, the paths of the trees are stored in a suffix array and then the number of mismatching paths are counted between the query tree and the data tree. To speedup the search, they use a hash-based technique to filter out unqualified data trees at an early stage of the search. The algorithm has been implemented into two special Web-based search engines and proved to be fast, particularly when the dictionary size of node labels is large.

Other widely researched problems include *tree pattern matching* [27, 28, 29, 30, 31], *maximum agreement subtree* [32, 33] and *smallest common supertree* [34, 35].

3. DISTANCE MEASURES FOR TREE COMPARISON

As can be seen, most of the proposals in subtree matching are based on the edit distance between trees. This distance metric is a natural extension of the edit distance concept used for string comparisons. This metric provides an exact distance measurement between the trees. The drawback of these algorithms is the high cost of the computations. In the case of online applications with large tree datasets, the execution time is a crucial factor. In these kinds of applications, an approximate but low cost solution is preferred to a high cost exact solution. Our investigation is devoted to the approximate nearest neighbor search (ANN) for unordered labeled trees. Our goal is to construct an efficient heuristic method for

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

the ANN problem. Since the ANN problem for edit distance metric is an NP-problem as is proven in [8], a modified distance definition is introduced.

Let D denote a domain set. This contains the node labels. The symbol T denotes an unordered, labeled tree. The following denotations related to the tree structure are used in the paper:

n	a node of the tree
$l(n)$	the label of node n, $l(n) \in D$
T_D	set of unordered, labeled trees on D
$V(T)$	vertices of T
$E(T)$	edges of T
$r(T)$	the root node of T

The goal of the investigation is to find the neighboring trees based on the similarity values. The distance or similarity is usually measured by a metric function. A space X is called metric space if a $d(A,B)$ real non-negative function of two objects is defined with the following properties:

For every $A \in X$, $B \in X$, and $C \in X$

1. $d(A, B) = 0$ if and only if $A = B$ (the distance is 0 if and only if the points coincide),
2. $d(A, B) = d(B, A)$ (the distance from A to B is the same as the distance from B to A),
3. $d(A,B) + d(B,C) \geq d(A,C)$ (the sum of two sides of a triangle is never less than the third side).

The $d(A,B)$ function is known as the distance between the two points.

In the case of edit distance, a set of elementary transformation functions is defined on T_D. This set is denoted as E_D. The cost value of the elementary transformations is a non-negative real number. The corresponding cost function is denoted by

$$c \quad E_D \rightarrow R^+$$

It is assumed that T_D is closed to E_D, i.e.

$$e \quad T_D \rightarrow T_D \quad \forall e \in E_D,$$

$$\forall T_1, T_2 \in T_D \quad \exists e_1, e_2, \ldots, e_m \in E_D: e(T_1) = e_m \circ e_{m-1} \circ \ldots e_2 \circ e_1(T_1) = T_2.$$

Let us denote the set of chain of transformations from T_i to T_j by $E_{i,j}$. The cost of chain e is defined as the sum of the single transformation steps:

$$c(e) = \Sigma c(e_i).$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The edit distance between T_i and T_j is defined as the minimal cost of transformation chains from T_i to T_j:

$$c_{i,j} = min\{ c(e) / e \in E_{i,j} \}.$$

Usually, like in [36] the following elementary e operations are defined for tree objects:

relabel: assigns a new node name to the root of the tree,
insert: inserting a new node into the children of the root node,
delete: deleting a node from the children of the root node,
insert tree: inserting a tree under the root node,
delete tree: deleting a tree from the children of the node.

The list of elementary transformations with minimal cost is usually generated with a dynamic programming method. According to [2, pp.7], the tree distance value can be calculated using the following recursive formula:

$$\begin{aligned} d(0,0) &= 0 \\ d(F,0) &= d(F\text{-}v,0) + c(v,0) \\ d(0,F) &= d(0,F\text{-}v) + c(0,v) \\ d(F_1,F_2) &= min \begin{cases} d(F_1\text{-}v,F_2) + c(T(v),0) \\ d(F_1,F_2\text{-}v) + c(0,T(v)) \\ d(F_1\text{-}T(v),F_2\text{-}T(w)) + c(T(v),T(w)) \end{cases} \end{aligned}$$

where F denotes a tree, $T(v)$ denotes a tree with root element v, and $c(x,y)$ denotes the cost of transforming node x to node y. The computation cost of the basic dynamic programming method for trees is $O(|T|^4)$. This is a very high cost value for an ANN problem, as the distance computation should be calculated for a large number of pairs. It is proved in [26] that the ANN problem for edit distance metric is an NP-complete problem. In spite of this difficulty, most of the proposals for ANN searching for trees use the edit distance measure. There are very few proposals that apply a simplified distance function to provide a lower cost solution.

A good example for this approach is [26], where the distance from T_1 to T_2 is measured with the total number of root-to-leaf paths in T_1 that do not appear in T_2. The nodes in T_2 that do not appear in T_1 can be freely removed. As can be seen, this definition introduces an asymmetric distance concept. In the definition T_1 denotes the query tree while T_2 is the searched tree. In our approach, another simplified distance function was selected.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

4. MODIFIED BEST-FIRST ALGORITHM

Two trees are said to be similar if they have similar vertices with similar edges. During the editing process every vertex of the query tree is either transformed into a vertex of the base tree or it is deleted. Based on this transformation, every vertex of the query tree can be mapped either to a target vertex or to the sink symbol. Using this approach, a generalized mapping can be defined between the query and the base tree. We define $m(\,)$ as a monomorphism from T_1 to T_2 in the following way:

1. $m \quad V(T_1) \rightarrow V(T_2) \cup \varepsilon$
2. $\forall v, m(v) \in V(T_2) \quad l(v) = l(m(v))$
3. $\forall v_1 \neq v_2, m(v_1), m(v_2) \in V(T_2) \quad m(v_1) \neq m(v_2)$
4. $\forall v_1 \neq v_2, m(v_1), m(v_2) \in V(T_2): v_1 < v_2 \Leftrightarrow m(v_1) < m(v_2)$

According to the first property, every node in T_1 is mapped either to a node in T_2 or is deleted, i.e. it is mapped to the ε symbol. The second property says that a vertex should be mapped only to nodes of the same label. Due to the third property, the different query vertices can not be mapped to the same base vertex. The fourth property is called ancestor condition, the ancestor-descendants relationship among the query vertices must be preserved in the target tree, too.

Other types of relationships among the query vertices are neglected and not preserved. In this approach, the sibling vertices may be mapped to parent-child vertices, if the existing parent-child relationships are preserved. The parent-child relationships are the only important information stored in the query tree. The absence of an edge means in our approach a 'do not know' information. In this case, we don not care about the existence of an edge between the mapped vertices in the base tree. Figure 1 shows an example for this mapping.

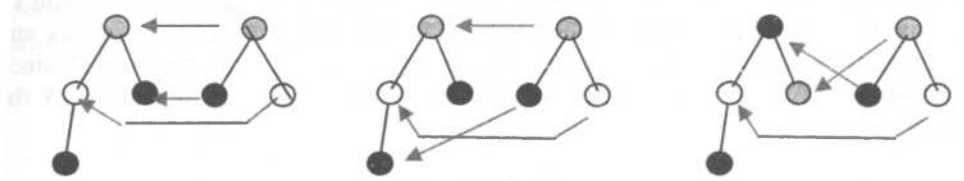


Figure 1: Distance mapping example

Figure 1a) and Figure 1b) show valid mappings. The sibling nodes in the query tree are mapped to sibling nodes in Figure 1a), and to parent-child nodes in Figure 1b). Figure 1c) shows an invalid mapping as the parent-child relationship is not preserved. This kind of similarity value differs from the usual edit distance in the following aspects: 1) it does not take the re-labeling operation into account, and 2) only one side of the operands can be deleted.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Based on this mapping, a similarity value can be defined between two trees. The cost of mapping m is defined as the sum of the vertex mappings related to the query tree:

$$cost(m) = \Sigma_{n \in V(T)} c(n),$$

where

$$c(n) \begin{cases} C_2, \text{ if } m(n) = \varepsilon \vee m(r(T)) = \varepsilon \\ 0, \text{ if } n = r(T) \wedge m(r(T)) \neq \varepsilon \\ C_1 \cdot (d(m(n), m(a(n)) - 1) \text{ otherwise.} \end{cases}$$

In this definition, $a(n)$ denotes the nearest ancestor of n in the query tree which is mapped to a non-ε element. If the root of the query tree is mapped to ε then $c(n)$ is C_2, otherwise the path from n to $r(T)$ (excluding n and including $r(T)$) contains minimum one vertex mapped to a non-ε value. In this case both $m(n)$ and $m(a(n))$ are non-ε elements. The $d(\,)$ function denotes the length of path from $m(a(n))$ – $m(n)$ in the base tree. As mapping m preserves the parent-child relationship, $m(a(n))$ is an ancestor of $m(n)$. Thus $d(\,)$ yields a positive integer value. C_1 and C_2 are cost units. C_1 corresponds to gap-lengths between two preserved vertices and C_2 denotes the cost for vertex deletion. In our approach, C_2 is greater than C_1 since the absence of an element means a larger difference than the relocation of the element.

In this definition, $a(n)$ denotes the nearest ancestor of n in the query tree which is mapped to a non-ε element. If the root of the query tree is mapped to ε then $c(n)$ is C_2, otherwise the path from n to $r(T)$ (excluding n and including $r(T)$) contains minimum one vertex mapped to a non-ε value. In this case both $m(n)$ and $m(a(n))$ are non-ε elements. The $d(\,)$ function denotes the length of path from $m(a(n))$ – $m(n)$ in the base tree. As mapping m preserves the parent-child relationship, $m(a(n))$ is an ancestor of $m(n)$. Thus $d(\,)$ yields a positive integer value. C_1 and C_2 are cost units. C_1 corresponds to gap-lengths between two preserved vertices and C_2 denotes the cost for vertex deletion. In our approach, C_2 is greater than C_1 since the absence of an element means a larger difference than the relocation of the element.

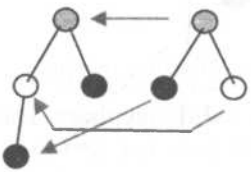


Figure 2: An example for mapping

As an example, let the calculation of the mapping cost for Figure 2 stay here. The cost for root mapping is 0. The cost for white node is also 0 (there is no gap in the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

mapped path). The cost for black node is 1 (one node length gap). The total cost is 0 + 0 + 1 = 1. We remark that in some applications it seems useful to introduce a weight factor in the cost expression. In this case the different edges may have different importance factors.

It can be seen that the distance measure based on this cost value does not meet the requirements of a metric space. The metric distance function should be symmetric while the given cost function is asymmetric. The roles of the query and base trees are distinguished. This corresponds to our intention, as we try to find a best matching sub-tree included in the base tree. The goal is to find a mapping with minimal cost value.

Taking a query tree T_1 with N_q nodes and a base tree T_2 with N_b nodes, the number of potential mappings is $O(N_b! / (N_b-N_q)!)$. Although the ancestor criteria restricts the set of potential mappings, the number of possible enabled mappings is too high. It would be very costly to test all of the possible mappings. Thus some kind of heuristics should be applied to speed up the matching process. In our investigation a variant of the best-first search method was selected.

The best first search method works on a state-tree. Each node of the tree is assigned to a cost value. The goal is to find the path with the minimal cost value. The best-first search divides the nodes into three distinct groups: the nodes tested (G_1), the nodes ready to be tested (G_2), and the rest (G_3). Initially, G_1 is empty and G_2 contains only the root element. In a loop, the node from the ready state with the best (minimal) cost value is selected to be tested. During the test, the children of the node are evaluated and moved from the G_3 group into the G_2 group. The loop terminates if a leaf node is selected for testing.

In the applied variant, the nodes of the state-tree are assigned not to the vertices but to the vertex mappings of the query tree. Thus each node represents a decision about the mapping function. The state-tree is expanded and traversed in the following way:

1. Generating a label vector for every node. The label vector contains the counter values for the different labels related to the nodes in the descendant set. This vector works similar to one-grams used in the string distance problem. In the example shown in Figure 3a), the description vector for the root node is $lv(3,2,1,3)$, where the first dimension is assigned to the green label, the second to the red label, the third to the blue label and the fourth to the black label.
2. Calculation of the label vectors for the query tree.
3. Selecting maximum K nodes in the base tree with the same label as the root of the query tree and with the first K best similarity values regarding the label vectors. The similarity value for label vectors is defined by

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$d(l_q,l_b) = \Sigma_j \max(l_{qj}-l_{bj},0)$$

where l_q belongs to the query tree and l_b to the base tree.

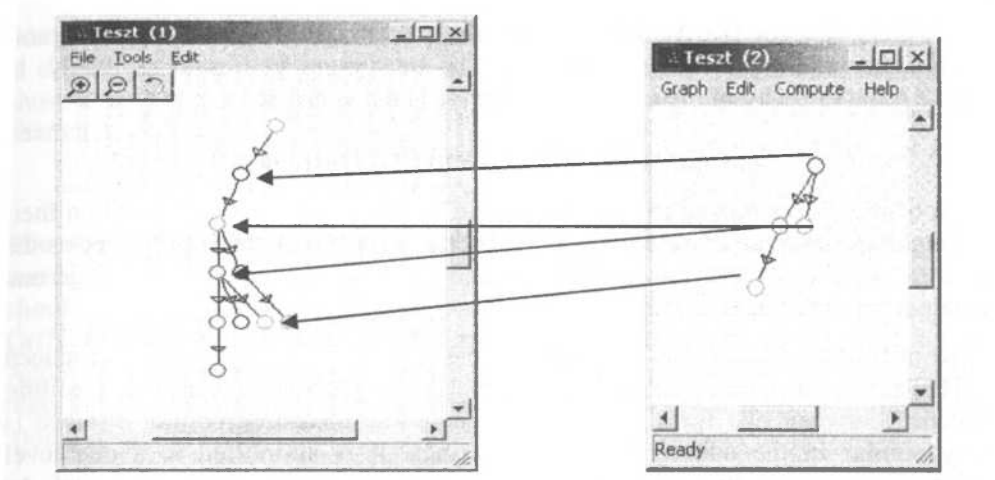
4. Loop on the selected nodes. Let w denote the vertex actually tested. Map the root of the query tree to w. Empty G_2 and G_1.
5. Insert the mapping of w into G_2.
6. Take the element x from G_2 with the lowest cost value. Move x from G_2 into G_1. Disable the other mappings in G_2 from x or to $m(x)$.
7. Test the children vertices of x considering the query tree. For every vertex generate the set of possible mappings. Evaluate these mappings and insert them into G_2.
8. If G_2 is empty, the procedure terminates. The sum of cost values for the selected mappings is the approximation of the best mapping cost value for w, denoted by $C(w)$. Go back to step 4.
9. Return $min\{C(w)\}$ as the approximation of the optimal mapping cost.

The cost of generating the label vectors is $O((N_q+N_b)M)$ as every vertex should be accessed only once. The label vector of a node can be built from the label vectors of its children. In the cost expression M denotes the number of different label values. M corresponds to the length of the label vectors. During the best-first search N_q vertices are tested and expanded. A vertex from the query tree may be mapped to $O(N_b/M)$ vertices in average. As the best-first search is repeated by K times, the cost estimation for the algorithm is $O((N_q+N_b)\cdot M + K\cdot N_q\cdot N_b/M)$. Thus the cost is linear in both N_q and N_b. This cost is a significant reduction compared with the $O(M\cdot N_b! / (N_b-N_q)!)$ value for the brute force search method.

5. RESULTS

The implementation tests show a similar linearity for the computation costs. The test programs are implemented in the Scilab language. The next small example illustrates the cost relations between the brute-force and the heuristic method. The base tree has 10 vertices and is shown in Figure 3a). The query tree has 4 vertices and is shown in Figure 3b). The number of labels is 4. The trees were generated randomly.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM



a) base tree b) query tree

Figure 3: Mapping example

The elements of the best mapping are given in the Figure with blue arrows. The cost value is only 1. Both methods can detect this optimal mapping but with a very different cost value. Table 1 shows the execution cost values for the investigated methods related to this example query.

Table 1: Comparing the costs of execution

Method	**Cost**
Brute-force	69.48 sec
M. Best-first	00.08 sec
Selkow	02.46 sec

To test the cost values for examples of larger tree sizes, a test run was implemented with values $N_q = 12$, $N_b \in [15..500]$. The cost values are shown in Figure 4.

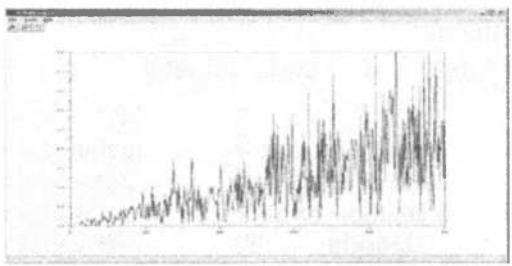


Figure 4: Cost values for larger trees

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The x-axis denotes the N_b value, while the y-axis shows the computation cost (where the maximum value is 3.6 sec). The trees were generated randomly. In Figure 4 the linearity of the cost function is well demonstrated.

6. COMPARISON WITH EDITING DISTANCE

Most of the works related to the comparison of unordered trees are based on their editing distance or Levenshtein distance. The Levenshtein distance is defined as the smallest number of insertions, deletions, and substitutions required to change one string or tree into another [37].

Most of the related works in this field are based on **Selkow**'s algorithm introduced in [12] and summerized by **P. Bille** in [2]. Selkow's algorithm calculates the editing distance between two forests of ordered trees. The measured editing distance is very similar to the editing distance of strings. It is simplified to a one level comparison, so edition is necessary each time the root nodes are not identical. As an ordered tree is a special case of an unordered tree, Selkow's algorithm can be used for unordered trees as well. It is a basic algorithm which needs a tremendous computation power of $O(N^2 \cdot M^2)$, where N and M are the node numbers of the trees, to find the editing order of the least cost. In case of unordered trees the algorithm has to take each order of the tree in account, so the computation cost will grow by $O(2^N \cdot 2^M)$ for the repetitions on each possible permutation of the trees. The basic operations Selkow's algorithm makes use of node delete, node insert and node substitution or relabeling. Each operation can have different cost functions. The algorithm itself is recursive very much like above mentioned in section 3. It is working in the following way:

1. searching for the roots of the source forests,
2. extracting each tree of the forests,

3.a computing the cost (by doing a recursion) of deleting the root-node for each tree of one forest,

3.b computing the cost (by doing a recursion) of inserting a new root-node for each tree of the other forest,

3.c computing the cost (by doing a recursion) of substituting the root node for each tree of both forests,

4. selecting the minimum of all costs and returning it to the upper level of the recursion.

The generalization of the algorithm to forests is inevitable due to the fact, that the algorithm is recursive and that deleting the root node of a tree will result in a forest. We have tested an implementation of Selkow's algorithm on small trees. Running our implementation of the algorithm with the trees shown on Figure 3 we got the editing distance of 10 units, considering 1 as the cost of each edit operation. The

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

time needed to complete the calculation in the same environment was 2.463 sec as is shown in Table 1. Running the algorithm with randomly generated trees will show a very noisy cost function, however, the limits should show the $O(N^2 \cdot M^2)$ characteristics.

7. CONCLUSION

The approximate sub-tree search for trees with edit distance metric is an NP-complete problem. To establish a fast search engine an efficient heuristic method should be implemented. The proposed modified best-first method provides a $O((N_q+N_b)\cdot M + K\cdot N_q \cdot N_b/M)$ cost function with simple implementation details. According to our test results, realized with smaller trees where the brute-force algorithm could be tested, the yielded results are a good approximation of the global optimum values.

REFERENCES

[1] CRACRAFT, J. DONOGHUE, M.: *Assembling the tree of life: Research needs in phylogenetics and phyloinformatics*. Report from NSF Workshop, Yale University, July 2000.

[2] BILLE, P.: *Tree Edit Distance, Alignment Distance and Inclusion*. IT University of Copenhagen, Technical Report Series TR-2003-23, ISSN 1660-6100, March 2003.

[3] BARNARD, CLARKE, DUNCAN: *Tree-to-Tree Correction for Document Trees*. Technical Report 95-372, Queen's University Canada, January 1995.

[4] TAI: *The Tree-to-Tree Correction Problem*. Journal of the Association for Computing Machinery (JACM), 26:422-433, 1979.

[5] ZHANG, SHASHA: *Simple fast algorithms for the editing distance between trees and related problems*. SIAM Journal of Computing, 18:1245-1262, 1989.

[6] KLEIN: *Computing the edit-distance between unrooted ordered trees*. In Proceedings of the 6th annual European Symposium on Algorithms (ESA) 1998, pp. 91-102.

[7] CHEN: *New algorithm for ordered tree-to-tree correction problem*. Journal of Algorithms, 40:135-158, 2001.

[8] ZHANG, STATMAN, SHASHA: *On the editing distance between unordered labeled trees*. Information Processing Letters, 42:133-139, 1992.

[9] SHASHA, ZHANG: *Fast algorithms for the unit cost editing distance between trees*. Journal of Algorithms, 11:581-621, 1990.

[10] UKKONEN: *Finding approximate patterns in strings*. Journal of Algorithms, 6:132-137, 1985.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[11] LANDAU, VISHKIN: *Fast parallel and serial approximate string matching.* Journal of Algorithms, 10:157-169, 1989.

[12] SELKOW: *The tree-to-tree editing problem.* Information Processing Letters, 6(6):184-186, 1977.

[13] CHAWATHE: *Comparing hierarchical data in extended memory.* In Proceedings of VLDB, 1999, pp. 90-101.

[14] KILPELÄINEN, MANNILA: *Ordered and unordered tree inclusion.* SIAM Journal of Computing, 24:340-356, 1995.

[15] KILPELÄINEN: *Tree Matching Problems with Applications to Structured Text Databases.* PhD Thesis, University of Helsinki, Department of Computer Science, 1992.

[16] RICHTER: *A new algorithm for the ordered tree inclusion problem.* In Proceedings of the 8th Annual Symposium on Combinatorial pattern Matching (CPM), in Lecture Notes of Computer Science (LNCS), vol. 1264, pp 150-166, Springer 1997.

[17] CHEN: *More efficient algorithm for ordered tree inclusion.* Journal of Algorithms, 26:370-385, 1998.

[18] MATOUSEK, THOMAS: *On the complexity of finding iso- and other morphisms for partial k-trees.* Discrete Mathematics, 108:343-364, 1992.

[19] TORSELLO, HANCOCK: *Computing approximate tree edit distance using relaxation labeling.* Pattern Recognition Letters 2003, PII: S0167-8655(02)00255-6, 2002.

[20] BUNKE, KANDEL: *Mean and maximum common subgraph of two graphs.* Pattern Recognition Letters 21, 2000, pp. 163-168.

[21] BARROW, BURSTALL: *Subgraph isomorphism, matching relational structures and maximal cliques.* Information Processing Letters 4, 1976, pp. 83-84.

[22] PELLILO et al.: *Matching hierarchical structures using association graphs.* IEEE PAMI 21, 1999, pp. 1105-1120.

[23] WANG et al.: *An Algorithm for Finding the Largest Approximately Common Substructures of Two Trees.*

[24] KHANNA, MOTWANI, YAO: *Approximation algorithms for the largest common subtree problem.* Technical Report, Stanford University, 1995.

[25] AKUTSU, HALLDÓRSSON: *On the Approximation of Largest Common Subtrees and Largest Common Point Sets.* Science Institute University of Iceland, October 1997.

[26] SHASHA et al.: *AtreeGrep Approximate Searching in Unordered Trees.* In Proceedings of SSDBM 2002, Edinburgh, July 2002, pp. 89-98.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[27] KOSARAJU: *Efficient tree pattern matching*. In Proceedings of the 30th IEEE Symposium on the Foundations of Computer Science (FOCS), 1989, pp. 178-183.

[28] DUBINER, GALIL, MAGEN: *Faster tree pattern matching*. In Proceedings of the 31st IEEE Symposium on the Foundations of Computer Science (FOCS), 1990, pp. 145-150.

[29] HOFFMANN, DONNELL: *Pattern matching in trees*. Journal of the Association for Computing Machinery (JACM), 29(1):68-95, 1982.

[30] RAMESH, RAMAKRISHNAN: *Nonlinear pattern matching in trees*. Journal of the Association for Computing Machinery (JACM), 39(2):295-316, 1992.

[31] ZHANG, SHASHA, WANG: *Approximate tree matching in the presence of variable length don't cares*. Journal of Algorithms, 16(1):33-66, 1994.

[32] KESELMAN, AMIR: *Maximum agreement subtree in a set of evolutionary trees – metrics and efficient algorithms*. In Proceedings of the 35th Annual Symposium on the Foundations of Computer Science (FOCS), 1994, pp. 758-769.

[33] FARACH, THORUP: *Fast comparison of evolutionary trees*. In Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 481-488.

[34] NISHIMURA, RAGDE, THILIKOS: *Finding smallest supertrees under minor containment*. International Journal of the Foundations of Computer Science, 11(3):445-465, 2000.

[35] GUPTA, NISHIMURA: *Finding largest subtrees and smallest supertrees*. Algorithmica, 21:183-210, 1998.

[36] NIERMAN, JAGADISH: *Evaluating Structural Similarity in XML Documents*. University of Michigan, IIS-0002356.

[37] LEVENSHTEIN, V. I.: *Binary codes capable of correcting deletions, insertions, and reversals*. Doklady Akademii Nauk SSSR, 163(4):845-848, 1965 (Russian). English translation in Soviet Physics Doklady, 10(8):707-710, 1966.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 53-71

INTERPOLATION-BASED FUZZY REASONING IN BEHAVIOUR-BASED CONTROL STRUCTURES

SZILVESZTER KOVÁCS*
Department of Information Technology, University of Miskolc,
Miskolc-Egyetemváros, Miskolc, H-3515, Hungary
szkovacs@iit.uni-miskolc.hu

*Intelligent Integrated Systems Japanese Hungarian Laboratory,
Budapest University of Technology and Economics, Hungary

[Received November 2004 and accepted January 2005]

Abstract. Some difficulties emerging during the construction of fuzzy behaviour-based control structures are inherited from the type of the applied fuzzy reasoning. Classical fuzzy reasoning methods need a complete fuzzy rule base. In case of fetching fuzzy rules directly from expert knowledge e.g. for the behaviour coordination module, the way of building a complete rule base is not always straightforward. One simple solution for overcoming the necessity of the complete rule base is the application of interpolation-based fuzzy reasoning methods, since interpolation-based fuzzy reasoning methods can serve usable (interpolated) conclusion even if none of the existing rules is hit by the observation. These methods can save the expert from dealing with derivable rules and help to concentrate on cardinal actions only. For demonstrating the applicability of the interpolation-based fuzzy reasoning methods in behaviour-based control structures a simple interpolation-based fuzzy reasoning method and its adaptation for behaviour-based control will be discussed briefly in this paper.

Keywords: Interpolation-based Fuzzy reasoning, Behaviour-based Control

1. INTRODUCTION

In behaviour-based control systems (a good overview can be found in [3]), the actual behaviour of the system is formed as one of the existing behaviours (which fits best the actual situation), or as a kind of fusion of the known behaviours appeared to be the most appropriate to handle the actual situation. Beyond the construction of the behaviours, this structure has two other important tasks. The first is the decision, which behaviour is needed, or in case of behaviour fusion the determination of the necessity levels for each behaviour in solving the actual situation. The second is the way of the behaviour fusion. The first task, the behaviour coordination can be viewed as an actual system state approximation, where the actual system state is the set of the necessities of the known behaviours

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

needed for handling the actual situation. The second is the fusion of the known behaviours based on their necessities. In case of fuzzy behaviour based control structures both tasks are solved by fuzzy logic controllers. If the behaviours are also implemented on direct fuzzy logic controllers, the behaviours together with the behaviour fusion modules form a hierarchical fuzzy logic controller. Since the classical fuzzy reasoning methods (e.g. compositional rule of inference) demand complete rule base, all these rule bases have to be built taking care of filling all the possible rules. In case if there is some missing rule, there are observations may exist which hit no rule in the rule base and therefore no conclusion is obtained. Having no conclusion at any level of the fuzzy behaviour based control structure is hard to explain. E.g. one solution could be to keep the last real conclusion instead of the missing one, but applying historical data automatically to fill undeliberately missing rules could cause unpredictable side effects. Another solution for the same problem is the application of the interpolation-based fuzzy reasoning methods, where the derivable rules are deliberately missing. Since the rule base of a fuzzy interpolation-based controller, is not necessarily complete, it could contain the most significant fuzzy rules only without risking the chance of having no conclusion for some of the observations. In other words, during the construction of the fuzzy rule base, it is enough to concentrate on the cardinal actions; the "filling" rules (rules could be deduced from the others) can be deliberately omitted.

In the followings, first an approximate fuzzy reasoning method based on interpolation in the vague environment of the fuzzy rule base [4], [5], [6] will be reviewed. The main benefit of the proposed method is its simplicity, as it could be implemented to be simple and quick enough to be applied in practical direct fuzzy logic control too. Then its adaptation to behaviour-based control structures together with two simple examples will be discussed briefly.

2. INTERPOLATION-BASED FUZZY REASONING

One way of interpolative fuzzy reasoning is based on the concept of vague environment [2]. Applying the idea of the vague environment the linguistic terms of the fuzzy partitions can be described by scaling functions [2] and the fuzzy reasoning itself can be replaced by classical interpolation. The concept of vague environment is based on the similarity or indistinguishability of the elements. Two values in the vague environment are ε-distinguishable if their distance is grater than ε. The distances in vague environment are weighted distances. The weighting factor or function is called *scaling function (factor)* [2]. Two values in the vague environment X are ε-distinguishable if

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$\varepsilon > \delta_s(x_1, x_2) = \left| \int_{x_2}^{x_1} s(x)dx \right| \tag{1}$$

where $\delta_s(x_1, x_2)$ is the vague distance of the values x_1, x_2 and $s(x)$ is the scaling function on X. For finding connections between fuzzy sets and a vague environment the membership function $\mu_A(x)$ can be introduced as a level of similarity **a** to x, as the degree to which x is indistinguishable to **a** [2]. The α-cuts of the fuzzy set $\mu_A(x)$ are the sets which contain the elements those are $(1-\alpha)$-indistinguishable from **a** (see Fig.1):

$$\delta_s(\mathrm{a},\mathrm{b}) \le 1-\alpha \qquad \mu_A(x) = 1 - \min\{\delta_s(\mathrm{a},\mathrm{b}),1\} = 1 - \min\left\{\left|\int_{\mathrm{a}}^{\mathrm{b}} s(x)dx\right|,1\right\} \tag{2}$$

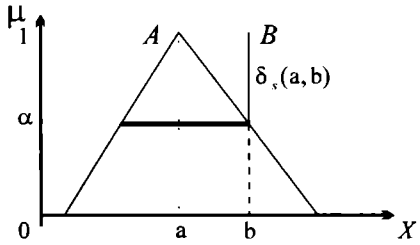


Figure 1: The α-cuts of $\mu_A(x)$ contains the elements that are $(1-\alpha)$-indistinguishable from a

This case (Fig.1) the vague distance of points a and b ($\delta_s(\mathrm{a},\mathrm{b})$) is the *Disconsistency Measure* (S_D) of the fuzzy sets A and B (where B is a singleton):

$$S_D = 1 - \sup_{x \in X} \mu_{A \cap B}(x) = \delta_s(\mathrm{a},\mathrm{b}) \text{ if } \delta_s(\mathrm{a},\mathrm{b}) \in [0,1], \tag{3}$$

where $A \cap B$ is the min t-norm, $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$ $\forall\, x \in X$.

From the viewpoint of fuzzy reasoning and fuzzy rule bases, where an observation fuzzy set is needed to be compared to rule antecedents built up member fuzzy sets (linguistic terms) of the antecedent fuzzy partitions (2) and (3) means that the disconsistency measures between member fuzzy sets of a fuzzy partition and a singleton, can be calculated as vague distances of points in the vague environment of the fuzzy partition. The main difference between the disconsistency measure and the vague distance is, that the vague distance is a value in the range of $[0,\infty]$, while the disconsistency measure is limited to $[0,1]$.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Therefore if it is possible to describe all the fuzzy partitions of the primary fuzzy sets (the antecedent and consequent universes) of the fuzzy rule base by vague environments, and the observation is a singleton, the "extended" disconsistency measures of the antecedent primary fuzzy sets of the rule base, and the "extended" disconsistency measures of the consequent primary fuzzy sets and the consequence can be calculated as vague distances of points in the antecedent and consequent vague environments.

The vague environment is described by its scaling function. For generating a vague environment of a fuzzy partition, an appropriate scaling function is needed to be find, which describes the shapes of all the terms in the fuzzy partition. A fuzzy partition can be characterised by a single vague environment if and only if the membership functions of the terms fulfil the following requirement [2]:

$$s(x) = |\mu'(x)| = \left|\frac{d\mu}{dx}\right| \quad \text{exists if} \quad \min\{\mu_i(x), \mu_j(x)\} > 0 \Rightarrow |\mu'_i(x)| = |\mu'_j(x)|, \quad \forall i, j \in I \tag{4}$$

where $s(x)$ is the vague environment.

Generally the above condition is not fulfilling, so the question is how to describe all fuzzy sets of the fuzzy partition with one "universal" scaling function. For this task the concept of *approximate scaling function*, as an approximation of the scaling functions which describe the terms of the fuzzy partition separately [4], [5], [6] is proposed. If the vague environment of a fuzzy partition (the scaling function or the approximate scaling function) exists, the member sets of the fuzzy partition can be characterised by points in the vague environment. (These points are characterising the cores of the fuzzy terms, while the membership functions are described by the scaling function itself.) If all the vague environments of the antecedent and consequent universes of the fuzzy rule base exist, all the primary fuzzy sets (linguistic terms) used in the fuzzy rule base can be characterised by points in their vague environment. Therefore the fuzzy rules (build on the primary fuzzy sets) can be characterised by points in the vague environment of the fuzzy rule base too. In this case the approximate fuzzy reasoning can be handled as a classical interpolation task. Applying the concept of vague environment (the distances of points are weighted distances), any interpolation, extrapolation or regression method can be adapted very simply for approximate fuzzy reasoning [4], [5], [6].

Because of its simple multidimensional applicability, for interpolation-based fuzzy reasoning in this paper the adaptation of the *Shepard operator* based interpolation (first introduced in [16]) is suggested. Beside the existing deep application oriented investigation of the Shepard operator e.g. [17], it was also successfully applied in the *Kóczy-Hirota fuzzy interpolation* [15]. (The stability and the approximation rate

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

of the Shepard operator based Kóczy-Hirota fuzzy interpolation is deeply studied in [7] and [8].) The Shepard interpolation method for arbitrarily placed bivariate data was introduced as follows [16]:

$$S_0(f,x,y)=\begin{cases} f_k & \text{if } (x,y)=(x_k,y_k) \text{ for some } k, \\ \left(\sum_{k=0}^{n} f(x_k,y_k)/d_k^{\lambda}\right)\Big/\left(\sum_{k=0}^{n} 1/d_k^{\lambda}\right) & \text{otherwise,} \end{cases} \tag{5}$$

where measurement points x_k, y_k ($k \in [0,n]$) are irregularly spaced on the domain of $f \in \Re^2 \to \Re$, $\lambda > 0$, and $d_k = \left[(x-x_k)^2+(y-y_k)^2\right]^{1/2}$. This function can be typically used when a surface model is required to interpolate scattered spatial measurements.

The adaptation of the Shepard interpolation method for interpolation-based fuzzy reasoning in the vague environment of the fuzzy rule base is straightforward by substituting the Euclidian distances d_k by scaled distances $\delta_{s,k}$

$$\delta_{s,k} = \delta_s(\mathbf{a}_k,\mathbf{x}) = \left[\sum_{i=1}^{m}\left(\int_{a_{k,i}}^{x_i} s_{X_i}(x_i)\,dx_i\right)^2\right]^{1/2} \tag{6}$$

where s_{X_i} is the i^{th} scaling function of the m dimensional antecedent universe, $\mathbf{x}$ is the m dimensional crisp observation and $\mathbf{a}_k$ are the cores of the m dimensional fuzzy rule antecedents A_k.

Thus in case of singleton rule consequents the fuzzy rules R_k has the following form:

$$\textbf{If } x_1 = A_{k,1} \textbf{ And } x_2 = A_{k,2} \textbf{ And } \quad \textbf{And } x_m = A_{k,m} \textbf{ Then } y = c_k \tag{7}$$

by substituting (6) to (5) the conclusion of the interpolative fuzzy reasoning can be obtained as:

$$y(\mathbf{x})=\begin{cases} c_k & \text{if } \mathbf{x}=\mathbf{a}_k \text{ for some } k, \\ \left(\sum_{k=1}^{r} c_k/\delta_{s,k}^{\lambda}\right)\Big/\left(\sum_{k=1}^{r} 1/\delta_{s,k}^{\lambda}\right) & \text{otherwise.} \end{cases} \tag{8}$$

The interpolative fuzzy reasoning (8) can simply extend to be able to handle fuzzy conclusions by introducing the vague environment (scaling function) of the consequence universe. This case the fuzzy rules R_k has the following form:

$$\textbf{If } x_1 = A_{k,1} \textbf{ And } x_2 = A_{k,2} \textbf{ And } \quad \textbf{And } x_m = A_{k,m} \textbf{ Then } y = B_k. \tag{9}$$

By introducing vague distances on the consequence universe:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$\delta_s(b_k, y) = \left[\left(\int_{b_1}^{y} s_Y(y)dy \right)^2 \right]^{1/2} \tag{10}$$

where s_Y is the i^{th} scaling function of the one dimensional consequent universe, b_k are the cores of the one dimensional fuzzy rule consequents B_k.

Introducing the first element of the one dimensional consequence universe b_0 the ($Y: b_0 < y \ \forall \ y \in Y$), based on (8) and (10) the requested one dimensional conclusion $y(\mathbf{x})$ can be obtained from the following formula:

$$\delta_s(y(\mathbf{x}), b_0) = \begin{cases} \delta_s(b_k, b_0) & \text{if } \mathbf{x} = \mathbf{a}_k \text{ for some } k, \\ \left(\sum_{k=1}^{r} \delta_s(b_k, b_0) / \delta_{s,k}^{\lambda} \right) \Big/ \left(\sum_{k=1}^{r} 1/\delta_{s,i}^{\lambda} \right) & \text{otherwise.} \end{cases} \tag{11}$$

A simple one-dimensional example for the approximate scaling function and the Shepard operator based interpolation (11) is introduced on Fig. 2 and on Fig. 3.

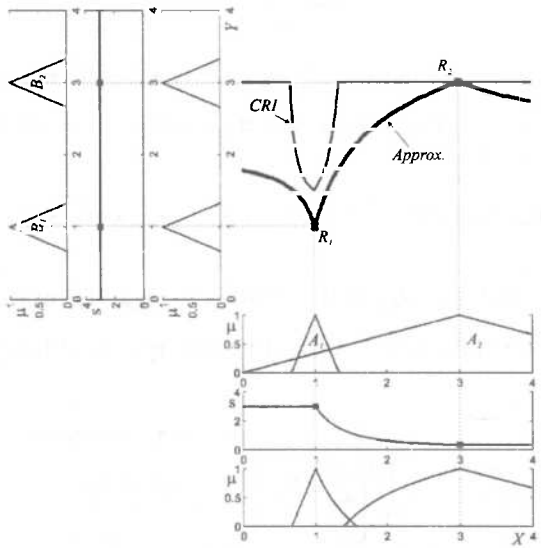


Figure 2: Interpolation of two fuzzy rules (R_i: $A_i \rightarrow B_i$) (see fig. 3 for notation)

For comparing the crisp conclusions of the interpolation-based fuzzy reasoning and the classical methods, the conclusions generated by the max-min compositional rule of inference (CRI) and the centre of gravity defuzzification for the same rule base is also demonstrated on the example figures (Fig. 2, Fig. 3). More detailed

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

description of the proposed approximate fuzzy reasoning method can be found in [4], [5], [6].

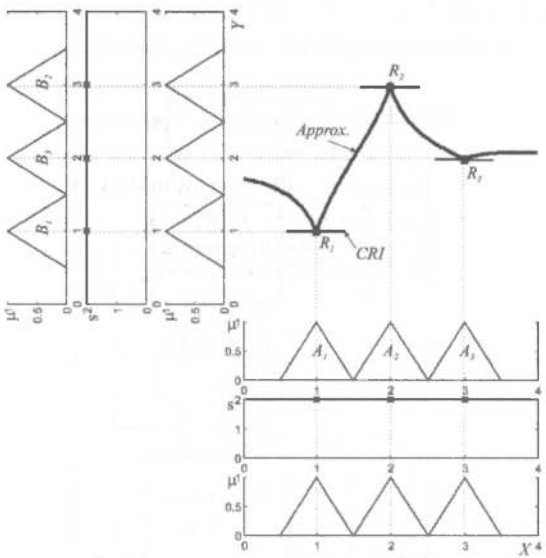


Figure 3: Interpolation of three fuzzy rules (R_i: $A_i \rightarrow B_i$) in the approximated vague environment of the fuzzy rule base. Using the Shepard operator based interpolation (p=1) (*Approx.*), and the min-max CRI with the centre of gravity defuzzification (CRI). Where μ is the membership grade and s is the scaling function.

3. THE APPLIED FUZZY BEHAVIOUR-BASED STRUCTURE

The main benefit of the interpolation-based fuzzy reasoning method, discussed in the previous chapter, is its simplicity. Applying look-up tables for pre-calculating the vague distances, it could be implemented to be simple and quick enough to fit the speed requirements of practical real-time direct fuzzy logic control systems, e.g. the requirements of fuzzy behaviour-based control too. The calculation efforts of many other interpolation-based fuzzy reasoning methods "wasted" for determining the exact membership shape of the interpolated fuzzy conclusion prohibits their practical application in real-time direct fuzzy logic control. The lack of the fuzziness in the conclusion is a disadvantage of the proposed method, but it has no influence in common applications where the next step after the fuzzy reasoning is the defuzzification.

In the followings a pure fuzzy behaviour-based control structure and the adaptation of the proposed interpolation-based fuzzy reasoning method will be discussed more detailed.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In case of pure fuzzy behaviour-based control structures all the main tasks of the behaviour-based control – the behaviour coordination, the behaviour fusion, and the behaviours themselves – are implemented on fuzzy logic controllers. (Such a structure is introduced on Fig.4.) Any of these controllers can apply the proposed interpolation-based approximate fuzzy reasoning method.

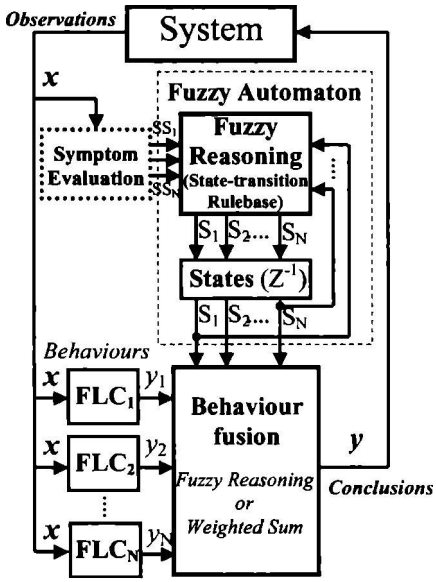


Figure 4: The suggested fuzzy behaviour-based control structure

For demonstrating the main benefits of the interpolation-based fuzzy reasoning in behaviour-based control, this paper concentrates on the many cases most heuristic part of the structure, on the behaviour coordination.

The task of behaviour coordination is to determine the necessities of the known behaviours needed for handling the actual situation. In the suggested behaviour-based control structure, for this task the finite state fuzzy automaton is adapted (Fig.4) [9]. This solution is based on the heuristic, that the necessities of the known behaviours for handling a given situation can be approximated by their suitability. Moreover the suitability of a given behaviour in an actual situation can be approximated by the similarity of the situation and the prerequisites of the behaviour. (Where the prerequisites of the behaviour is the description of the situations where the behaviour is valid (suitable itself)). In this case instead of determining the necessities of the known behaviours, the similarities of the actual situation to the prerequisites of all the known behaviours can be approximated.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Thus the first step of this kind of behaviour coordination is determining the similarities of the actual situation to the prerequisites of all the known behaviours – applying the terminology of fault classification; it is the symptom evaluation (see e.g. Fig.4). The task of symptom evaluation is basically a series of similarity checking between an actual symptom (observations of the actual situation) and a series of known symptoms (the prerequisites – symptom patterns – of the known behaviours). These symptom patterns are characterising the systems states where the corresponding behaviours are valid. Based on these patterns, the evaluation of the actual symptom is done by calculating the similarity values of the actual symptom (representing the actual situation) to all the known symptoms patterns (the prerequisites of the known behaviours). There exist many methods for fuzzy logic symptom evaluation. For example fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [1] can be adopted, where the known symptoms patterns are the cluster centres, and the similarities of the actual symptom to them can be fetched from the fuzzy partition matrix. On the other hand, having a simple situation, the fuzzy logic symptom evaluation could be a fuzzy rule based reasoning system itself.

One of the main difficulties of the system state approximation in behaviour coordination is the fact that in most of the cases the symptoms of the prerequisites of the known behaviours are strongly dependent on the actual behaviour of the system. Each behaviour has its own symptom structure. In other words for the proper system state approximation, the approximated system state is also needed. A very simple way of solving this difficulty is the adaptation of fuzzy automaton. In this case the state vector of the automaton is the approximated system state, and the state-transitions are driven by fuzzy reasoning (Fuzzy state-transition rule base on Fig.4), as a decision based on the previous actual state (the previous iteration step of the approximation) and the results of the symptom evaluation.

4. APPLICATION EXAMPLES

For demonstrating the simplicity of defining the rule base for interpolation-based fuzzy reasoning, as the first example, the state-transition rule base of the previously studied fuzzy automaton style behaviour coordination module applied for user adaptive information retrieval system in [10] and [11] will be discussed briefly in the followings. In this user adaptive information retrieval system example (introduced in [10] and [11] in more details) the user adaptivity is handled by combination of existing (off-line collected) human opinions (user models) in the function of their approximated similarity to the actual user opinions. As an analogy to the behaviour-based control applications, the different behaviours are the different existing user models, and the actual situation is the similarity of the actual

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

user to the evaluators, originally gave the existing user models. Based on the observations (inputs) the conclusion of the user feedback (the symptom evaluation about the state-transition to state i, SS_i for all the possible states $\forall i \in [1,N]$) and the previous state values S_i – the new state values (i.e. the vector of the suitability of the existing user models) are needed to somehow be estimated. The suggested heuristic in this example is very simple. If a suitable model (S_i) is already found and the user feedback is still supporting it (SS_i), it is needed to be kept even if the user feedback began to support some other models too. If there were no suitable model, but the user feedback began to support one, it has to be picked it at once. In case of interpolation-based fuzzy reasoning, the above heuristic can be simply implemented by the following state-transition rule base [10], [11]. For the i^{th} state variable S_i, $i \in [1,N]$ of the state vector **S**:

If S_i=One **And** SS_i=One **Then** S_i=One (12.1)

If S_i=Zero **And** SS_i=Zero **Then** S_i=Zero (12.2)

If S_i=One **And** SS_i=Zero
And SS_k=Zero **Then** S_i=One $\forall k \in [1,N], k \neq i$ (12.3)

If S_i=Zero **And** SS_i=One
And S_k=Zero **And** SS_k=Zero **Then** S_i=One $\forall k \in [1,N], k \neq i$ (12.4)

If S_i=Zero **And** SS_i=One
And S_k=One **And** SS_k=One **Then** S_i=Zero $\exists k \in [1,N], k \neq i$ (12.5)

where SS_i is the conclusion of the symptom evaluation about the state-transition to state i, $\forall i \in [1,N]$; N is the number of known behaviours (state variables). The structure of the state-transition rules is similar for all the state variables. `Zero` and `One` are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of the `Zero` and `One` fuzzy sets can be different in each S_i, SS_i universes.

Please note that rule base (12) is sparse. It contains the main rules for the following straightforward goals only: Rule (12.1) simply keeps the previously chosen state values in the case if the symptom evaluation also agrees. The rule (12.2) has the opposite meaning, if the state values were not chosen, and moreover the symptom evaluation also disagrees, then the state value should be suppressed. The rule (12.3) keeps the already selected state values (previous approximation), even if the symptom evaluation disagrees, if it has no better "idea" Rules (12.4) and (12.5) have the task of ensuring the relatively quick convergence of the system to the sometimes unstable (changeable) situations, as new state variables which seem to

Digitalizálta: Miskolci Egyetem Könyvtár, Levéltár, Múzeum

be fit, can be chosen in one step, if there is no previously chosen state, which is still accepted by the symptom evaluation (12.4). (Rule (12.5) has the task to suppress this selection in the case if exists a still acceptable state, which has been already chosen.) The goal of this heuristic is to gain a relatively quick convergence for the system to fit the opinions of the actual user, if there is no state value high enough to be previously accepted. This quick convergence could be very important in many application areas e.g. in case of an on-line user adaptive selection system introduced in [10], where the user feed-back information needed for the state changes are very limited.

Some state changes of the fuzzy automaton in the function of the conclusion of the symptom evaluation (SS_1, SS_2) for the two states case (applying the state-transition rule base (12)) are visualised on Fig.5 and Fig.6.

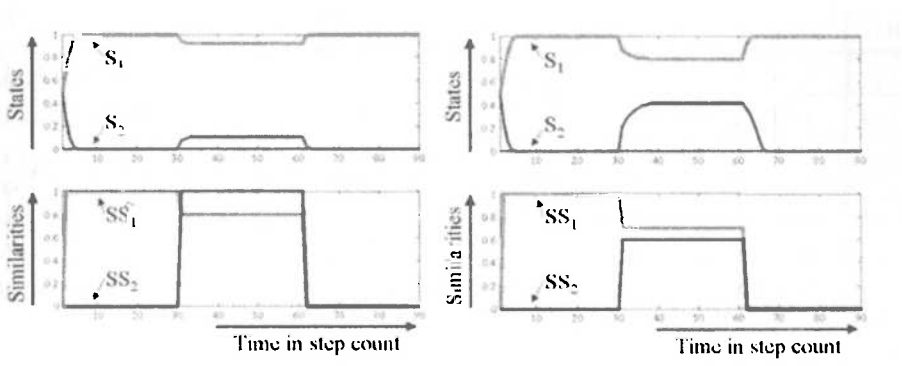


Figure 5: Do not "pick up" a new state if the previous approximation is still adequate

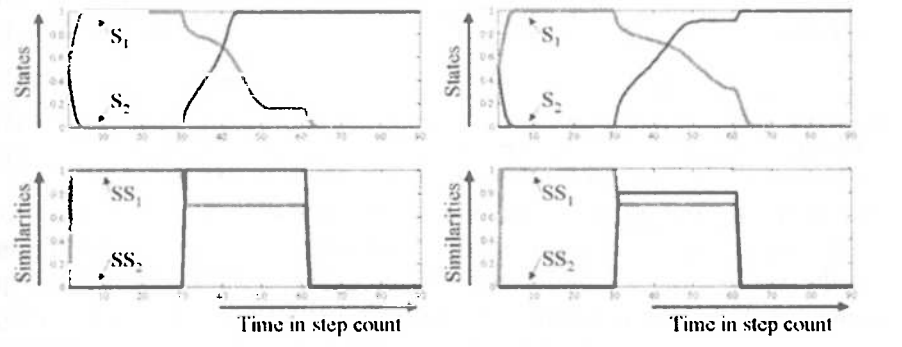


Figure 6: But "pick it up" if it seems better, or at least as good as the previous was

Counting the rules of the classical (e.g. compositional) fuzzy reasoning for the same strategy, in the two state case the complete rule base needs 16 rules (four

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

observation universes (S_1, SS_1, S_2, SS_2) each with two terms fuzzy partitions (Zero, One) - 2^4 rules), while the sparse rule base (12) contains 5 rules only (see table 1 for the state-transition rule base of state S_1). Taking into account that in the proposed behaviour-based control structure a separate rule base is needed for each state variable, the behaviour coordination needs 32 rules, while 10 is enough in case of applying the proposed interpolation-based fuzzy reasoning method. Increasing the number of the state variables, the situation becomes even worse. In case of three state variables (S_1, S_2, S_3) the rate become $3 \cdot 2^6$ ($n \cdot 2^{2 \cdot n}$, where n is the number of the states) and $3 \cdot 6$ ($n \cdot (n+3)$) up to the interpolation-based method (see table 2).

Table 1: State-transition rule base of state S_1 in case of two state variables (S_1,S_2) according to rule base (12)

R_{S1}:	S_1	SS_1	S_2	SS_2	S_1	
1.,	One	One			One	according to (12.1)
2.,	Zero	Zero			Zero	according to (12.2)
3.,	One	Zero		Zero	One	according to (12.3)
4.,	Zero	One	Zero	Zero	One	according to (12.4)
5.,	Zero	One	One	One	Zero	according to (12.5)

Table 2: State-transition rule base of state S_1 in case of three state variables (S_1,S_2,S_3) according to rule base (12)

R_{S1}:	S_1	SS_1	S_2	SS_2	S_3	SS_3	S_1	
1.,	One	One					One	see (12.1)
2.,	Zero	Zero					Zero	see (12.2)
3.,	One	Zero		Zero		Zero	One	see (12.3)
4.,	Zero	One	Zero	Zero	Zero	Zero	One	see (12.4)
5.,	Zero	One	One	One			Zero	see (12.5)
6.,	Zero	One			One	One	Zero	see (12.5)

The exponential rule number "explosion" in case of increasing the number of the input variables makes many heuristic ideas unimplementable and therefore useless. E.g. in the case of the original source of the example application of this paper (introduced in [10]), the behaviour coordination module applied for user adaptive information retrieval system had 4 state variables (one for each emotional model), which makes this simple rule base (12) practically unimplementable as a complete rule base ($4 \cdot 2^8 = 1024$ rules). While the working demonstrational example (which

Digitalizálta: Miskolci Egyetem Könyvtár, Levéltár, Múzeum

can be downloaded from [18]) had only 28 rules thanks to the applied interpolation-based fuzzy reasoning method.

4.1. Vehicle navigation control example

For another example of the interpolation-based fuzzy rule base definition simplicity, in the followings, the behaviour coordination module of an automated guided vehicle (AGV) steering control [12], [13] will be discussed briefly.

In this example application the steering control has two main goals, the path tracking (to follow a guide path) and the collision avoidance. The simulated AGV is first trying to follow a guide path, and in the case if it is impossible (because of the obstacles), it leaves it, and as the collision situation is avoided, it tries to find the guide path and follow it again.
The AGV has two simulated sensor systems. The path sensing system senses the position of the guide path by special sensors (guide zone) tuned for the guide path. The obstacles are sensed directly by three ultrasonic distance sensors (on the front of the AGV, one in the middle (U_M) and one-one on both sides (U_L, U_R) (see Fig.7) and the obstacle boundaries are approximated based on dead reckoning and previous obstacle distances [12]. The global goal of the path tracking strategy is to follow the guide path by the guide zone with minimal path tracking error on the whole path (see Fig.7).

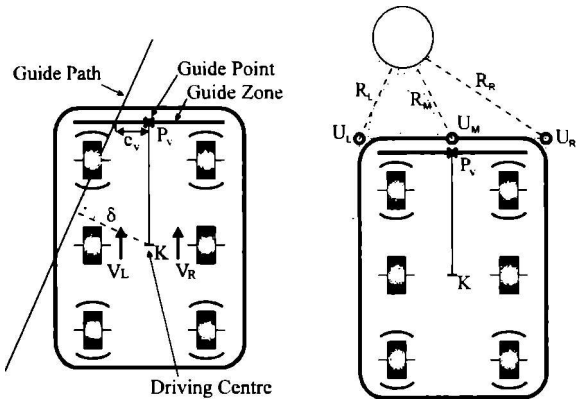


Figure 7: Differential steered AGV with guide zone, δ is the path tracking error, e_v is the distance of the guide path and the guide point P_v, K is the driving centre, R_L, R_R, R_M are the distances measured by the left, right and middle ultrasonic sensors (U_L, U_R, U_M).

Because of the requirement of being able to find the guide path after leaving it, the complete path tracking and the collision avoidance strategy needs four component behaviours:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Path tracking and restricted collision avoidance strategy: The main goal of this strategy is the path tracking (to follow a guide path) and as a sub goal, a kind of restricted (limited) collision avoidance [13]. (Here the restricted collision avoidance means, "avoiding obstacles without risking the chance of loosing the guide path".)

The collision avoidance strategy: The second known behaviour is a simple collision avoidance steering strategy. Its only goal is to avoid collisions.

The collision avoidance with left/right tendency strategy: The next two behaviours are basically the same as the collision avoidance steering strategy, expect the left or right turning tendencies in case of no left or right turning difficulties. These strategies are needed to help finding the path after leaving it (because of the fail of the first strategy).

In this vehicle navigation control example (introduced in [13] in more details) the studied behaviour coordination module has the task of determining the necessities of the four component behaviours. Having four known behaviours, the automaton has four state variables (see Fig.4).

These are the necessity of the path tracking and restricted collision avoidance strategy (S_P), the necessity of the collision avoidance strategy (S_C), and the necessities of the collision avoidance strategies with right tendency (S_{CR}), and left tendency (S_{CL}) in solving the actual situation.

Having four necessities (four conclusions), four state transition rule bases are required. The $\mathbf{R_{SP}}$ state transition rule base is determining the next value of the S_P state variable, $\mathbf{R_{SC}}$ is for determining S_C, and the $\mathbf{R_{SCR}}$ and $\mathbf{R_{SCL}}$ are determining the next values of S_{CR} and S_{CL}. The available observations [13] of the state transition rule bases are the distance between the guide path and the driving centre (e_v), the distances measured by the left middle and right ultrasonic sensors (R_L, R_M, R_R), the approximated maximal left and right turning angle without side collision (α_{ML}, α_{MR}), the availability of the path sensing (P_V), and the state variables themselves (S_P,S_C,S_{CR},S_{CL}).

Based on heuristic considerations and simulated experiments the four state-transition rule bases became the following ones:

$\mathbf{R_{SP}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P	
				Z	V			L			L	(13.1)
				PL	V					S	Z	(13.2)
				NL	V				S		Z	(13.3)
					NV						Z	(13.4)

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$\mathbf{R_{SC}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_C	
					V			S			L	(14.1)
					V			L			Z	(14.2)
					NV						Z	(14.3)

$\mathbf{R_{SCR}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_{CR}	
L				NVL	V						L	(15.1)
		L			NV						L	(15.2)
				Z	V			L			Z	(15.3)
			L								Z	(15.4)

$\mathbf{R_{SCL}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_{CL}	
L				PVL	V						L	(16.1)
			L		NV						L	(16.2)
				Z	V			L			Z	(16.3)
		L									Z	(16.4)

where the linguistic labels of fuzzy sets (linguistic terms) stand for N: negative, P: positive, VL: very large, L: large, S: small, Z: zero, V: path valid, NV: path not valid.

The heuristic considerations laying behind the state-transition rule base of the path tracking and restricted collision avoidance strategy ($\mathbf{R_{SP}}$) are quite straightforward: Rule (13.1) simply takes the path tracking strategy in case if there is a valid path with no path tracking error (e_V=Z) and there is no collision situation. The rest of the rules are suppressing the path tracking strategy in case of collision situation (13.2)-(13.3), or if the path sensing is not available (13.4).

The state-transition rule base of the collision avoidance strategy ($\mathbf{R_{SC}}$) is also straightforward: Rule (14.1) calls the collision avoidance strategy in case of valid path sensing and collision situation. Rule (14.2) suppresses the collision avoidance strategy if there is no collision situation (the distance of the obstacle and the middle sensor is large) and the path sensing is valid. Rule (14.3) suppresses the collision avoidance strategy if the path sensing is not valid (PV=NV), as these situations are handled by the collision avoidance strategies with left and right tendencies.

The remaining two state-transition rule bases are serving the requirement of being able to find the guide path after leaving it. They are symmetric in the sense of the left and right directions. The right turning tendency is called if the vehicle leaves

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

the guide path on the left side (see rule (15.1)) and left turning tendency is called if the vehicle left on the right (rule (16.1)). Rules (15.2) and (16.2) have the task to keep the already selected right or left direction tendency if the path sensing is still not available (PV=NV). Rules (15.3) and (16.3) are suppressing the strategies serving the free run if the guide path is found again (valid path PV=V with no path tracking error e_v=Z). Rules (15.3) and (16.3) are serving of the mutual exclusion of the two contradictive (left or right turning tendencies) strategies.

Figure 8 introduces some results of the simulated AGV steering application.

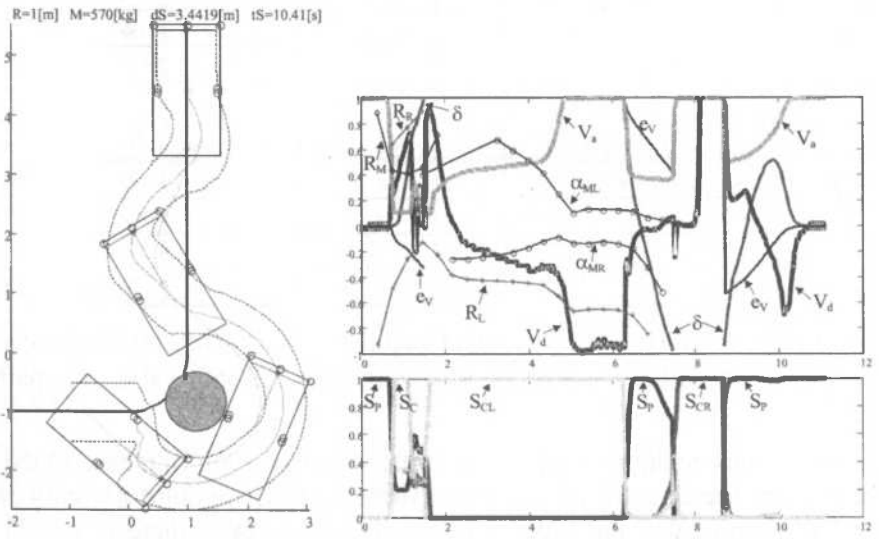


Figure 8: Track of a single run in case of one obstacle and the time function of observations, conclusions and system state values (S_P, S_C, S_{CL}, S_{CR}).

A downloadable and runable code of the application examples and the code of the interpolation-based fuzzy reasoning method studied in this paper can be found at [18].

5. Conclusions

The goal of this paper was to review an interpolation-based fuzzy reasoning method, which could be implemented to be simple and quick enough to fit the requirements of behaviour-based control structures in real-time direct fuzzy logic control systems. The suggested approximate fuzzy reasoning method based on interpolation in the vague environment of the fuzzy rule base gives an efficient way for designing direct fuzzy logic control applications. The lack of the fuzziness in the conclusion is a disadvantage of the proposed method, but it has no influence in

Digitalizálta: Miskolci Egyetem Könyvtár, Levéltár, Múzeum

Production Systems and Information Engineering
Volume 2 (2004), pp. 73-89

SYNERGIC USE OF SOFTWARE QUALITY MODELS

KATALIN BALLA
Technical University Budapest, Dept. of Control Engineering and Information Technology
e-mail: balla@iit.bme.hu

[Received November 2004 and accepted January 2005]

Abstract. The article supports the idea that nowadays using more software quality models in a synergic way and customizing them to fit the specific needs of an organization is the only viable option for doing efficient process improvement in software companies.

The argumentation is done by presenting a theoretical framework in which the quality models can be placed. Next, we describe our personal experience with using this framework in a Hungarian software company, presenting the main results of a 11 years-long case study[1] The article emphasizes the business-driven SPI project of the company, started to enrich the ISO 9001:2000-conform quality system, and the way of consiously using more quality models to do this. In the end of the article we shortly present the huge organizational change the company has undergone, and its consequences on the previously started software process improvement program.

Keywords: software quality management, ISO 9001, CMM, software process improvement (SPI), project management (PM)

1. INTRODUCTION

In the intense international competition software companies are more and more forced to think about proving their capability of delivering good products. One way of having such a proof is to obtain an official certificate about usage of a certain standard or model. However, introducing a quality approach based on a standard or model, and institutionalising it, so that the organisation is able to pass an audit, requires a lot of investment from the software companies, both in terms of money and effort – which a company would not like to waste. Therefore, really business-driven software companies will be willing to do only *really efficient* software process improvement.

[1] The author worked between 1993-2004 as a quality manager at the company where the case study was run. Thanks are due to IQSOFT management for making possible to do research and record the steps of process improvement.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Efficient improvement programs are always based on real needs of companies and will always start from understanding the actual situation of that company. Choosing the right approach, model or standard for the improvement program would be the next step.

The difficult question is: which model to choose to best fit the company's needs in improving software quality? What activities to execute and in which sequence in order to transform the initial – probably almost chaotic – situation, step by step, into some controllable and provable "order"?

We faced the described problem while doing quality management in a Hungarian software company. As a result of the research and practical work done over 11 years, a theoretical framework has been worked out and used. Our experience has shown that the framework is well usable in a "real" environment. Therefore, we consider it worth to present in the following.

1.1. The quality framework

1.1.1. QMIM elements

For answering the questions about a "good approach of software quality", we have to understand what software quality means for the companies, in each particular situation. Quality of software is a very complex subject, and, as such, it is extremely hard to define. If we wish to deal with software quality in its complexity, we have to think about the software *product*s, the *process*es that produce the products and the *resource*s that execute the processes. We have to *define* these objects, to choose the right *quality attributes* for them and verify their actual value by the means of objective *metrics*. In conclusion, a framework capturing the important elements of software quality can be defined (see Figure 1). The framework has been named QMIM (Quality through Managed Improvement and Measurement). It is presented in detail in [1] and [2]. Here we describe its most important element and features.

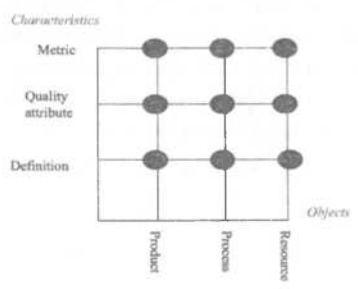


Figure 1: QMIM: the quality framework

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Basic idea is that if we wish to deal with software quality in its complexity, *we have to completely fill the QMIM framework.* This means that the software product, the processes producing it and the resources executing the processes have to be equally well understood, their quality attributes need to be defined (we have to understand when do we say that an object "is good"), and the actual value of the attributes need to be measured by objective metrics. Normally, such a detailed quality approach is not possible to follow: it would mean a huge investment to the company both in terms of effort and time, which might not be in line with the actual business needs. However, a long-term vision about quality within a software company requires the understanding of the QMIM elements and the capability to *unambiguously identify them in the actual environment.* With other words: a software company needs to know what its products are, what their characteristics mean and how they can be measured, has to be able to define its processes and ensure the right resource management. Efficient quality management can be done having QMIM elements always in mind and *consciously choosing* the most appropriate ones in each particular situation. This means that *certain elements* of the QMIM framework will be dealt with in more detail at a certain moment, but the company will always be aware of the fact that *all* elements need to be addressed sooner or later.

1.1.2. QMIM static aspects

The processes of a software company can be grouped into project management processes (common for all development projects) and technical processes (bearing the particularities of each development, depending e.g. on technology, methodology etc.). If we agree that resources are managed via project management processes, thus including resources into project management, we have the following objects of software production: project management processes, technical processes and products.

The principles of QMIM help in structuring the quality-related data of a company: the objects, characteristics and metrics are understood and their relationships can be represented. The static aspect of the QMIM framework (see [1]) describes a possible database for storing the data related to quality. See Figure 2 for the structure of the database.

As seen from the figure, the products can be grouped into software systems, that contain items. The items are developed by technical processes that follow a development methodology, described in guidelines. Technical processes are executed within projects. The projects are managed in a way that is described in a PM methodology. The projects use resources. All important elements - software item, technical process, project management process and resource – have associated quality attributes that can be measured by metrics.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The database can be used for storing both generic information (that describes the basic objects and their relationships) and the data of the concrete projects.

To be noted that such a database will not need to be built from scratch, as (almost) all companies have some data-collections that can be incorporated. If the understanding of the structure and relationships is there, the database will always be possible to complete with new elements.

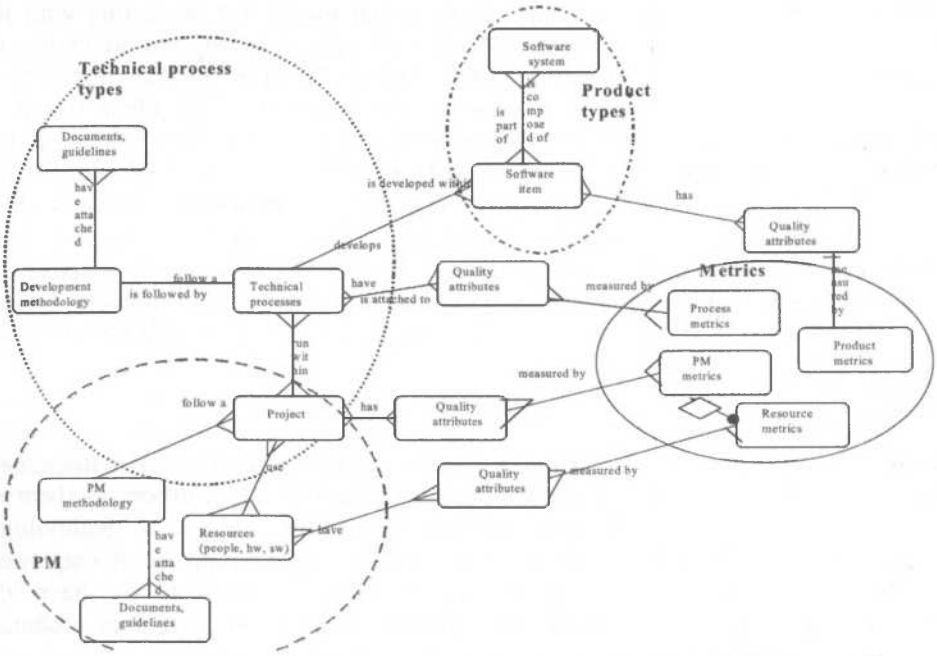


Figure 2: Database structure suggested by QMIM

1.1.3. QMIM guidelines

If a company has the right understating about the quality of its software, it can direct the company-wide processes and process improvement in a way that will permit a step-wise completion of the QMIM framework. There is no strict recommendation about the sequence of "filling" the framework: it can be done both "horizontally" and "vertically" The horizontal approach means that the company starts by obtaining an equally deep understanding of all *objects* of software production, focusing its effort to quality objectives and metrics afterwards. The vertical approach means that the company chooses one particular object, and defines its quality attributes and metrics before starting to deal with the next object.

The two approaches might be mixed: a company can at one time define more elements to different degrees of detail. The choice has to be made based on the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

understanding of the actual situation and the importance each element presents to the company.

Our experience shows that some elements are easier to understand than others, e.g. processes are easier understood and measured than products. Experience shows also that project management processes are the easiest to define and measure. Technical processes can follow, and product characteristics and measurement is the element hardest to approach.

QMIM guidelines describe these possibilities in detail (see [1] and [2]).

It would be extremely difficult for software companies to define quality approaches from scratch. At the same time, this would be a useless waste of effort, since many approaches, standards, methods, models have been worked out by the software technology - and quality community. QMIM emphasizes this aspect, and *suggests the usage of the most appropriate method, standard or approach in each case*, giving aid in understanding of what "the most appropriate" means.

The approaches, standards and models in software industry are extremely various in their approach used. Here we present some of the most popular ones.

The early Boehm ([3]) and McCall ([4]) quality models concentrate on software product quality, and so does standard ISO 9126 family ([5]), which gives important guidance in defining and measuring software product characteristics.

The nowadays widely used standard, ISO 9001:2000 ([6]), focuses mainly on the processes. It addresses product quality and measurement also, but without giving guidance for these elements.

Managerial and technical processes are addressed by the popular CMM model ([7]) that addresses product characteristics too (but mainly from a process/organisational aspect), and the Bootstrap methodology ([8]) worked out to asses organisation maturity. The SPICE model / ISO 15504 standard ([9]) is also process oriented, as well as CMMI ([10]), developed to integrate (among others) staged approach of CMM and continuous approach of SPICE. PSP (see ([12]) and TSP (see [13]) are completing the CMM(I) approach, concentrating on individual and team aspects of software development. Project management aspects are addressed by many PM methodologies (e.g. PRINCE [14]). Human resource characteristics are dealt with in e.g. Weinberg's theory ([15]) and People- CMM ([16])

Metrics and measurement methods are addressed by e.g. [11] and Basili's GQM paradigm ([17]). Function point counting methods (see [18], [19], [20]) are dedicated to understand software product size and complexity.

Some of these approaches are widely known, while others are used only by a restricted number of companies. The choice a company makes in terms of software quality model/approach used depends on many factors. For instance, ISO

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

9001:2000 is nowadays a condition of staying in the market. To this (rather general, therefore not easily applicable for software) approach the companies normally add a model or standard their customers prefer. USA was preferring CMM, Europe used SPICE more, but CMMI will probably solve this problem.

QMIM framework does not impose the usage of one specific model, but is suggests the conscious choice of a model, while understanding how that approach, standard or model is related to the important elements of software production.

However, since we consider that the notions of capability, maturity and CMM(I) levels are widely known and accepted by the software development community, we suggest to make a choice based on the company's actual maturity (in terms of CMM[2]). On the lowest maturity level it seems best to approach software quality by project management process definition, characteristics and measurement. This will probably bring the company to maturity level 2, where technical processes can be defined, the result being a level 3 company maturity. At this level product characteristics can be understood, defined and measured.

QMIM strongly emphasizes the need for measurement already on the lowest maturity level. This aspect is in line with the CMMI – structure, that brought the process of measurement and analysis down to maturity level 2 (while in CMM it appeared explicitly only on level 4).

1.2. Using more models in a synergic way

Studying in detail the models presented shortly in this paper, one will remark that *no approach, model or standard covers all the important aspects of software quality* (although new versions of earlier models are definitely more broad in their scope, in the number of objects they are dealing with). We can state that companies will have to choose the right approach based on their business needs. Understanding the *business needs* in a right way is a rather complex job that claims solid professionalism both in the field of software development's nature and existing quality models and standards. Choosing a wrong approach could do considerable harm to a company, by misleading the efforts from the really important objectives.

One way of avoiding the trap of a badly chosen quality model or approach is to quit exclusively relying on *one* certain quality model in favor of choosing among several approaches, consciously *using more approaches in a synergic way*, according to the specific business needs of a certain organisation.

[2] The maturity levels of an organization, in CMM terms, are: level 1: chaotic, level 2: repeatable, level 3: defined, level 4: managed, level 5: optimizing.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In the following part of this article we describe our experience in using QMIM and point out how its concepts helped in using more quality models in a synergic way.

2. THE CASE STUDY

In this chapter we describe the main results of a 11 years long case study done at a Hungarian software company. We make the presentation having QMIM in mind, and showing at every phase how its concepts were used.

The case study took place at IQSOFT Ltd. / later IQSYS Ltd., one of the main representatives of the software industry in Hungary. The initial company was formed in early 1990 from part of a large state organisation, the Theoretical Laboratory within the Computer Technology Co-ordination Institute (SZKI). The company was medium-sized, having around 100 employees. Three main software activity types could have been defined: software development (mainly in a database environment, using 4GL development tools), software integration, and software implementation. The projects were generally small to medium sized and could differ widely in their characteristics. Research and training were also important activities in IQSOFT's activity profile.

In the following sections we will describe the phases of the case study in more detail.

2.1. The enthusiastic start and seeking new ways

Since 1993, efforts have been made at IQSOFT to develop and introduce an ISO 9001-comform internal quality management system (QMS).

As a preparatory step for building the QMS and to be aware of the good practice existing, in 1994 the overall company and two concrete projects were assessed according to the Bootstrap methodology (see Table 1). Due to an insufficient understanding of the interconnections between CMM and ISO 9001, the company was unable to use the Bootstrap assessment results in an appropriate way.

Experiencing the "failure" of the quality-exercise, the management and the employees became skeptic about the possibility to improve daily practices. However, we came to the idea that a software QMS will not be really operational and useful if we would take into account the ISO 9001-prescriptions only. Research done in parallel confirmed our ideas. We learned about Fenton's basic entities ([11]): products, processes and resources. Before these entities would be understood, precise definitions would be needed (which seldom existed). Next to that we began to understand the relationships with the several software quality attributes. We placed all these elements into a matrix, representing in fact our "chosen quality framework" – a part of the later QMIM (see Figure 1).

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

We remarked the existence of many other approaches on software quality. Process oriented approaches completing ISO 9001, like CMM, Bootstrap, and SPICE have been studied. We learned about ISO 9126 and approaching software quality by the several product characteristics.

In 1995 IQSOFT won an EU PHARE tender "Technology Development and Quality Management", thus gaining financial support for quality oriented activities. That was the beginning of the so called IQPM^2 project, which used PM^2 methodology of Lucas Management Systems in developing a customized PM system.

The project started in February 1996 and finished in May 1997, and we can state that it was successful: it reached its goal within the planned time and budget limits. Besides the planned ones (developing and introducing a PM system based on the needs of the company) the IQPM^2 project produced a series of side results, which, from the company's long-term perspective, were even more important than the planned ones.

We understood that the processes of a software company could be - in our case: should be - divided into at least two distinct types: *project management processes* and *technical processes*. Projects can be modeled according to both activity types, so project management models and project type models can be situational configured. We noticed that project management activities in IQSOFT were more stable than the technical ones, which justified again their separation. With the standardisation of project management activities –building the (*unique*!) project management model of the company we made the first important step towards bringing order in the company. At that moment we *consciously left the technical activities undefined.* Following this argumentation, *we regrouped Fenton's entities into the following objects: Project Management (PM), Technical Process (TP)*, and *Product. (P)*. Basic reason for this regrouping was the fact that the objects we were talking about were not Fenton's "narrow" objects, but were more business objects. Fenton's "resources" were incorporated into project management, because we considered that all resource-related subjects were addressed within the project management issues. This way, QMIM framework was refined further.

According to a Bootstrap assessment carried out in 1997, the overall organisation had the maturity level 2, while the pilot projects reached 2.50 in CMM. See the results related to some process areas in Table 1.

With the positive experiences of PM^2, IQSOFT's management decided to go again for ISO 9001 registration. On this basis a project was started to obtain registration. It can be called a "new approach" project because it had the scope of obtaining ISO 9001 certification using all former experience of IQSOFT in building a *customised QMS*. The project was declared a top-priority one, having an internal effort of 250 man-days.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Table 1: Results of Bootstrap assessments

Area	1994	1997
SPU	2.25	3
Process description	2	3.25
Process control	1.5	3
Project management	2.75	3.5
Development model	2.5	3
Detailed design& implementation	2.25	4

The quality management system was fully operational beginning with February 1998. The final audit for the registration took place in April 1998, and it was successful. It is important to show the structure of IQSOFT's internal QMS: it followed the recommendations of the ISO standards, but it was built *to fit the specific needs of the company itself.* QMIM concepts were used throughout this project, the company making the conscious decision to concentrate on processes.

The QMS built was used actively, but IQSOFT did not want to stop quality-related activities on ISO 9001 level. Having in mind the QMIM framework, we decided to concentrate on elements not taken into account so far. Literature (theory of measurement, the Quality Improvement Paradigm, GQM, Experience Factory Organisation, e.g. in [11], [17]) shows that *measurement* has to be done to assure that quality of each object of software production is of the requested level. The QMIM framework was completed with the "metric" element, an important item "telling something" about the quality of the objects.

2.2. Broadening the scope of software quality management: ISO 9001:2000

As IQSOFT's first ISO-certificate was valid until April 2001, switching to the new ISO 9001:2000 standard ([6]) with the renewal of the registration was the obviously market- requested step by that moment.

A project was launched to build up an ISO 9001:2000-conform quality management system (QMS). The QMS the company was using for 3 years has undergone some major changes. The quality procedures referred from that moment *to all processes and departments of the company* (marketing, financial processes, human resource management were included). Previously existing procedures have been updated, understanding and following customer needs were emphasized, customer satisfaction-measurement has been started. Quality goals have been formulated, a measurement program to follow their realization has been established. Projects started to develop concrete quality plans.

As a result of the project, the company obtained the ISO 9001:2000 certificate in spring 2001, but the consequences of applying the new standard were more, in

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

terms of changes in software quality management. This process is presented in the next chapter, and is described in detail in [23].

2.3. Quality life after ISO 9001:2000

It became more obvious - it was explicitly stated - that quality management was not a separate process but rather *an aspect of the management processes*.

Being obliged to fulfill the standard requirement about setting quality goals and establish a metric program to measure them *guided the company towards connecting quality goals to business goals*.

It is interesting to analyze the changes in setting the quality goals of the company. If we look back to connecting quality goals to business goals while having in mind the Balanced IT Scorecard framework (BITS, see [21], [22]), we can notice that the quality goals of IQSOFT have been grouped in fact according to the elements considered important in the BITS. We established quality goals related to: financial issues, customers, people, processes, infrastructure and innovation elements.

To show the links between quality goals and business / strategic objectives of the company, we can use a customized form of a BITS-based representation.

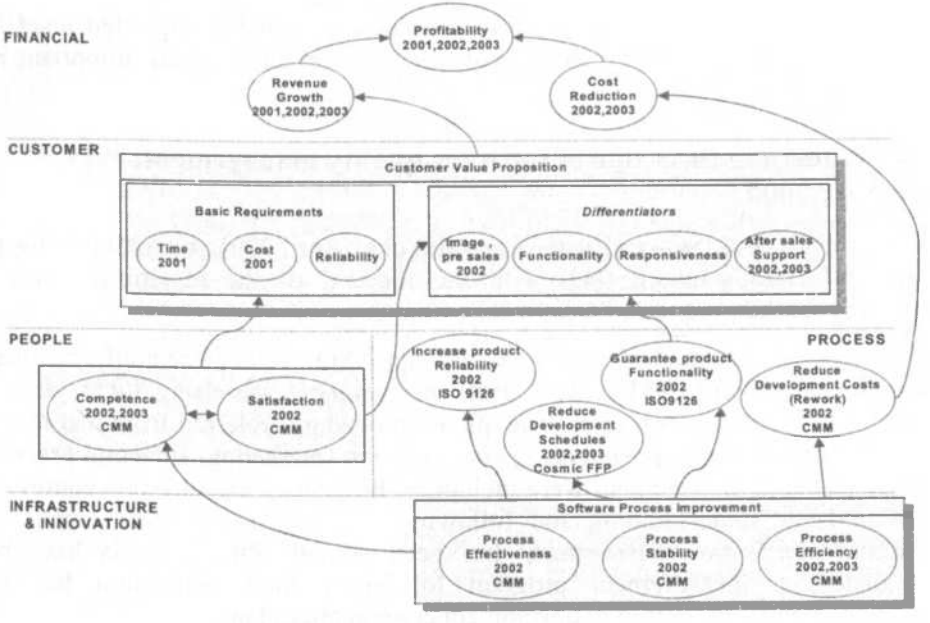


Figure 3: Connecting quality goals to business goals

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In Figure 3 we show the elements contributing to successfully execute a strategy, using a representation suggested in [21], marking also the year in which quality goals associated to each element were present in our company. The years are connected also to the moments when the usage of models different from ISO 9001:2000 emerged.

Looking to this picture, some important remarks can be made. First, it is obvious that quality goals in the first year (2001) were rather stereotypical (basically related to financial issues), while in the next years the company started to set quality goals more and more deriving from real business needs. Next, one will notice that *using further software quality models*, besides ISO 9001:2000, *appeared as a business driven quality goal*. Finally, the QMIM framework was at hand to aid the conscious choice for further quality models.

Since in 2001 there was no precise understanding about why and how measurement should be done, only project management- related data gathering was started (planned and actual time, cost and effort of projects were recorded). By 2002, measurement provided some data that, although not sufficiently accurate, guided the attention towards problematic areas of the company's activity.

2.3.1. Product quality issues

The biggest problem was considered to be the huge difference between planned and actual effort of projects, forcing us to face that our estimates were not accurate enough. The wish to make them more accurate resulted in several quality goals for 2002, related basically to *software product quality* and *software process improvement*. In Figure 3 these appear in the elements related to increasing product reliability, guaranteeing functionality and starting software process improvement.

The understanding that IQSOFT was using in fact only one of the possible process oriented approaches – ISO 9001:2000 - towards software quality was there, while other possibilities in choosing appropriate models for different important software-quality–elements were aided by the QMIM framework. This way, the need to use further quality models for further important elements of software production came natural to the company.

In the wish of having more accurate estimates the company came across the *differences between products built for different end-users*.

The need for a better understanding of product types raised, therefore we tried to define the most important *product quality characteristics and metrics* (quality goal for 2002). We formulated general guidelines based on ISO 9126 ([5]), offering a menu of possible quality attributes and metrics, from which every project manager would choose the ones most fitting to his project, and, implicitly, define the quality profile for the type of that product.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

On the other hand, the wish to have more accurate estimates drove IQSOFT to trying to connect project effort to the *complexity of the software developed.* Complexity was expected to become an extra element within the criteria used to define product types.

Among the methods for function point counting / software sizing ([18],[19],[20]), Cosmic FFP ([18]) was chosen, as it promised to give good results both in case of business applications and real time applications. This appeared as a product-related and process improvement-related quality goal in 2002. The sizing project was later incorporated into the CMM-based software process improvement project started in September 2002 (see 2.4).

2.4. The CMM –based software process improvement project

The fact that the majority of our problems presented before (estimation, defining and managing product size and quality, increasing process efficiency, managing the knowledge of human resources) could be regarded within the framework of one well known model, the CMM, was continuously promoted by keeping QMIM-principles "alive" The management-level recognition came in 2001, when an informal assessment was performed at the company by the European Software Institute. According to its results, in 2002 we already had the quality goal to run an SPI based on CMM (appearing in Figure 3 within software process improvement). The first, informal assessment resulted in a report and several improvement opportunities, according to which the company started a global process improvement project, planning to get certified according to CMM level 3 by July 2003. The high level results of the assessment are presented in

Table 2. As one can see from the table, the informal assessment at IQSOFT produced results similar to "best case profile" found in such assessments. It seemed a feasible main goal to reach CMM level 3 within a reasonable period. An SPI project was started for this purpose, planned to last 367 days and to use an effort of 800 man-days.

The project activities were possible to group into several groups of tasks. One task necessary throughout the entire life cycle was management of the CMM project. The next big group was developing and introducing the procedures required by CMM. Two basic activity types had to be performed: development and introduction of management procedures and development and introduction of technical procedures. The *management procedures* were concerned with the CMM KPA-s related to this type of activity.

To fulfill SQA KPA, the basic issue was to develop the quality management phased to projects. Within the PM related issues, the already existing planning, tracking and oversight procedures had to be updated to fit CMM requirements. The definition of the estimation procedure was not that easy due to those presented

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

in the previous chapter, but a procedure has been developed. IQSOFT's resource management, TP (Training Program) and IC (Intergroup Coordination) processes were basically good, but we did not have any peer reviews and the SSM (Software Subcontract Management) procedure had also be substantially updated.

Table 2: CMM assessment results of IQSOFT compared to the "best case profile" of an ISO-certified company

ML	Key process areas	IQSOFT assessment result	ISO best case profile
5	Process change management (PCM)	(Not rated)	Partially satisfied
	Technology change management (TCM)	(Not rated)	Partially satisfied
	Defect prevention (DP)	(Not rated)	Partially satisfied
4	Software quality management (SQM)	Not satisfied	Partially satisfied
	Quantitative process management (QPM)	Not satisfied	Not satisfied
3	Peer reviews (PR)	Not satisfied	Fully satisfied
	Intergroup coordination (IC)	Fully satisfied	Fully satisfied
	Software product engineering (SPE)	Partially satisfied	Fully satisfied
	Integrated software management (ISM)	Not satisfied	Not satisfied
	Training program (TP)	Fully satisfied	Partially satisfied
	Organization process definition (OPD)	Not satisfied	Not satisfied
	Organization process focus (OPF)	Partially satisfied	Not satisfied
2	Software configuration management (SCM)	Fully satisfied	Fully satisfied
	Software quality assurance (SQA)	Partially satisfied	Fully satisfied
	Software subcontract management (SSM)	Not applicable	Partially satisfied
	Software project tracking & oversight (SPTO)	Partially satisfied	Partially satisfied
	Software project planning (SPL)	Partially satisfied	Partially satisfied
	Requirements management (RM)	Fully satisfied	Partially satisfied

Development of a project data base was regarded as an outcome of the previous tasks. We proposed the structure described within QMIM (see Figure 2), and a first draft of the database, the process model of the company has been worked out as presented in Figure 4.

From the figure we can see that the product (Termék) is in the center of the attention, all processes: marketing, sales, managerial processes (Menedzsment folyamatok), technical development processes (Műszaki folyamatok), quality assurance (Minőségbiztosítás), support, subcontractor's work (Alvállalkozók kezelése) are executed around this item. The product development is aided by further processes i.e. human resource management and training (Hr, képzés), system engineering (Rendszergazdai folyamatok), secretarial processes (Titkársági folyamatok), financial processes (Gazdasági folyamatok). Further connections between different processes exist (e.g. quality management is connected to every

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

other process). The product and the processes all have associated quality attributes and metrics (Minőségi attribútumok és mérőszámok).

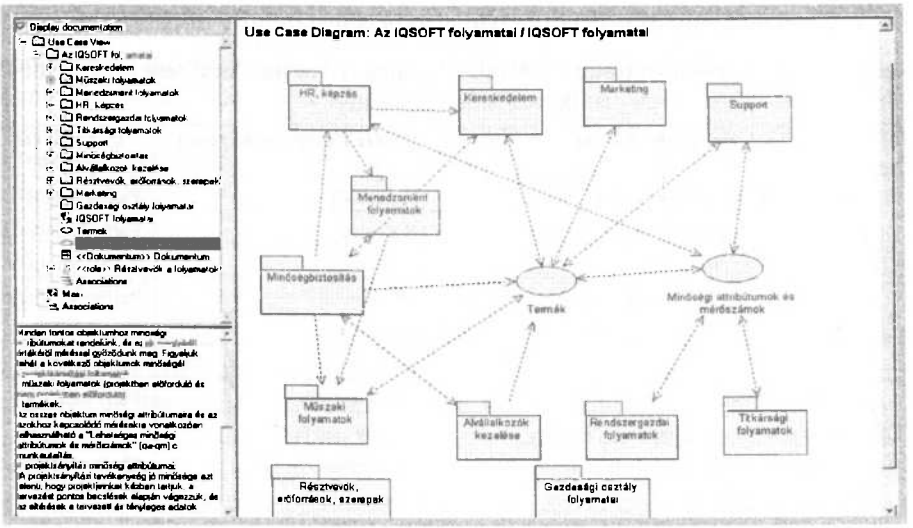


Figure 4: Use case diagram of IQSOFT objects related to quality

The presented Rose model contained all handbooks, templates, guidelines existing at that moment. Figure 4. shows in the bottom left corner the guidelines for determining quality attributes and metrics.

If we compare the structure of the Rose model to the structure of a QMIM-database suggested in the QMIM model (see Figure 2), we can see that they are in fact similar.

In the CMM-based SPI project the part connected to *development and introduction of technical procedures* caused most difficulties to the company. While RM and SCM were rather clear regarding the requirements to be fulfilled, it was not obvious how project types, project life cycle models had to be differentiated.

Based on the opinion that projects were integrating elements of different life cycles and different technologies, technicians planned to decompose the projects into smaller (basic) elements, and to develop a matrix that would show all possible associations between basic tasks, technologies, running environment, development tool, reference project.

The other important difficulty we encountered was connected to *product management*. The need to have well defined product types emerged again. We used our previous sizing results to define product types. The idea was to define a quality profile for every project type, using the previous experience described in 2.3.1.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

2.5. 2003: the year of change

General recession has not left the (Hungarian) IT sector untouched. The KFKI group – whom IQSOFT has joined in 1999 – had a decentralized structure, that no longer met the new challenges. The greatest problem of the organizational structure model the group has followed so far was the significant overlaps between the business activities of different member-companies. In 2002 the Holding had 1 consulting company, 4 software development and integration companies, 2 IT infrastructure building and safety-issues –related companies and 2 IT application and service providers, 2 companies of them being involved in more than 1 of the business areas mentioned above.

As a consequence of the IT market regression, the organizational structure of the KFKI Group was greatly simplified in early 2003. The aim was to build one big company in each of the existing business areas. Companies doing software development IQSOFT, CLASSYS, a part of ICON, and a part of ISIS were merged to create the largest Hungarian company in software application development and integration, named IQSYS Ltd. As IQSYS was the successor of IQSOFT in legal terms, its quality management system was built around IQSOFT's former QMS, integrating all the good practices, procedures, methods of the other companies into it. QMIM principles proved to be a good aid towards finding a common language: we regarded PM as being the common framework for all projects, possible to define and implement in short time, while the definition of the technology-specific technical processes was left to the next level. This way, it was possible that the departments use their previously defined processes, company-wide agreement being needed only on PM issues.

Based on these, the QMS has been developed, introduced by mid May 2003. In June 2003 IQSYS was certified by SGS and was entitled to carry on IQSOFT's former ISO 9001:2000 certificate.

The CMM-based SPI project was declared to survive the organizational change, and the quality goals set for 2003 contained this issue. However, despite the initial plans, no SEPG group was established, and software quality made a step back compared to IQSOFT-situation, as its basic scope was to integrate and keep operational the quality systems of the former companies, integrated in the QMS of IQSYS. CMM-based SPIU might continue, after IQSYS stabilized its organizational structure – which was the aim of the company for year 2004.

3. CONCLUSIONS

The first chapter of the article focused on presenting a theoretical framework that can be used to understand the important elements of software quality. The presented QMIM framework has more aspects: the static aspect helps in connecting objects and characteristics of software quality, while the guidelines help using the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

model in different organizations. The most popular software quality models, standards, methods can be placed within QMIM. Moreover, the framework helps in choosing the right approach, starting from the concrete needs of the company. The framework emphasizes the idea that using one software quality model can be misleading, while the synergic use of more quality approaches can be the best solution.

In the second chapter we presented the main results of a 11 years long case study, recorded at a Hungarian software company. We described the most positive results obtained by IQSOFT in the field of software quality management, while having QMIM in mind. We emphasized the necessity of consciously choosing the right elements and approaches at a certain moment, taking into account the maturity of the company. Connecting quality goals to business needs is extremely important. The evolution of the quality-related activities followed the principles of the QMIM model: the company started SPI by organizing its project management processes, followed by the technical processes. Issues related to product and human resources has followed as a natural requirement on a certain level of maturity. Using more software quality models at the same time was also a must resulting from concrete business needs.

The QMIM framework that resulted in fact from the SPI and research, is usable in other software companies also.

REFERENCES

[1] BALLA, K.: *The complex quality world. Developing quality management systems for software companies*. Ph.D. thesis. Beta Books, Technische Universiteit Eindhoven, 2001. ISBN: 90-386-1003-3

[2] BALLA, K., BEMELMANS, T., KUSTERS, R., TRIENEKENS, J.J.M.: *The QMIM model.* Software Quality Journal, Volume 9 Number 3, November, 2001, Kluwer Academic Publishers. ISSN 0963-9314, pp. 177-193.

[3] BOEHM, B.W.: *Characteristics of software quality*, North Holland, 1978.

[4] MCCALL, J. A.: *The utility of software metrics in large scale software systems development*, IEEE Second software life cycle management workshop, August 1978.

[5] ISO/IEC 9126. *Information technology Software product evaluation Quality characteristics and guidelines for their use.* See also: ISO IEC 9126-1., FCD 9126-1.2.,. ISO/ IEC 9126-2,. ISO / IEC 9126-3:Jan. 18, 1999.

[6] ISO 9001:2000: *Quality management systems. Requirements*. December 2000.

[7] *The Capability Maturity Model. Guidelines for Improving the Software Process*. SEI Serties in Software Engineering. Addison-Wesley, March 2002. ISBN 0-201-54664-7

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[8] KUVAJA, P., SIMILA, J., KRANIK, L., BICEGO, A., SAUKKONEN, S., KOCH, G.: *Software Process Assessment & Improvement. The Bootstrap approach.* ISBN O-631-19663-3, Blackwell, 1994.

[9] ROUT, T. P.: SPICE *A framework for software process assessment.* In: Software Process - Improvement and Practice 1(1), pp. 57-66, August 1995.

[10] CMMI 2001. *By Carnegie Mellon University Capability Maturity Model Integration,* Version 1.1. Continuous representation. Staged representation. December 2001.

[11] FENTON, N.E.: *Software metrics – a rigorous approach.* Chapman&Hall, 1992.

[12] HUMPHREY, W.: *Using a Defined and Measured Personal Software Process.* In: IEEE Software, May 1996, pp. 77-87. http://www.sei.cmu.edu/tsp/psp.html

[13] *Team Software Process.* http://www.sei.cmu.edu/tsp/tsp.html

[14] BENTLEY, C.: *Practical PRINCE2.* ISBN 0 11 702853 3. 2002.

[15] WEINBERG, GERALD M.: *Quality Software Measurement.* Vol.1.: System Thinking. Dorset House Publishing, ill. 1992. Vol.2., 1993. Vol.3.:, 1994.

[16] P-CMM People CMM. http://www.sei.cmu.edu/cmm-p/version2/

[17] BASILI, V.: *SEL's Software Process Improvement Program*, IEEE Software, Nov. 1995.

[18] ISO/IEC 19761:2003 *Software engineering COSMIC-FFP A functional size measurement method*

[19] ISO/IEC 20926:2003 *Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting practices manual*

[20] ISO/IEC 20968:2002 *Software engineering Mk II Function Point Analysis Counting Practices Manual*

[21] REO, D.: *The Balanced IT Scorecard. Quality of Strategy Vs. Strategy Execution.* In: Proceedings of Business Information Technology Management Conference, BITWorld 2001, Cairo, Egypt

[22] REO, D.: *Balanced IT Scorecard.* ESI Commercial Network presentation. 2000.

[23] BALLA, K.: *Software Process Improvement and Organizational Change.* In: 50 jaar informatiesystemen 1978-2028. Liber Amicorum voor Theo Bemelmans. pp. 181-198. March 2004. Edited by Capaciteitsgroep Infomation Systems, Faculteit Technologie Management, Technische Universiteit Eindhoven. ISBN 90-386-1928-6, NUR 982

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 91-105

SELECTION WITH THE HELP OF DATA MINING

LÁSZLÓ KOVÁCS
Department of Information Technology, University of Miskolc
H-3515 Miskolc, Hungary
kovacs@iit.uni-miskolc.hu

MARIANNA LIZÁK
Department of Human Resource, University of Miskolc
H-3515 Miskolc, Hungary
alkmlm@iit.uni-miskolc.hu

GÁBOR KOLCZA
Department of Information Technology, University of Miskolc
H-3515 MISKOLC, Hungary
kolcza@iit.uni-miskolc.hu

[Received October 2004 and accepted February 2005]

Abstract. One of the most important resources of a company is the human resource. An economic organisation has to take care of procurement and human resource management. At procurement it is decided in the process of selection which candidate will be given a position-offer. A great importance is set on this decision during the life of the company. Data mining ensures an adequate background for making well-grounded decision. This article is dealing with the methods and technics that make this process faster, more efficient and more reliable.

1. INTRODUCTION

Nowadays it is a widely accepted fact that the most important resource of a company is the human resource. One precondition of an efficiently working and profitable company is that it has to have an adequate human resources strategy. This includes the procurement (recruiting, selecting and launching), management (manpower development, performance appraisal and career planning) and the 'drain' (reducing staff, retirements, etc.) of the human resources. From the cost effectiveness point of view it is equally important that the man-power should be well-skilled and performance-orientated, thereby making profitable the organisation. It is an important point of view as well that the possible least cost is to be spent for manpower-development and recruiting, because costs can be saved with this (Figure 1). One of the important decisions for human resources managers

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

is that these two mutually exclusive factors should be optimized. Thus bring out the maximum profit and performance from the organization.

One key-task of human resources strategy is the selection. The selection is a filter which has the task to pick out from the applicants the candidates who best fulfil the requirements of the particular positions and trustworthily classify them. The selection process has a considerable responsibility, as in this phase of recruiting of staff we arrive at a decision about who will be the employee of the organization thereby having an influence on the performance and profit of the company. Selection as other activities related to human resources goes with time- and energy-cost which has no direct effect on profit. In companies the need has come to the front to reduce somehow the expenses of such tasks. With the improvement of information technology and mathematical methods and the appearance of data mining the opportunity is given to reduce these costs without reducing the energy devoted to human resources. [2]

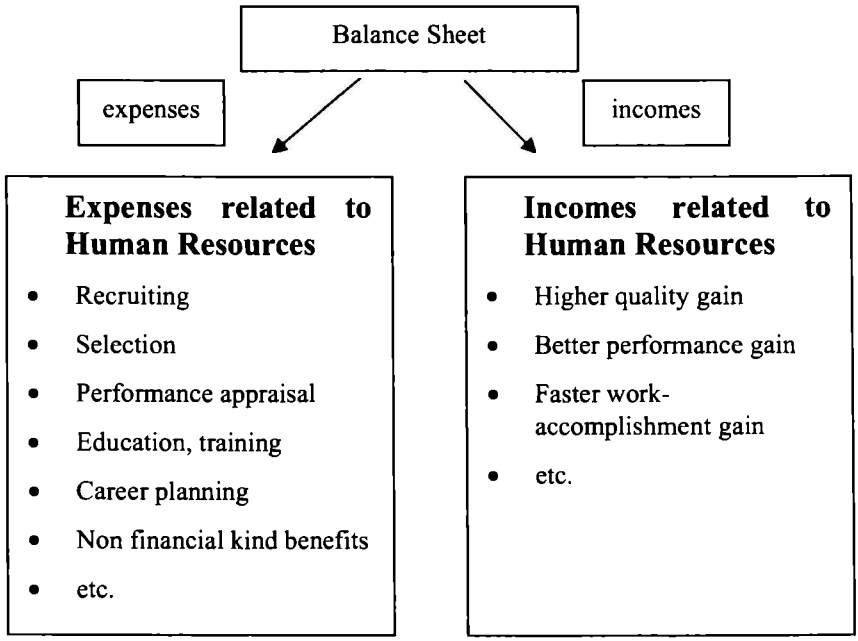


Figure 1: The optimization of the expenses and incomes related to human resources

2. OVERVIEW OF DATA MINING CONCEPTS

Because of the increasing competition among companies and the extreme consumer expectations the companies have to carry out continuous developments and analysis in order that they can keep their market position. In order to do this

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

they have to process huge amount of data. To solve this problem the data mining has arisen, which became popular in the business world in the 90's. Data mining is more and more widespread in information technology and business. Data mining gives opportunity for the companies to handle the enormous amount of data with adequate efficiency and from which they can distil useful information in order to achieve greater profit. According to the definition of the SAS Institute the task of data mining can be composed as follows: 'the process of selecting, discovering and modelling huge amount of data, which purpose is to find correlations and patterns of data, that could have not seen in advance, in order to gain business advantage.'

In this approach it is important to make difference between the data that can be found in various files or databases in unprocessed form, that are actually useless, and the information that is distilled from data and used to gain business advantage.

2.1. The application of data mining

Nowadays there are lots of application of data mining, such as: health care, direct marketing, commerce, risk-analysis, logistics, telecommunication, transportation, decision-support and human resource management. In commerce with the obtained information it is used to increase customer satisfaction, turnover and profit. In logistics with the integration of data mining the efficiency and promptness of distribution can be improved.

It is more and more widespread in large enterprises that data mining is used in human resources management, since by this means considerable cost-saving may be achieved. Also at the Dutch airways KLM as well data mining was applied in favour of cost reduction. One determinant costfactor of airways the cost of pilots, i.e. the human resources. This system was necessary because serious career-planning is carried out in the company. For the pilots the company assures continuous progress, i.e. there is a continuous labour fluctuation inside the company. The problem is that they do not know what kind of jobs the pilots will apply for. To help solving this problem they apply the software called CAPTAINS, which is developed for the Dutch airways company KLM, and which uses data mining as well. This is a planning and optimizing software in which machine-learning methods are used as well.

This system makes it possible to foretell what kind of positions the pilots are going to apply for and thereby the right number of pilots will be employed, neither too many, nor too few. KLM with applying this human resources planning software could reduce the human resource management cost by 2 per cent and the cost of investing this system returned in one year. [4]

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Data mining can be used in different fields. The different fields however need different methods, which are discussed in details in the following chapter. [5][6]

2.2. Data mining methods

The term data mining includes all methods and techniques that can be used for discovering new and relevant rules and dependencies from a huge amount of row data. The algorithms are based usally on statistical and heuristic methods. The data mining applications are complex systems as they require the adoption of the general methods to the special problem. The parametrization of the methods and the interpretation of the results requires expert users. The data mining systems are useful tools only in the hand of good experts.

From the logical and functional viewpoint, the process of data mining can be divided into the following phases:

- The first step is the analysis of the required information and the available information sources.
- The analysis of data format of the data sources. The quality of the data sources should also be examined.
- Data from different sources should be transfered to a common data stage. The data transport includes the extraction, the transportation and the format conversion steps.
- The next step is the data integration on the staging area. This includes the development of the common data schema and the discovery of the inconsistencies among the data from different sources.
- To increase the efficiency of computation, the amount of data to be proccessed should be decreased with some kind of data reduction method.
- A pre-analysis phase with some kind of OLAP or statistical tool. These tools help to illustrate the overall behavior of the data set without any deep analysis. This helps to localize the problem areas.
- The definition of the concrete goals of the analysis. The selection of the data mining tasks best fitting to the investigated problem area is an important step done by experts.
- Based on the task, the selection of the data mining algorithm and method best fitting to the investigated task is also an important step done by experts.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

- The definition of the parameters and the constraints related to the selected data mining method.
- Running the tests.
- Interpretation of the results, definition of the discovered rules and dependencies for non-expert users.
- The evaluation of the method, how correct is the generated result.

The algorithms of the data mining tools are based on some well-known mathematical methods. In the practice, the following methods are usually applied for rule discovery problems:

- Discovery of association rules. The goal is to determine which objects implicate wich other objects. The method determines first which objects occur often together and the direction of the implication rule is calculated next.
- Classification. The training objects are assigned to some predefined classes. The method discovers the hidden dependencies among the object parameters and the class labels assigned to the objects.
- Clustering. Only the training objects are known without any predefined class labels. The method determines the groups of similar objects based on their attributes. The number of groups is usually not known a priory.
- Detection of typical event chains. The method discovers the ordering of the events and the most possible chain segments.
- Deviation analysis. First the group of objects with average behaviour are determined then the outlier objects are discovered. The calculation is based on the object attributes.
- Nearest neighbour search: The training set includes a large amount of samples. The method is aimed at determining the most similar objects to a query object. [6]

In our investigation, the classification method is the appropriate method. In classification processes, it is assumed that the patterns are stochastically independent. A d-dimensional pattern vector is denoted by $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_n) \in \mathbf{R}^n$ Every pattern vector is associated with a class c_j, where the total number of class is m. Thus, a classifier can be regarded as a function

$$g(\mathbf{x}) : \mathbf{R}^n \rightarrow \{c_1,\ldots,c_m\}. \quad (1)$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The optimal classification function is aimed at minimizing the misclassification risk. The R risk can be measured by an appropriate cost value. The risk value depends on the probability of the different classes and on the misclassification cost of the classes.

$$R(g(\mathbf{x}) \mid \mathbf{x}) = \Sigma_{cj}\, b(g(\mathbf{x}) \rightarrow c_j) P(c_j \mid \mathbf{x}), \tag{2}$$

where $P(c_j \mid \mathbf{x})$ denotes the conditional probability of c_j for the pattern vector $\mathbf{x}$ and $b(c_i \rightarrow c_j)$ denotes the cost value of deciding in favour of c_i instead of the correct class c_j. The b cost function has usually the following simplified form:

$$b(c_i \rightarrow c_j) = \begin{matrix} 0, & \text{if } c_i = c_j \text{ and} \\ 1, & \text{if } c_i \neq c_j. \end{matrix} \tag{3}$$

Using this kind of b function, the misclassification error value can be given by

$$R(g(\mathbf{x}) \mid \mathbf{x}) = \Sigma_{g(x) \neq cj} P(c_j \mid \mathbf{x}). \tag{4}$$

The optimal classification function minimizes the $R(g(\mathbf{x}) \mid \mathbf{x})$ value. As

$$\Sigma_{cj}\, P(c_j \mid \mathbf{x}) = 1 \tag{5}$$

thus if

$$P(g(\mathbf{x}) \mid \mathbf{x}) \rightarrow max \tag{6}$$

then the

$$R(g(\mathbf{x}) \mid \mathbf{x}) \tag{7}$$

has a minimal value. The decision rule which minimizes the average risk is the Bayes rule which assigns the $\mathbf{x}$ pattern vector to the class that has the greatest probability for $\mathbf{x}$.

The Bayes classifier that minimizes the misclassification error is defined by

$$qB(\mathbf{x}) = \text{argmax}\, q_j(\mathbf{x}), \tag{8}$$

where q_j is the a posteriori probability of class j at pattern $\mathbf{x}$:

$$q_j(\mathbf{x}) = P(c_j \mid \mathbf{x}). \tag{9}$$

The misclassification cost is equal to

$$R(g(\mathbf{x}) \mid \mathbf{x}) = 1 - qB(\mathbf{x}). \tag{10}$$

The lower is the $qB(\mathbf{x})$ value the greater is this cost. The greatest cost is yielded if every class has the same probability for the pattern vector $\mathbf{x}$:

$$q_j(\mathbf{x}) = 1/m\,, \tag{11}$$

in this case

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$R(g(\mathbf{x}) \mid \mathbf{x}) = 1 \quad 1/m \tag{12}$$

The lowest misclassification value is equal to 0. This occurs if only one class has a nonzero probability for the pattern vector, i.e. $qB(\mathbf{x}) = 1$. [6][7]

3. THE PROCESS OF SELECTION

Data mining can be used in the process of selection independently of the fact that it is carried out with inside, outside, innovative or conservative strategy. Figure 2 shows the process of selection. It can be seen that selection is not an easy task. In function of the size of the company and the type of the vacancies this process can be simpler or more complex, thus some examination may be cancelled or even may be expanded.

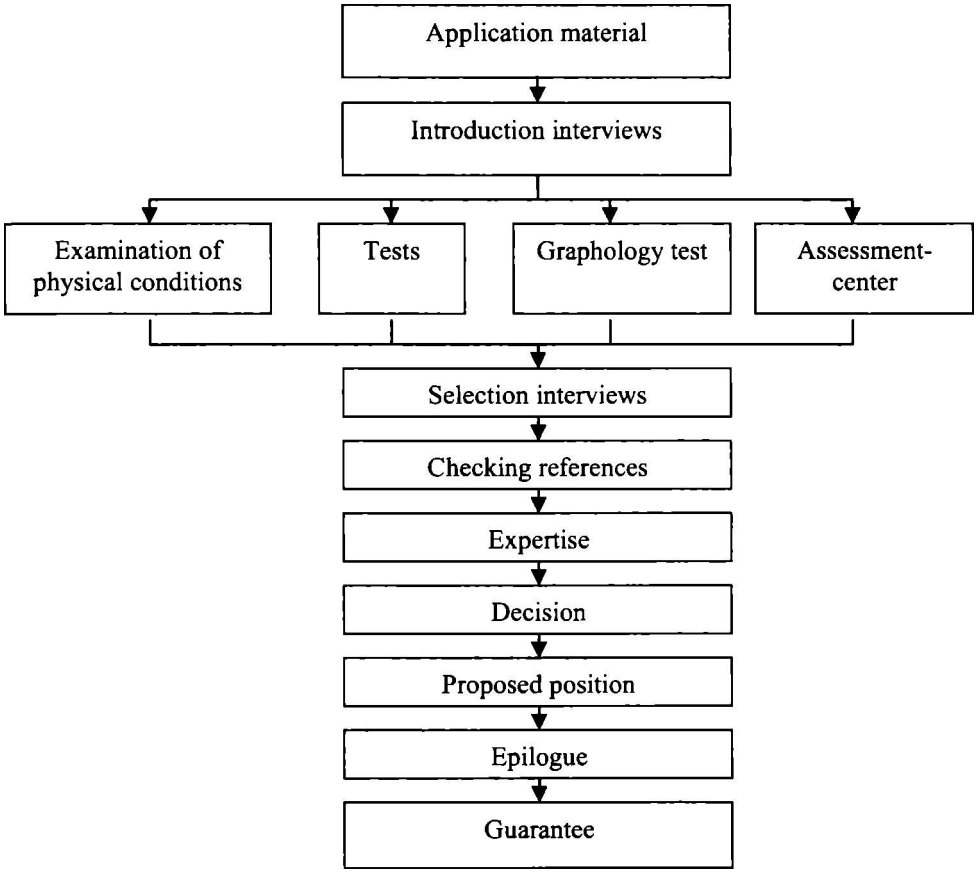


Figure 2: The process of selection

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The complexity of the selection does not depend only on the size of the company but on the importance of the vacancy.

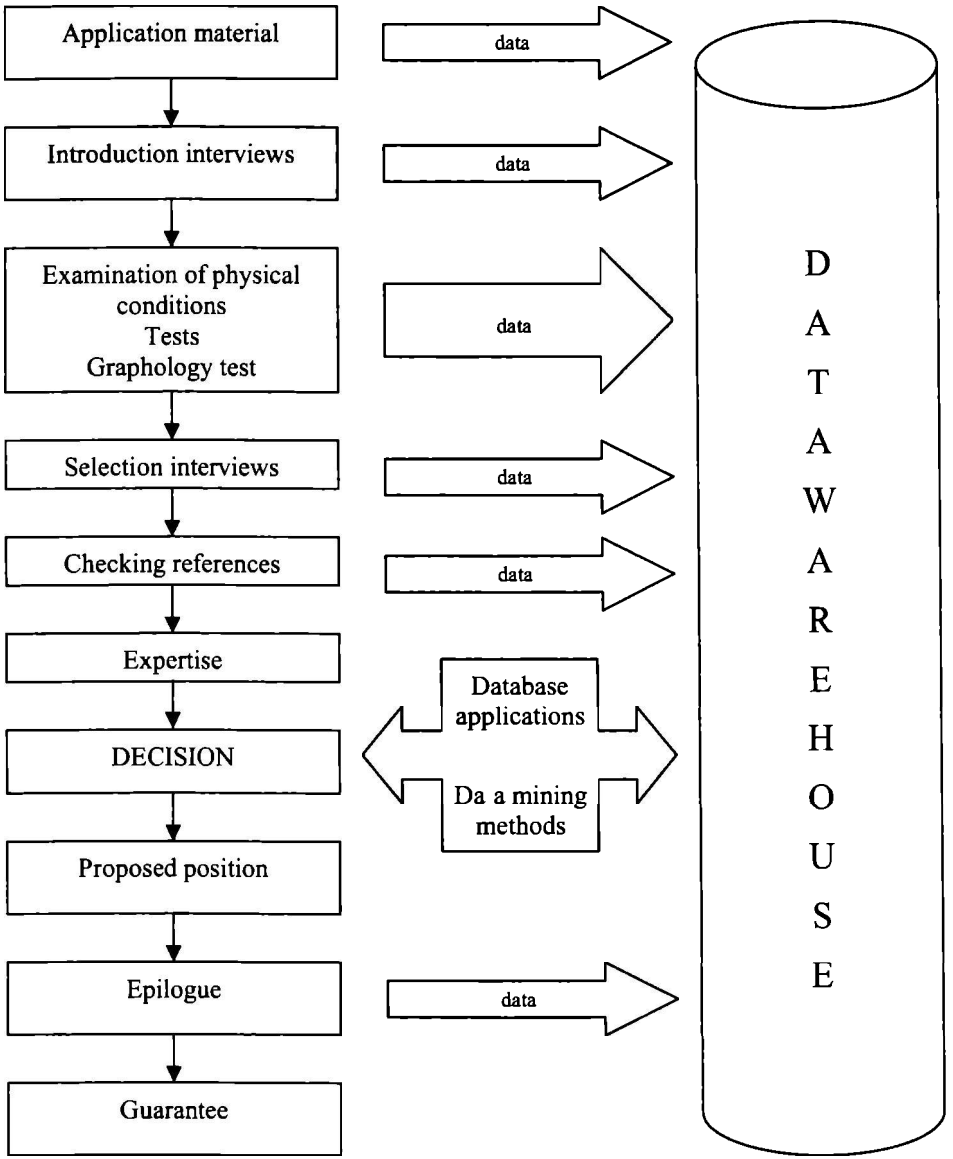


Figure 3: The process of selection with applied data mining

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In order to be able to use data mining in selection we need a datawarehouse filled with data. In the age of Internet it does not make difficulties to store cv-s, motivation letters and other test results about a candidate in a database, as most company has an application form available on Internet.If a candidate fills it automatically gets in a database. On Figure 3 we can see how selection can be made effective with data mining. [1][2]

Nowadays enterprises record almost everything on computers. Thus it cannot be a problem to store useful information in datawarehouses.

In every step of the selection we record data in the datawarehouse. By Inmon „a Data Warehouse is a subject-oriented, integrated, non-volatile, and time variant collection of data in support of managements decisions".The datawarehouses give a possibility for storing a huge quantity of data in a suitable structure and for ensuring the quick performing of putting questions on the data.Instead of datawarehouse traditional databases or files can be used, though it has the disadvantage that the access of data is slower and the storage of same amount of data consumes more space. At application material we can store the qualifications, further purposes, marital status etc. of the candidate. At introduction interviews the personal impression of the examiner, the results of the candidate in various examinations, personality-, IQ- and other tests can be stored. In the selection interview we can get information about the candidate's motivation and characteristics.

The most important thing is to store data in the datawarehouse in the epilogue phase, because in this step turns out how well the new employee is approved. This ensures the feedback which helps to find out which attributes the 'good' and 'best' manpower possess.

It is important to have as much data as possible, because the more data we have the more efficient data mining can help in selection to classify candidates and make better decision. [8][9]

4. SELECTION WITH THE HELP OF DATA MINING AT THE DEPARTMENT-STORE OF TESCO, MISKOLC

The TESCO Global Inc. makes retailing by selling 70000 different types of products. To carry out this an adequate number of employees with the necessary qualification is needed. Currently about 300 employees work in the TESCO in Miskolc. According to this an effective staff-procurement strategy has been elaborated.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

If there is some vacancy, the corresponding candidates are selected primarily from the database, maintained by TESCO. If in that database an adequate candidate is not found, which has a quite small chance, because currently there are 6000 pieces of application given in, the position is advertised in the press. For these advertisements application forms made by TESCO are accepted because in these forms all necessary data are asked for. The first step of filtering is based on these data. Each application material is provided with a serial number and every applicant has to sign a paper suitably to the relevant passage of the Data Protection Act that the data of the applicant is going to be deleted in half a year. The application materials are stored in Excel files, and it is classified into a position to which it is potentially suitable. After one month of the registration of the applicant's data a respond is sent to the applicant which informs about having been registered in TESCO database and as soon as there is a job the applicant will be contacted.

In case of a vacancy, only the application materials will be found that fulfil the requirements of qualifications and expertise. These filtered data is given to the head of the department who arranges a suitable time with the applicants and selects from the applicants by a personal interview. In case of every position the interview to which the managers are prepared in a training is enacted similarly.

The data needed to data mining are stored in Excel spreadsheets, which is not the most effective method as the structure of the files could not make possible the fast access, but it is one solution to store and access data.

In TESCO department-stores there are various performance classification levels into which every employee is classified once in a half year. These are registered in files as well. The method of association makes it possible to determine that mostly what kind of attributes from the application form are associated to the excellent and good performance. With the help of this method a rule-based deduction system can be made. In this case this can be regarded as a learning-algorithm, as it filters the most adequate by the experiments came from the existing data. With the association method relations can be discovered. These relations, however, usually are ambiguous, stochastic-featured. The closeness of stochastic relations can be analysed with coefficients and corellation- and regression-calculation used in statistics. The closeness of the relationships can be analyzed simply, clearly and effectively by means of the association coefficients therefore I will investigate this method in the future.

With the Yule association coefficient only relations between alternative criterions can be analysed. Primarily we would like to analyse the relation between the applicant's data and later performance. This could be e.g. the relation between the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

received score for communication skill or other skills during the interview and the later performance evaluation. This coefficient could only be used in this case if we put aside the scalability of an attribute and we would only determine if the applicant has or has not the specified skill. On the one part this would not be the reality and on the other part this would worsen the efficiency of the software.

With the use of Csuprov association coefficient there is a possibility to analyse the attributes that are not alternative. In the first step the necessary data should be inserted in a contingency-table (Figure 4).

A\B	B_1	B_2		B_j		B_t	Σ
A_1	f_{11}	f_{12}	...	f_{1j}	...	f_{1t}	$f_{1.}$
A_2	f_{21}	f_{22}	...	f_{2j}	...	f_{2t}	$f_{2.}$
...							...
...	...	...	...	...	...	...	...
A_i	f_{i1}	f_{i2}	...	f_{ij}	...	f_{it}	$f_{i.}$
...							...
...	...	...	...	...	...	...	...
A_s	f_{s1}	f_{s2}		f_{si}		f_{st}	$f_{s.}$
Σ	$f_{.1}$	$f_{.2}$	...	$f_{.j}$	...	$f_{.t}$	n

Figure 4: Contingency-table

With the use of this association coefficient the relation between two attributes of the applicants can be analysed. Thus e.g. we can determine how strong the relation between qualification (elementary, secondary, higher) and communication skill or customer-orientation. In the TESCO supermarkets the ranking of performance levels of employees is indicated by different letters in a given field as e.g. in the fields of the communication relationship and customer-service. In a given field the excellent performance is indicated by A and B, the good one – by C, the weak performance is indicated by E and F; D is used for indicating the developing performance and X for indicating the performance that cannot be evaluated. According to it the relationship between these two attributes can be determined in the following Table. Figure 5 shows this.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Customer-orientation / qualification	A	B	C	D	E	F	X	Σ
Elementary	f_{11}	f_{12}	...		...			$f_{1.}$
Secondary	...	...	...		...			$f_{2.}$
Higher	f_{31}						f_{37}	...
Σ	$f_{.1}$	$f_{.2}$	...	$f_{.j}$	...	$f_{.t}$		n

Figure 5: Analysis of relation between performance and qualification

Based on the contingency-table f_{ij}* (frequency assumed for independence) can be determined:

$$f_{ij}^{\cdot} = \frac{f_{i.} * f_{.j}}{n} \tag{13}$$

From which χ^2 can be calculated by the following equation:

$$\chi^2 = \sum_{i=1}^{s} \sum_{j=1}^{t} \frac{(f_{ij} - f_{ij}^{\cdot})^2}{f_{ij}^{\cdot}} \tag{14}$$

The Csuprov association index can be calculated by the following formula:

$$T = \sqrt{\frac{\chi^2}{n * \sqrt{s-1} * \sqrt{t-1}}} \tag{15}$$

As the number of knowledge variants are not equal, hence instead of Csuprov coefficient the Cramer index is used, which can be calculated as follows:

$$C = \frac{T}{T_{\max}} \tag{16}$$

where

$$T_{\max} = \sqrt[4]{\frac{s-1}{t-1}}. \tag{17}$$

With the help of Csuprov association coefficient we can analyse how strong the relation between the attributes of an applicant. Besides this naturally all possible reasonable attribute-pairs should be examined.

With the combination of assotiation and similarity methods there is a possibility to classify the applicants filtered by the rules created with the association method by selecting the applicants whose application material resembles best in attributes to

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

the application material of excellent and good performance employees. It presents itself that clustering belongs to these two methods. With the help of this the classification of applicants can be carried out with higher confidence, as not only the data of individual employees are analysed but also the group of excellent and good performance employees and the rules are determined by these criteria as well.

But we should think about that generally the performance of the employees is non-uniform. Sometimes it comes to a rise another time to a fall. A generally excellent worker can present poor performance sometimes. If we take the previous combination example then it can be seen that it cannot give adequate confidence, as the created rules are strongly influenced by the performance of the individual eemployees at any given time. This problem can be eliminated by the method of prediction. This is a statistical method as well that takes account of the trend of performance, the seasonal variation and turns in performance as well. In a word if this method is integrated in our existing system then the variation in performance can be considered and our rules are not based only a given month's performance.

At analysing time series in the first step we should determine if it is additive or multiplicative. The time series analysed by us are in any case additive, as the employee is fired in case of too big and increasing variations in performance. In case of additive time series we apply the following formula:

$$y_{ij} = \hat{y}_t + S_j + c + v_t. \tag{18}$$

Where '$\hat{y}_t$' means the trend-value of the time series at a given 't' moment, 'S_j' means the seasonal variation, 'c' is the random effect and 'v_t' is the turns in performance. With this formula we can calculate the probable performance at a later date. This value is determined by four factors. The first factor is the '$\hat{y}_t$' trend , which determines the basic direction of the time series. Because of the nature of the data here we will calculate with linear trend. The linear trend-function can be determined by the following formula:

$$\hat{y}_t = b_0 + b_1 * t, \tag{19}$$

where

$$b_0 = \frac{\sum y}{n}, \qquad b_1 = \frac{\sum t * y}{\sum t^2} \tag{20}$$

The next factor determines in what direction and what extent the seasonal variation and the period deviates the data of the time series from the basic direction. Subtracting the trend value from the original value the seasonal variation can be determined for each quarter.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

We do not study the cyclical swings, as for such a long time data are not available.

However we can calculate with the random factor – after having calculated the seasonal variations (S_j) – by the following formula:

$$y_{ij} - \hat{y}_{ij} - S_j = v_{ij}. \quad (21)$$

Using the method of prediction alone there is a possibility to reduce or eliminate the performance-swing of the employees, as with the help of the method we can recognize the events and circumstances that had generated the performance-swing. With the help of this method the breweries discovered that the beer-consumption is strongly affected by temperature.

One rule can be for example when somebody is unemployed, young and was employed for a short period formerly then that person's performance will not be adequate or he will notice in a short time. In both cases the fluctuation rate will rise, which will result the restart of the human-resources procurement process, which goes with serious expenses. Recently the realization of the present program is in an initial stage, the program has not been implemented yet. In addition to the implementation of the aforementioned processes and methods, other different data mining methods (as e.g. the claster analyzing) that can effectively be used in the course of staff procurement will also be investigated. [10][11][12][13]

5. CONCLUSIONS

During human-resources procurement the data mining supported selection makes it possible to reduce cost and speed up the process execution. In TESCO department-stores one of the most important human-resources strategy principal is to reduce fluctuation. Applying data mining there is a possibility to carry out this, as based on the processed data it is predictable that the particular applicant how effective and loyal will be.

REFERENCES

[1] LIZÁK, M.: *Személyzetbeszerzés, munkaerő-toborzás in Tothné Sikora Gizella: Humán erőforrások gazdaságtana,* Bíbor Kiadó, 2004. pp. 259-277.

[2] BALOGH, G.: *Emberi erőforrás menedzsment – felsőfokon*, Management Budapest, 2002.

[3] BORGULYA, I., FARKAS, F.: *Emberi erőforrás menedzsment kézikönyv*, KJK-Kerszöv Budapest, 2003.

[4] ADRIAANS, P., ZANTINGE, D.: *Adatbányászat*, Panem Könyvkiadó Kft, 2002.

[5] PARR RUD, O.: *Data Mining Cookbook,* Wiley Publishing Inc., Canada, 2001.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[6] HAN, J., KAMBER, M.: *Adatbányászat koncepciók és technikák*, Panem Könyvkiadó, 2004.

[7] BERRY, M.J.A., LINOFF, G.S.: *Data Mining Techniques,* Wiley Publishing Inc, Indianapolis, Indiana, 2004.

[8] ENSOR, D., STEVENSON, I.: *Oracle tervezés*, Kossuth kiadó, 2000.

[9] GARCIA, H., ULLMAN, M.J.D. WIDOM, J.: *Adatbázisrendszerek megvalósítása*, Panem Könyvkiadó Kft, 2001.

[10] *Sztochasztikus kapcsolatok elemzése* Oktatási segédlet, Miskolci Egyetem Gazdaságtudományi Kar, 2001.

[11] VÁGÓ, ZS.: *Idősorok sztochasztikus modelljei*, Tantárgyi segédlet, 1995.

[12] MICHELBERGER, P., SZEIDL, L.: *Alkalmazott folyamatstatisztika és idősor analízis*, Typotex, 2001.

[13] HUNYADI, L., VITA, L.: *Statisztika közgazdászoknak*, KSH, Budapest, 2002.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 107-119

A DETAILED EXAMPLE OF APPLYING CONSTRAINTS ON A LOGISTICAL PROBLEM

ELEMÉR KÁROLY NAGY
Department of Control Engineering and Information Technology,
Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics
eknagy@kempelen.iit.bme.hu

[Received November 2004 and accepted February 2005]

Abstract. This article demonstrates the use of constraints to reduce the algorithmic complexity of industrial problems through a numerically detailed example. To achieve this, the article first presents the basics of algorithmic complexity and constraints as well as the types and uses of constraints. Second, the article presents a logistical example and determines its complexity. Third, constraints are applied on the problem. Fourth, the problem is modeled in a constraint programming environment and the different results received from different constraint sets are evaluated.

Keywords: Algorithmic complexity, constraint programming, logistical problem

1. THE GROWTH OF COMPUTING POWER

In the early days of computing, computing power was so rare and expensive that dozens of scientists had to work on the same computer and they could solve only simple problems. Later, every scientist could have a personal computer and they could do most of their work on it. However, if they had difficult problems they still had to work on the mainframe. At the end of the 20th century, everyone had a personal computer that was almost as powerful as a mainframe, and scientists solved difficult problems on their PCs. When they needed even more computing power, they entered the problem in a cluster of thousands of mainframes. Yet there are problems that cannot be solved even with clusters. These problems have enormous algorithmic complexity and they are believed to take at least a hundred year even if the available computing power is doubled every year.[1]

2. ALGORITHMIC COMPLEXITY

"Algorithmic complexity" is an abstract expression which is used to describe the computation power needed to solve a problem with a given algorithm.[2] It is mostly independent of hardware and implementation details, depending only – at least in theory - on the algorithm, the problem class and the type of resource used.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

One such problem class is the N –queen problem, in which we need to find all the solutions of the problem "N queens on an $N*N$ chess table, none of them can hit any of the others in one move"

One such resource is the memory, measured in Memory Units or MUs, which – in our example – denotes the memory needed to store an $N*N$ table with N queens. Another such resource is CPU time, measured in Computational Units or CUs which – in our example – denotes the time needed to check an $N*N$ chessboard if it fulfills the requirements of the N-queen problem. The exact amount of these resources may vary depending on the hardware, the programming language, the data structures, etc. In our example, the memory denoted by "one MU" may vary from about $N*\log_2(N^2)$ bits to $4*N^2$ bytes, namely 6 to 256 bytes for $N=8$.

Using these definitions, we could state that my hypothetical algorithm solves the 4-queen problem with using the maximum of 256 MUs and 128 CUs, solves the 8-queen problem with using the maximum of 2048 MUs and 512 CUs.

To describe the memory and CPU usage of my hypothetical algorithm for all possible Ns, I either need to produce a table that contains every N and the corresponding MUs and CUs or I need to find a formula that generates this table. In practice, it is widely accepted to use the formula of "maximum algorithmic complexity" which is denoted as $O()$ and pronounced as "big ordo" This operator is adopted from Calculus, in which the definition is (1), where C and N are constants and a_n and b_n are sequences:

$$a_n = O(b_n) \Leftarrow \exists C : |a_n| \le C|b_n|, n > N \tag{1}$$

In computer science, "$O(N^3)$ in terms of memory usage" is used to denote that "the given algorithm solves the given problem class with parameter N using the maximum of $C*N^3$ MUs, even in the worst case, where C depends on hardware and implementation but is independent of N" For example, my hypothetical algorithm solves the N-queen problem in $O(N^3)$ MUs and $O(N^2)$ CUs. In this case, $C=4$, but $C=4096$ would still result $O(N^3)$. This is the consequence of the definition and reflects the fact that scalability[1] cannot compensate non-linear algorithmic complexity.

[1] Faster processors, increased storage capacities, multiprocessor systems, clusters, distributed applications, etc.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

There is another adopted operator in use, namely $o()$ or "small ordo" with the original Calculus definition of (2) , where c and N are constants and a_n and b_n are sequences:

$$a_n = o(b_n) \Leftarrow \forall c > 0 : |a_n| \leq c|b_n|, n > N \quad (2)$$

This operator is used sometimes incorrectly to denote the "minimum algorithmic complexity" which is slightly different from the exact mathematical meaning[2] This may be tolerable as the minimum algorithmic complexity usually has far less importance in practice than the maximum algorithmic complexity. Therefore, I will use the operator $q()$ to denote the minimum algorithmic complexity, so "$q(N^3)$ in terms of memory usage" is used to denote that "the given algorithm solves the given problem class with parameter N using the minimum of $C * N^3$ MUs, even in the best case, where C depends on hardware and implementation but is independent of N "

There are algorithms/problems where q and O differ. For example, if we are not interested in all solutions of the N -queen problem, just in the first solution, the complexities may be $q(N^2)$ and $O(N^4)$ in terms of memory usage. If q differs from O, it is practical to use both of them when we speak about complexity, but we may refer to O only as it denotes the worst case. In algorithms where q and O are equal, it is sufficient to use O only.

In practice, an English-Mathematician Dictionary [3] contains the following translations: $O(const)$ = "utopian", $O(\log N)$ = "excellent", $O(N)$ = "very good", $O(N * \log N)$ = "decent", $O(N^2)$ = "not so good", $O(N^3)$ = "pretty bad", $O(N^4)$ = "terrible", $O(const^N)$ = "disaster"[3] This sequence of complexities is also referred to as "complexity classes", so an $O(N^2)$ problem is two classes harder than an $O(N)$ problem. Until now, we used to denote q and O to describe the complexity of an algorithm. However, we may use q and O to describe the complexity of a problem, especially if we have an algorithm with known complexity that solves the problem.

[2] If the algorithm uses $4 * N^3$ MUs in every case, it is $q(N^3)$ and $o(N^{3.0001})$ but not $o(N^3)$

[3] $const > 1$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

An $O(N * \log N)$ problem denotes a problem that has at least one $O(N * \log N)$ solution. Sorting elements in an unsorted array with N elements is an $O(N * \log N)$ problem in terms of execution time. [4]

In some cases, it is mathematically proven that no faster algorithm exists, these problems are often referred as "proved $O(X)$ problems"

It is generally accepted that an algorithm designed to find a simple solution to a problem is "way faster" than an algorithm designed to find all solutions while algorithms designed to find the best solution are almost identical, at least in the terms of MU and CU, with the ones designed to find all solutions. This is reflected in algorithmic complexity as these three cases define three different problems. The maximum complexities of these problems are often the same and the minimum complexities of these problems almost always differ by at least two classes.

3. INDUSTRIAL LOGISTICS EXAMPLE

As defined in [5], "in an industrial context, logistics means the art and science of obtaining, producing, and distributing material and product in the proper place and in proper quantities" One such logistical problem is the problem of manufacturing cells[4] Manufacturing cells can produce different products but changing the type of product requires time (referred to as retooling). The products and the resources may have different restrictions that are direct consequences of storage capacities, workforce limits, delivery deadlines, etc.

One such example is the following (MU is money unit, TU is time unit): There are 3 identical manufacturing cells (C1, C2, C3). There are 3 resources (R1, R2, R3). There are 4 tools , two of each type (T1/1, T1/2, T2/1, T2/2). T1 tools produce one R3 from two R2s and two R1s in every TU. T2 tools produce three R2s from two R1 in every TU. The storage costs of R1, R2, R3 are 1, 1, 3 MU/TU. Retooling takes 1 TU. R1 and R2 resources can be bought at a price of 5 MU/piece in quantities of 20 while R3 can be sold at a price of 100 MU/piece in quantities of 10. Buying and selling takes 1 MU. At the beginning of the shift, we have 1000 MUs, C1 has T1/1, C2 has T2/2, C3 has T1/2, and we have 100 pieces of R1s, 11 pieces of R2s and 1 pieces of R3. We have to prepare 50 pieces of R3s in the

[4] The problem of manufacturing cells is often treated as scheduling or manufacturing problem when "logistics" is used in a narrower sense. However, when the definition omits production from logistics, then logistics (in a broader sense) becomes a sequence of logistics (in a narrower sense) and production problems. As these problems are not independent, the complexity of the main problem is not reduced but increased.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

storage at the end of the 56th TU when it will be removed as ordered by the CEO. We have to achieve the most MUs at the end of the 100th TU. What shall we do?

This example is a simplified real-life example, in which the quality of the solution found has great impact on profit. To find the best solution, a PPS[5] should utilize an algorithm that finds the optimal solution before the shift starts. Let us assume that we have a state-of-the-art computer that can store 10 000 000 states in its memory and can analyze 100 000 states per second. Let us assume that the computer has 24 hours to find the best solution.

4. THE COMPLEXITY OF THE EXAMPLE

The problem of manufacturing cells can be modeled as a single-source multi-destination directed grah search problem. In a graph search problem, there are nodes (also referred as states) and edges (also referred as transitions). In the example, there is a single source node (the beginning of the shift), there are destination nodes (the possible outcomes at the end of the shift) and there are intermediate (mid-shift) nodes. In the example, the nodes are arranged in layers, each layer contains nodes with the same TUs. Also, the edges always connect two adjacent layers and the destination's TU is always higher than the source's TU exactly by one. In a general graph search problem, these restrictions do not apply.

A search algorithm takes the current node and chooses an edge to follow, entering into the next current node. The difference between the search algorithms comes from the difference in the choices they make. If the edges have "distance" or "cost" values and it affects the choices made by the algorithms, we speak of guided search algorithms.

In this example, as each cell can either retool or produce or stay idle, we have 3 possibilities for each cell. We may decide to buy or to not buy 20 pieces of R1s, to buy or not to buy 20 pieces of R2s, to sell or not to sell 10 pieces of R3s, thus we have 2 possibilities for each resource. This means we have to make six choices and thus we have $3*3*3*2*2*2=216$ possibilities in each TU (216 transitions from every state).

The basic unguided algorithm to solve such a problem is the breadth-first search.[6] To find the best solution (the destination state with the most MUs) with the breadth-first search it is necessary to examine all possible states. To find all possible solutions, we would have to check 216^{100} possibilities and we would need to compute for about 10^{221} years with the given computer. It is impossible to

[5] In this context, Production Planning System

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

do so because finding the most profitable solution(s) for this problem with simple breadth-first search without applying constraints has the complexity of $O(216^N)$ where N is the number of TUs. Guided algorithms like the A* search find one of the best solution faster than unguided search algorithms if certain conditions apply[6] and so if they do not need to check all possible states. In this example, we require that the search algorithm checks all states that are not disqualified by constraints, so the breadth-first search is used for the sake of simplicity.

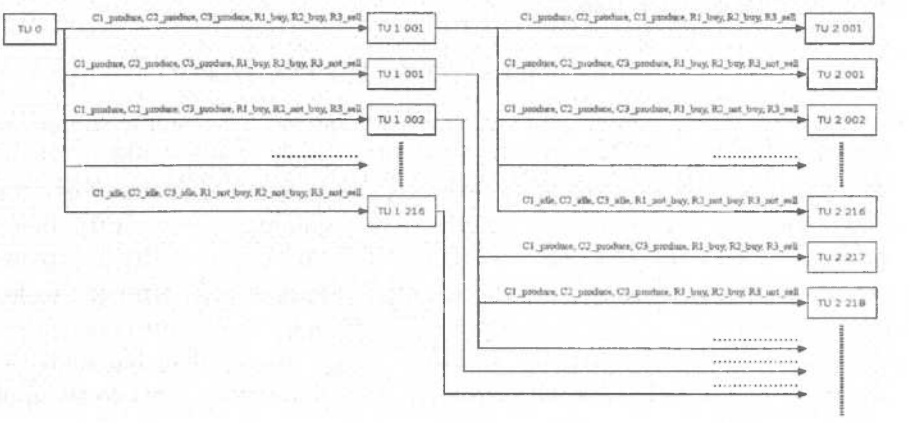


Figure 1: Simple BFS

5. CONSTRAINTS

Constraints are restrictions that reduce the number of possible actions, reducing the number of states and thus they reduce the necessary computing power.[7]

Constraints are either:

- internal problem constraints (we cannot produce R3s if we have no R2 in the storage),
- external problem constraints (we need to have 50 R3s at the end of the 56th TU), or
- algorithm constraints[7] (one algorithm could find out that if we need to have 50 R3s at the 56th TU, then – as we cannot produce more than two R3s in a TU as we have only two T1s – we need at least 48 R3s at the 55th TU, 46 R3s at the 54th TU, , 2 R3s at the 32nd TU).

[6] Detailed discussion is not feasible within the frame of this article

[7] Algorithm constraints include deducted constraints (constraint deducted from other constraints) but also include constraints that are not formally deducted.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

A problem can be either – by internal and external constraints:

- over-constrained (there is no solution to the problem);
- under-constrained (the number of solutions is too great); or
- well-constrained.

Algorithm constraints may be added to the problem to reduce the number of transitions from the states as long as they do not remove any of the best solutions of the problem. To make the example solvable, we introduce the following four algorithm constraints:

- Constraint 1: We do not buy 20 R1s if we have 11 or more R1s.
- Constraint 2: We do not buy 20 R2s if there are two T2s equipped or we have 4 or more R2s.
- Constraint 3: We sell 10 R3s if we have more than 9 R3s in the storage and it is past the 56th TU.
- Constraint 4: We sell 10 R3s if we have more than 59 R3s.

These constraints can only be applied to this particular example. When applied, they reduce the 8 possibilities of buying/selling to the average of 3. This results in an enchanted performance as we reduced the number of states to check from 216^{100} to about 81^{100}, which is 10^{43} faster.

According to the definition, the problem still has $O(216^N)$ complexity as O denotes the maximum algorithmic complexity. If we analyzed all states, we would get $O(81^N)$ for the given specific example with the four example constraints added. However, if we change any specific data (the starting amount of R1s, for example) then it might not be $O(81^N)$ while it still would be $O(216^N)$.

6. ADDING AND USING CONSTRAINTS

There are two ways of adding constraints: automatically or manually. Manually added constraints tend to be "stronger" as they reduce the complexity more severely than automatically added constraint, but also have the tendency of disqualifying valid solutions even best solutions due to human error. Automatically added constraints often do not reduce the complexity enough to solve the problem in the given time but they usually speed it up.

If an algorithm tries to check all possible solutions it may use constraints in one of the following ways:

- In every state, the algorithm checks if all the constraints are fulfilled. If any of the constraints is violated, the algorithm steps back (backtrace). For

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

example, if we are in the 56th TU and we don't have 50 R3s then we go back to the previous state.

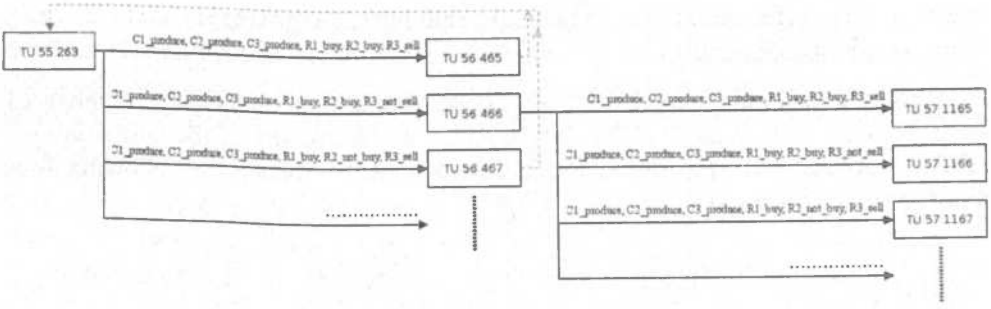


Figure 2: Backtrace

- The algorithm initiates a transition only if it is sure that all constraints will be fulfilled after the transition (forward checking). For example, in the 55th TU we do not sell 10 R3s if we have less than 60 pieces.

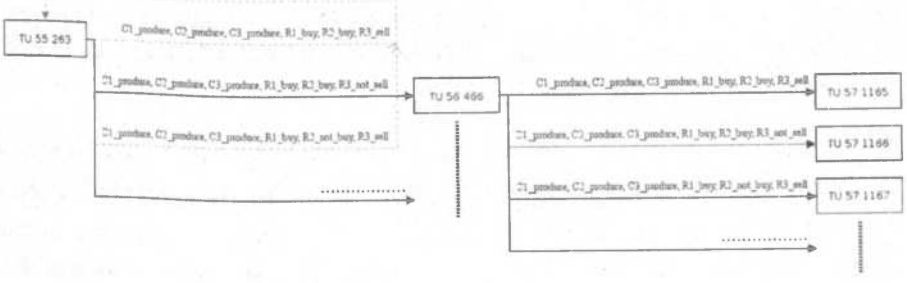


Figure 3: Forward checking

- The algorithm generates new constraints from the available ones and does not initiate a transition that violates any of them (constraint propagation). For example, deducting the need for 2 R3s at the 32nd TU.

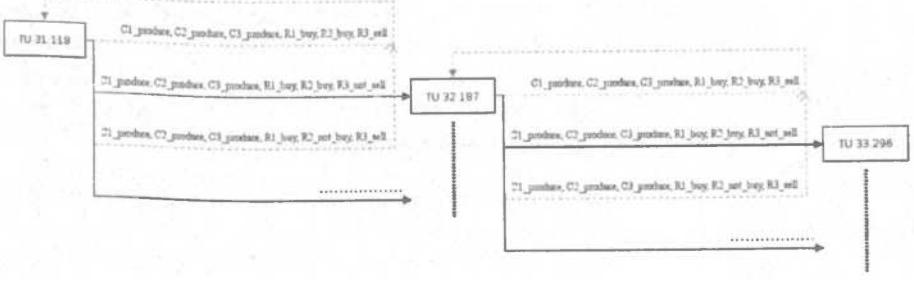


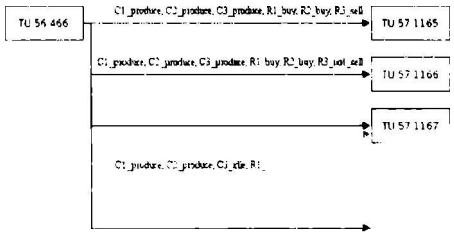
Figure 4: Constraint propagation

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

7. OTHER METHODS TO REDUCE COMPLEXITY

There are two ways of reducing complexity. Non-destructive reductions are reducing the state space by eliminating unnecessary nodes that can not lead to valid solutions. Destructive reductions reduce the state space by eliminating nodes that may or may not lead to valid solutions, thus possibly reducing the quality of the solution found. Constraints, as long as they are correct, are non-destructive. The other two non-destructive reduction methods are remodeling and equality checking. The destructive reductions include applying policies and changing the granularity.

Remodeling the problem decreases the complexity if the new model is simpler than the original. In our example, we may remodel the three identical cells and the four tools as a single cell with 8 tools, thus decreasing the complexity to $O(64^{100})$ In most cases, there are constraints with the same effect, for example, the ITACF constraint introduced in "Solving the given problem"



Equality checking is based on the fact that there may be identical state-pairs in a system from which any chosen transaction results identical states[8], and it is sufficient to keep only one of them and drop the others as they can not lead to a better solution. If a proper value function is given, it is even possible to find state-groups from which only one state is needed to be preserved. In our example, if two states have the same tool configuration, TUs and resources, it is feasible to store only the one with the highest amount of MUs and drop all other inferior states, using a proper value function.

Policies are constraints that are not formally deducible from other constraints and therefore they might disqualify valid solutions. In our example, one such policy is LRS which is described in "Solving the given problem"

Changing the granularity reduces the state space by reducing the domain of state variables. A linear reduction in the proper variable's domain may reduce the state space exponentially. In our example, by reducing the 100 TUs in a shift to 20 TUs, we reduce the complexity from $O(216^{100})$ to $O(216^{20})$, but we also decrease the quality of the solution. Another example could be splitting the problem into two smaller problems, namely the problem of the first 56 TUs and the problem of the last 44 TUs.

[8] There are many other definitions of „identical" not discussed here.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

If the problem is still too complex to be solved by an algorithm in the limited time, there are means to increase the quality of the solution found.

These include random sampling and guided search, which takes many forms from greedy search to A* search. Greedy search always checks the transitions with the most income first, while A* search checks the transition first with the highest value, which is provided by a heuristic function. One such heuristic function may return the total value of stored resources plus the current amount of MUs minus the estimated storage costs until the 100th TU. Random sampling chooses a transition at random, thus – on average – progressing through the state graph evenly.

8. SOLVING THE GIVEN PROBLEM

This particular problem can be solved with a number of tools. One such tool is SCPFW, which is developed by the author and is available at sourceforge.net under GPL. SCPFW is used in the education at BUTE and the given problem can also be, and is solved by SCPFW. The ProductionSytem class in SCPFW is a more abstract problem of manufacturing cells that can be easily customized to implement the given problem. The ProductionSystem has two built-in general constraints and a built-in optimization. The optimization is "Similarity Check" or SC, which drops the states that have less money than their state-pairs, as described in "Other methods to reduce complexity" The first constraint is "Lazy Resource Strategy" or LRS, which disables resource buying as long as the resource can be bought faster than consumed and there is enough resource left. The second is "Identical Tool Action Combinations Filter" or ITACF, which disables transitions that are permutations of other transitions and are identical in their effect. The customized ProductionSystem has a problem-specific constraint, namely "Max R3 Constraint" or MR3, which limits the maximum amount of R3 in the storage. ProductionSystem contains a simple BFS-backtrace algorithm that handles all inner constraints. It tries to execute all transitions from all the N-step states before entering any of the N+1-step states. In case of violation of an inner or outer constraint, a return value is set to false or an exception is generated and the algorithm drops the resulting state. If the transition is executed successfully, the algorithm adds the resulting state to the bank of good states. When all transitions are executed from all N-step states, the algorithm logs the number of good states and starts analyzing the N+1-step states.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

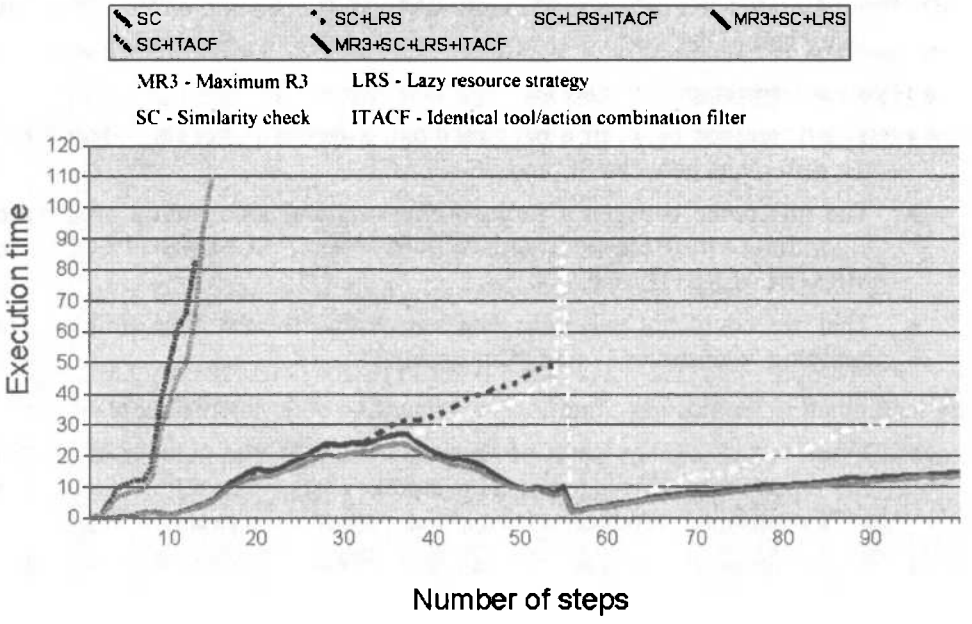


Figure 5: Impact of constraints on execution time

The algorithm was executed on a middle-class multi-user server several times with different constrains enabled and with the maximum execution time of 120 seconds/step. The results are summarized in Figure 5 (some results were dropped to increase readability).

As we expected, the algorithm could not solve the problem without constraints in the available time. At first glance, it may be a surprise to find that the algorithm runs out of memory or execution time in two to five steps only, if we do not apply any non-internal constraints. On second thought, four steps mean about 16 million valid states from the 69 billion possible states. Even with the SC and ITACF constraints enabled, execution time still increases like an exponential curve, but we may reach even step 15. If we add LRS (and thus the maximum R1 and R2 constraints named "Constraint 1" and "Constraint 2" in the paragraph "Constraints"), the problem becomes solvable in the limited time. There is, however, a spike in the execution time at the 56th TU, which is the result of the enormous number of rollbacks caused by not having enough R3s in the store. It even halts SC+LRS, which would need about 140 seconds of execution time for this step. The MR3 constraint (which incorporates "Constraint 3" and "Constraint 4" from the paragraph "Constraints") filters out this spike, reduces execution time

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

from about 40 steps and halves total execution time. It does not, on the other hand, help anything before the 31st TU.

In this particular example, we can see

- The difference between a backtrace and a forward checking algorithm in the difference between SC and SC+ITACF.
- The difference between a forward checking and a constraint propagation algorithm in the difference between SC+LRS+ITACF and MR3+SC+LRS+ITACF.
- That we could not solve the problem within the set time limit without applying "Constraint 1" and "Constraint 2"

By manipulating the starting conditions, we may find other interesting results:

- If we set the storage costs of the resources to 0, the number of possible states increases. This is because the storage costs create a hidden constraint of "Constraint 5: produce at least about 1000 MUs in every 20 TUs"
- By increasing the starting amount of MUs, "Constraint 5" can be weakened or eliminated.
- By decreasing the starting amount of R1, "Constraint 5" and LRS can be weakened.
- Strong constraints (the ones that reduce the number of possible states considerably) often disqualify the same states and thus they are rarely additive.

9. CONCLUSION

Even with the ever-increasing computing power available today, there are problems that still cannot be solved in a human lifetime. These problems have very high algorithmic complexity. Industrial optimization problems (especially problems from the domain of logistics, manufacturing and scheduling) are often such problems. Constraint programming is an effective tool to reduce the complexity of such problems. In this article, a problem is explained and its complexity is analyzed. After adding constraints to the problem, the formerly unsolvable problem is solved by a tool which is developed by the author and is used in the education, thus the complexity reduction effect of constraints is demonstrated. Equalities of constraints and non-constraint methods – in the terms of complexity reduction – are also demonstrated.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

REFERENCES

[1] RSA SECURITY INC, *Has the RSA algorithm been compromised as a result of Bernstein's Paper?*, http://www.rsasecurity.com/rsalabs/node.asp?id=2007

[2] CORMEN, T.H., LEISERSON, C.E., RIVEST, R.L.: *Introduction to Algorithms*, MIT Press, 1990

[3] PER J. KRAULIS: *Algorithmic complexity*, http://www.sbc.su.se/~per/molbioinfo2001/multali-algocomplex.html

[4] MICHAEL L.: *Algorithms & Data Structures , Sorting Algorithms*, http://linux.wku.edu/~lamonml/algor/sort/sort.html

[5] HOMER COMPUTER SERVICES PTY LTD, *Glossary of terms*, http://www.homercomputer.com.au/homer_software_guide/glossary.htm

[6] BLACK, P.E.: *Breadth-first search*, http://www.nist.gov/dads/HTML/breadthfirst.html

[7] KRZYSZTOF A.: *Principles of Constraint Programming*, Cambridge University Press, 2003

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 121-129

BEHAVIORAL ROBOT SIMULATION FOR PRODUCTION SYSTEMS

TAMÁS JUHÁSZ
Department of Control Engineering and Information Technology,
Budapest University of Technology and Economics,
Mobile and Microrobotics Laboratory,
Pázmány Péter sétány 1/d, I.B.411., Budapest, H-1117, Hungary
tjuhasz@seeger.iit.bme.hu

[Received November 2004 and accepted January 2005]

Abstract. The intelligent production systems have become more and more important for modern enterprises in the industry. Usually the design process of a new product has great many stages: each of them can introduce errors that are seriously retarding the heavy schedule. During the development of a new product a prototype is created usually in early design phase. This prototype can be used for verification purposes to hinder the unwanted events owing to design mistakes appearing in the final (commercial) product. Sometimes the manufacturing of the prototype can be a risky, expensive and time-consuming process – especially if the result does not meet our strict demands (thus we have to create a new prototype again). It is a modern approach to use virtual prototyping (by incorporating virtual reality) to cut down costs and increase productivity. This paper deals with this complex simulation and modeling problem through presenting our current simulator project at the department. Using our new kind of simulation architecture we can introduce physical hardwares to these tests to give better compliance with the behavior of the final product. The objectives and the future steps in the development of the software will also be discussed in this paper.

Keywords: Virtual reality, virtual prototyping, hardware-in-the-loop testing

1. INTRODUCTION

Virtual prototyping enables continuous iterative design and testing via simulation, allowing the developer of a complex manufacturing system to find errors earlier in the design phase. Today there is a wide variety of simulators on the market, but most of them are trade-, model- or application specific ones. A summary was created by Yiannis Gatsoulis (University of Leeds) who was claimed by the CLAWAR community to analyse the state-of-the-art of this field a few years ago [1]. It contains a snapshot of the available softwares (environment editors, image processing and control libraries, system simulators, etc.) that are in connection with

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

this research area. It assesses the cost, usability, expandability and rapid development abilities of the given applications.
In general the ideas and techniques developed during the simulation process yield to ideal conditions to raise synergy: a catalytic effect for discovering new and simpler solutions to traditionally complex problems. Using virtual prototyping by means of behavioral simulation reduces the time-to-market, as it shows the inconsistencies early in design phase. Abstract modeling with CAD tools, enhanced conceptual design and moving up life-cycle assessments by virtual prototypes allow devising the optimal layout and the best mechanical architecture of the system.

2. THE MICROSOFT .NET FRAMEWORK

The .NET platform (Figure 1) offers rapid application development using the new C# language [5] and its event-delegate based communication model. The .NET framework is a new development framework with a new programming interface to operating system services and APIs, integrating a number of technologies that emerged during the late 1990s [2]. Incorporated into .NET are COM+ component services; the ASP web development framework; a commitment to XML and object-oriented design; support for new web services protocols such as SOAP, WSDL, and UDDI; and a focus on the Internet.

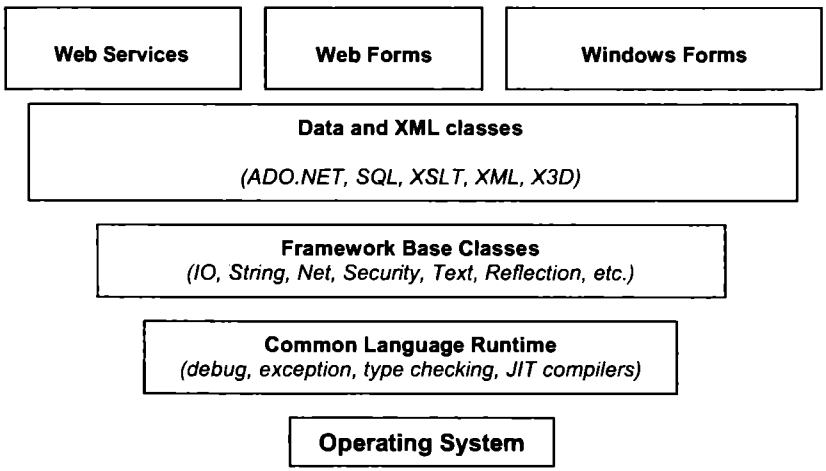


Figure 1: The .NET platform

Aside from embracing the Web, Microsoft .NET acknowledges and responds to the distributed computing and componentization trends within the software industry today:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

- Distributed computing:
  Simplifies the development of robust client/server or Web-based applications. Current distributed technologies require high vendor-affinity and lack interoperation with the Web. Microsoft .NET provides a Remoting architecture that exploits open Internet standards, including the Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML), and Simple Object Access Protocol (SOAP).

- Componentization:
  Simplifies the integration of software components developed by different vendors. The Component Object Model (COM) has brought reality to software plug-and-play, but COM component development and deployment are too complex. Microsoft .NET provides a simpler way to build and organize components.

The so called .NET Remoting enables easy communication for individual applications. It uses open Internet standards (HTTP, XML, and SOAP) at its core to transmit an object from one machine to another across the Internet. In fact, there is bidirectional mapping between XML and objects in .NET. For example, a class can be expressed as an XML Schema Definition (XSD); an object can be converted to and from an XML buffer; a method can be specified using an XML format called Web Services Description Language (WSDL); and an invocation (method call) can be expressed using an XML format called SOAP.
Another important feature that makes the .NET platform and the C# language suited for creating a powerful simulation application is the ability to use native modules that are exploiting the hardware through embedded operating system devices. For example we can use Universal Serial Bus (USB) drivers to communicate with external hardwares in case of hardware-in-the-loop testing (discussed later, in the section 3.5 of this paper) in a straightforward way.

3. THE GLBOT.NET SIMULATOR PROJECT

Cooperating mobile robots are commonly used in modern manufacturing environments. They usually do logistics tasks by carrying payload such as assembly parts, tools for service robots and so on. The functionally integrated mobile platforms play a key role in the manufacturing procedure as their efficiency gives a significant part of the overall performance concerning the entire production cell. Thus realistic simulation of their behavior is basically important during the planning of the target process.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Now there is an ongoing simulator project [3, 4] (called: GLBot.NET) at the Department of Control Engineering and Information Technology in Budapest University of Technology and Economics. Our component-based simulator is being written in the new generation C# language [5]. It is utilizing the platform-independent .NET framework and the X3D environment that is a standard 3D scene description language introduced by the Web3D Consortium [6].

3.1. Our objectives

In these days the key objectives for virtual prototyping systems are modularity (concerning all devices and aspects) and interoperability (using the output from- or serve input to other related applications). We are trying to give general solutions for the upcoming problems staying apart from model- or application-specific constraints. One of our main objectives is to develop a simulator that maintains testing of individual components by means of hardware devices as well (so to support hardware-in-the-loop testing). The main components of the simulated system are proposed to be connected through a standard interface and designed such way that they could be replaced by a corresponding physical hardware.
An obvious effort has to be taken to use standard data formats for describing the manufacturing environment (as well as the mobile robots being embedded in it), which makes easier to separate the notional design from other manufacturing steps. The new X3D standard is a powerful and extensible open file format for 3D visual effects, behavioral modeling and interaction. It can be considered as a successor of the well-known VRML format. By providing an XML-encoded scene graph and a language-neutral Scene Authorizing Interface, it makes scene verification much easier and allows 3D content to be easily integrated into a broad range of applications. Its base XML language lets incorporating 3D into distributed environments, and facilitates moving 3D data between X3D-aware softwares.
The Extensible Modeling and Simulation Framework (XMSF) is defined as a set of Web-based technologies, applied within an extensible framework, that enables a new generation of modeling and simulation (M&S) applications to emerge, develop and incorporate [7]. Distributed simulation through XMSF makes possible the efficient testing of cooperating platforms which are commonly used in modern applications. The XMSF will enable simulations to interact directly over a highly distributed network – which can be achieved through compatibility with Web technologies – and will be equally usable by humans and software agents. XMSF must therefore support composable, reusable model components. The Extensible Markup Language (XML) is the cross-cutting technology for root data structure representations, with Resource Description Framework and ontology-tagset support for semantics. Some of the primary challenges for the XMSF are: providing open and extensible M&S capabilities, improving speed of development by stimulating

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

rapid growth of interoperable simulations and providing support for all types and domains of M&S (constructive, live, virtual and analytical).

3.2. Modularity in GLBot.NET

A modular mobile robot system has generally three major components (sensing, processing and locomotion) that are communicating with each other. All mobile robots use locomotion that generates traction, negotiates terrain and carries payload. Some robotic locomotion also stabilizes the robot's frame, smoothes the motion of sensors and accommodates the deployment and manipulation of work tools. The locomotion system is the physical interface between the robot and its environment. It is the means by it reacts to gravitational, inertial and work loads. Thus the locomotion system is the basis of a mobile robot's performance.
Each main robot component (sensing, processing and locomotion) can be considered as a separate module in the application. These modules are connected through a standard communication interface (Figure 2).

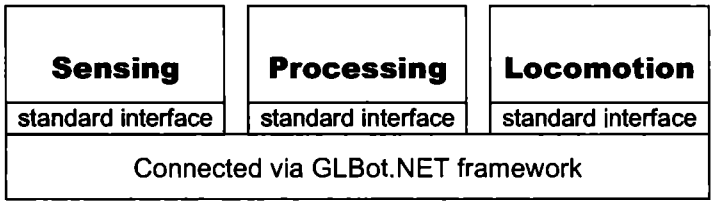


Figure 2: The modular simulator architecture

Telerobots, teleoperators, and remotely operated vehicles [8] belong to a class of machines used to accomplish a task remotely, without the need for presence on site. The modular architecture of the simulator, the .NET Remoting and the standard communication interface between these modules involve that the given sensing and locomotion components can be far away from the intelligent control component (running on different computers that are connected via a communication channel). Thus remote operation can be carried out in GLBot.NET.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

3.3. The user interface of the simulator

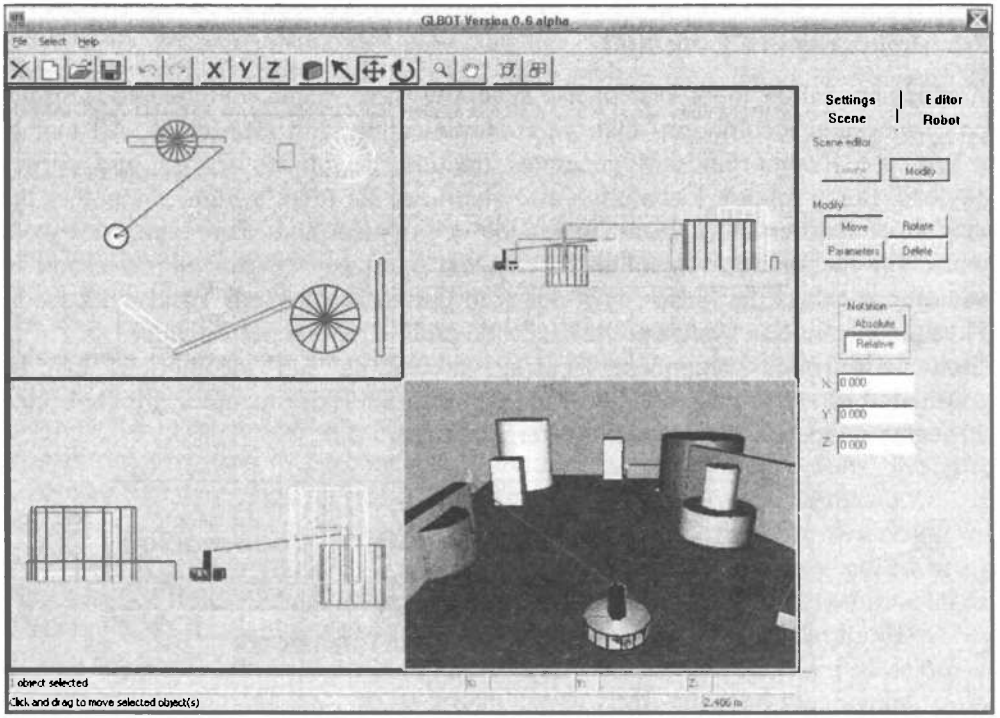


Figure 3: The user interface

Figure 3 shows a snapshot of the main screen of the application which is under construction at the moment. The example differential-driven mobile robot is equipped with a laser range finder and it is measuring the distance of a known marker using that device.

The program uses a CAD interface that is similar to the well-known 3DStudio MAX® software (proprietary of Discreet™) to let you easily design the desired virtual environment and mobile platforms. Many primitive object types are available (box, plane, cylinder, cone and sphere) and general 3D mesh objects can be imported from standard X3D files. All 3D objects (even cameras and lights) are nodes in a tree structure that are handled and stored hierarchically in the X3D scene description file. Thus one object can be a parent of another, where each object passes its transformation to its children. By construction all objects are the child of an unyielding root object (the one that has no parent at all), and they can be re-linked to their new parent (with the Link tool) as desired.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

The nodes that are representing the robot platforms themselves do not differ much from other general nodes. You can mark objects (3D objects, joints, cameras, etc.) as a part of a mobile platform in the Robot Construction Dialog (located in the right Control Panel). Generally there is a base chassis object that is the parent of the driving wheels, onboard cameras, etc. in each platform. Thus these parts can be manipulated independently within the user-defined simulation program.
After finishing up with the environment construction, you can switch to simulation mode ("Editor" on the right hand side Control Panel) on the Control Panel. Here you can load, edit and execute your own high-level program (a .NET library) that will control the robot. Of course the data of the onboard sensors (CCD cameras, sonars, tilt sensors, etc.) are at your disposal during the simulation.

3.4. The virtual driving subsystem

The simulator will offer the following kinds of transmission models:

Differential drive with caster
Ackermann steering
Synchronous drive
Tricycle
Omni-directional (a.k.a. Swedish wheel)

The user can manually select and configure the desired driving subsystem in the previously mentioned Robot Construction Dialog. The simulator incorporates the dynamic model of these platforms to behave the same way as a physical platform would do.

3.5. Hardware in the loop testing

Expensive or unique systems are generally hard to test. Hardware-in-the-loop testing lets you build extremely realistic visualization tools by inserting physical hardware in the testing loop.
If we use standard communication interfaces between the implemented modules (Figure 2), then the core framework can treat virtual and physical components as an equal. By using hardware implementation in a given module, we can talk about hardware-in-the-loop testing which is an important application in a virtual prototyping manufacturing environment.
Let's assume the following scenario: we have a proper virtual representation of the real scene where our real robot will be operating (Figure 4).

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Figure 4: A well reproduced indoor environment

We place this physical robot into an empty room with phantom obstacles (e.g.: sketch drawings on the floor: to avoid any unpredicted collisions) and control it according to the virtual environment's visual information. Thus we can verify our model being used whether the physical platform reacts the same way as the virtual one upon the same command sequence (for example a new navigation algorithm). If we got satisfactory results we can take our platform out of the room and put it to the real situation.

4. CONCLUSIONS AND FUTURE PLANS

Using virtual prototyping and hardware-in-the-loop testing in a modern production system can reduce development costs and time-to-market. To achieve this we have to use realistic, behavioral simulation. The incorporation of the dynamic model of the cooperating mobile platforms is needed to realistic response to virtual forces and torques. Large manufacturing environments ask for distributed simulation that can be accomplished with the conventions of the Extensible Modeling and Simulation Framework [7]. These are the most important tasks for the forthcoming development.

ACKNOWLEDGEMENTS

This research is supported by the Hungarian Scientific Research Fund (OTKA) grant No.: T-042634-OTKA, and the CLAWAR (CLimbing And WAlking Robots) Research Training Network Mobile Robotic Demonstrators and Applications: G1RT-CT-2002-05080.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

REFERENCES

[1] CLAWAR COMMUNITY: *Summary for WP2: Simulators*

[2] THAI T., LAM H. Q.: *.NET framework essentials*, 1st ed., 2001, ISBN 0-596-00165-7

[3] JUHASZ, T.: *OpenGL powered mobile robot simulator supporting research on landmark-based positioning*, Proceedings of MicroCAD'03 Conference, 2003, Vol. N., pp. 85-91, ISBN 9-636-61560-8

[4] JUHASZ, T.: *Graphics acceleration techniques for a mobile robot simulator*, Proceedings of CESCG'03: Central European Seminar on Computer Graphics, Section 5: Computer Vision, 22-24th April, 2003, Budmerice, Slovakia

[5] O'REILLY & ASSOCIATES: *Programming C#*, May, 2003 ISBN 0-596-00489-3

[6] WEB3D CONSORTIUM: *X3D overview*, http://www.web3d.org/x3d/overview.html

[7] XMSF HOMEPAGE: *Extensible Modeling and Simulation Framework*, http://www.movesinstitute.org/xmsf/xmsf.html

[8] JUHASZ, T.: *Surveying telerobotics and identification of dynamic model parameters*, Proceedings of MicroCAD'04 Conference, 2004, Vol. K., pp. 211-216 (in Hungarian), ISBN 9-636-61619-1

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 131-142

HIAC
HIERARCHICAL INTER-AGENT COMMUNICATION SYSTEM

FERENC VAJDA
Mobile and Microrobotics Laboratory
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
vajda@iit.bme.hu

[Received November 2004 and accepted January 2005]

Abstract. In our century, there seems to be more and more demand of multi-agent heterogeneous robotic systems whose agents together can fulfill tasks that had been very difficult to apply and required deep programming knowledge earlier. The most difficult task about heterogeneous systems is perhaps to implement the right communication between the devices. Although IP-based communication can hugely help to solve this problem, it is not really applicable in many respects for a robotic system whose structural, hierarchical architecture makes the construction of a special communication system expedient. The main aim of our article is to give a recommendation to the communication of the systems that have a structured device set of huge number of entities being in close relation to each other.

Keywords: Communication, Protocol, Hierarchy, Control, Multi-agent Robotics

1. INTRODUCTION

HIAC [1] is a communication stack of on-line systems. So first of all HIAC provides tools with a relative small amount of information for higher level communication. However, transmitting larger amounts are also possible, but this was of smaller moment in the course of development. HIAC supports the modern communication set of processes (or independent agents in HIAC) on higher levels. The easy implementability was one of the most important standpoints, so it is possible to implement HIAC on devices to which implementing more complicated protocols would be very difficult or impossible. This ability must not lead to smaller number of features if communication is led by devices with larger computation capability. It is important to note that the HIAC is suitable for handling the communication of systems that fulfill a common task or tasks, and not for making a huge number of operations that are totally independent of each other – as IP (of Internet) [6] does.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

This article is a short outline of the HIAC protocol stack, and it tries to introduce the main components of this communication system. We tried to point out some significant aspects in a more detailed way to show the importance of the development of the stack.

Testing the communication system is now under process in the frame of the Balaro [3] project. The hierarchy of the system is given by the architecture of the system.

The HIAC is still under development, so some questions will be able to get answered by the time the stack has been completely developed. Such questions are the enhanced gateway functionality of the participant devices, the more secure data transfer, or the inter-agent communication of mobile robotic platforms, etc.

1.1. Relation to the ISO-OSI model

HIAC covers multiple layers in the ISO-OSI model [7]. First of all HIAC is a set of protocols built over each other, moreover the higher level protocol covers the tasks of multiple layers. The two protocols are EDCP (Escape Driven Connection Protocol) and HIACP (Hierarchical Inter-Agent Communication Protocol) which does the substantive work. HIACP is built over EDCP, which supposes a two-way on-line connection between the devices. The connection may be indirect, so EDCP can operate over an open TCP port, RS232 channel, or any continuous open connected channel. The next figure shows the location of HIAC in the ISO-OSI model.

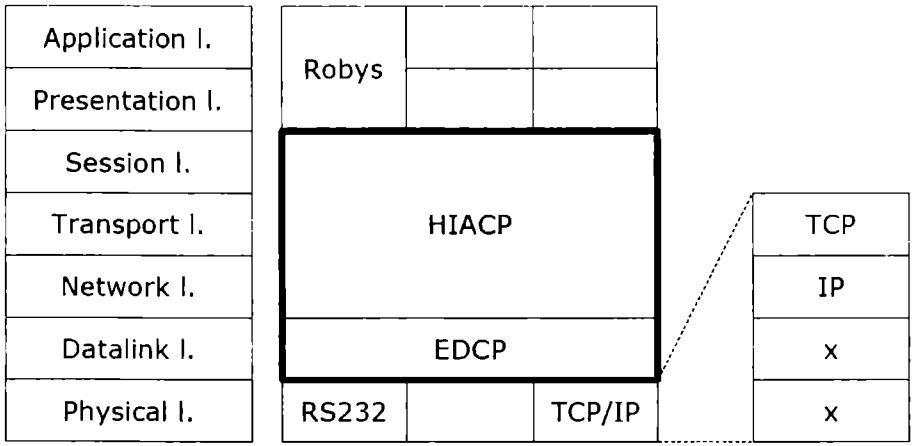


Figure 1: Location of HIAC in the ISO-OSI model

There are no restrictions for higher levels, but it is important to note that HIAC is in a close connection with Robys [2] – hierarchical, parallel, object- and task-

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

oriented [4] robot-programming – system, which is under development. Robys will be suitable for heterogeneous robot systems or even for distributed parallel systems.

2. EDCP

EDCP is a rather simple protocol, which helps the HIACP to separate packets and to monitor the continuous connection, etc. EDCP represents the datalink layer of the HIAC Protocol stack, so it realizes the point-to-point connection between two "direct" connected devices. This datalink layer is simpler than that of other systems (it contains not even a checksum), because it assumes quite a safe data-transfer (the layer beyond and the higher levels of HIAC take care of it).

The octet (or byte) based protocol is escape-driven. This means that some preset – so called escape – characters (EDCP codes) have special objectives while the other octets get through the protocol transparently. If we would like to transfer octets that have special meanings in the protocol, they will be transfered by the help of a distinguished EDCP code. Some EDCP codes require further information that are located after the code as additional octets.

2.1. EDCP codes

The 00-F7 range of octets makes the movement of the characters of the corresponding value that should be transfered in a higher level, and the F8-FF range is reserved for EDCP codes. SPKG (0xF8) starts and EPKG (0xF9) closes each packet. This makes the data transfer robust, because this way we know the start and the end of the packet, there cannot happen any misinterpretation or overrun, etc. as the consequence of lost data. The checking whether all the data of the packet has arrived without failure is done by the higher layers. If we want to transfer F8-FF code corresponding data in a higher level we have to come up with a special solution. In these cases the data transfer is helped by the EXCH (0xFC) code. The least significant 3 bits of the octet after the code determines which F8-FF character will be transfered. The value of the higher 5 bits: 11001. If the octet after EXCH is different from this, EXCH will be ignored.

2.2. Connection Monitoring

The EDCP does not give recommendation on building up or cutting off the connection; however, since it assumes continuous connection, this necessitates certain complementation. Since it may occur that a device is not sending data for a longer time, the communication partner does not know if the connection has broken or not. The EDCP specifies the maximal period of time within which the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

partner has to send any data. If there is no information to give they can also send an empty packet (SPKG+EPKG).

2.3. Overhead

Similarly to other protocols the EDCP has a certain amount of overhead. For huge packets with data that have even statistical distribution (e.g. longer messages) its value is around 4-5%, while for small packets (e.g. signals) it may be 20-25%, and in certain cases even much bigger then these. In practice, by choosing the suitable communication parameters of the higher level HIACP, the overhead can be held around 20% for the shortest packets as well. Using F8-FF coded characters is also avoidable for longer messages in most cases.

3. HIACP

The HIACP is a much more compound protocol. Besides the fact that it supports complex operations regarding a structural hierarchy, it provides minimal implementation in simple devices. We can send signals and messages, we can use shared memories, semaphores or synchronization through the protocol independently of hierarchical levels.

3.1. Structure

HIACP is a packet based protocol. This means, that each device of the system provides their information, commands, etc. in a packet to the partner device. We can use various types of packets for the different tasks. These packets are similar to each other in several aspects, but their content carries different aims.

Generally, during communication the devices have a direct contact with several other devices in a hierarchical arrangement. The possible target and the content of the messages depend on the level and the location in hierarchy.

Since it is not sure that the two devices that want to communicate with each other have direct contact, certain devices must have a gateway function, as well. Some devices can interpret the protocol in a higher while others in a lower level. Therefore, it may happen that the devices without a direct contact will try to communicate with each other in different way. (The devices in direct contact are equal in the set that both of them supports, thus their communication is not problematic). The task of the gateway is not only to transfer message packets, but also to change the packet itself based on the suitable communicational abilities. These changeable parameters include the type of checksum, response to the message, etc.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Each device has its own unique identifier, which determines its position and role in hierarchy. This identifier is the HIAC address. This is completed by an identifier (HIAC mask) by which we can specify with whom it can communicate, which is its workgroup, etc.

Despite other communication protocols we can surely assume that the devices have a lot of a priori information. In the case of other systems the abilities of each system element appears only in a higher level, while in the HIAC we should know the roles and abilities even in a lower level. In the initialization phase the devices share their important communication parameters, but there are several data which contain no relevant information for the communication partner. These information does not change too often in systems where using HIAC is expedient.

3.2. Hierarchy

One of the biggest advantages of the HIAC is the hierarchical structure. Thus, each device can communicate only with partners meeting its hierarchical level and position, and this way we can more easily supervise and handle the controlling of a system.

Based on hierarchical levels we can distinguish three communication directions: the *employer*, the *employee* and the *colleague* communication. Although the employer and the employee communication has a close relation, we have to make difference between them, since an employer can say different things to its employee, than vice versa.

3.2.1. Employer Communication

Under the name of Employer Communication we mean the communication with the higher ranked device that is directly above the other device. Although, in the course of employer communication certain regulations have to be observed when talking to a higher ranked device, most of the communication methods are open. The communication has to be initialized by the employer in any case (INCON – initialize connection). It cannot be addressed, until it initializes the connection. It is also not possible to give certain synchronization commands. Although, it is not regulated in the specification, when making the system plan we should not implement commands that are given by an employee to its employer.

The device cannot talk to its employer's employers. It can solve this situation only by „asking" its employer to „settle" the thing with its own employer.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

3.2.2. Colleague Communication

Under the name of Colleague Communication we mean the communication with devices of the same rank, in the same workgroup. Here we have more than one possibilities: any of the two parties can initialize the connection, and use the synchronization. The HIAC specification does not give smaller rights in the course of colleague communication than employer communication. However, it is important to take care when designing the higher level system that the colleagues cannot give commands to each other only information. There may be certain warnings that can be sent to each other.

3.2.3. Employee Communication

Under the name of Employee Communication we mean the communication with the lower ranked device that is directly under the other device. This communication direction includes the widest possibilities set of communication types.

In the course of this we can use any forms of communication, including that we can insert the lower ranked device into the network, remove it from there or we can pass it over another device.

3.2.4. Subemployee Communication

Subemployee communication is a special instance of employee communication. Under the name of Subemployee Communication we mean the communication with the lower ranked device that is *not* directly under the other device. At present this is a one-way communication: the device may give commands to which it does not expect a response, to the subemployee. If it wants to directly control the device it can take away the subemployee from his own employee, but if so, the subemployee's original employer cannot give commands to it.

Subemployee communication can be important especially in systems using mediation hierarchy [5], which means that users may intervene on all the hierarchical levels (from high-level control down to the low-level physical layers) of a robotic system.

3.3. Packet Structure

The HIACP is built up of packets. The structure of the packets is the following:

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

	7	6	5	4	3	2	1	0
1	PV				PT			
2	-	PRI			CR	MP	CT	FA
3		PSN (*byte* / none)						
4	TRG (*dword* / *adaptive*)							
5	SRC (*dword* / *adaptive*)							
6	LNG (*adaptive*)							
7	DATA (...)							
8	CHK (*byte* / *dword*)							

Figure 2: HIAC packet architecture

PV (Packet Version) – version number, at present always '1'
PT (Packet Type) – type of packet, see more in 3.5
PRI (Priority) – priority of packet, lower values mean higher priority
CR (Confirmation Required) – confirmation is required if value is 1
MP (Multiple Packet) – multiple packet mode, numbering packets, fragmentation
CT (Check Type) – type of checking, simple sum or CRC32
FA (Full Addressing) – full addressing is used, see more in 3.4
PSN (Package Serial Number) – automatically increasing serial number of packets
TRG (Target) – target's address
SRC (Source) – source's address
LNG (Length) – number of data octets
DATA (Data) – any information, depending on packet type
CHK (Checksum) – checksum

3.4. Addressing

Every device of the HIAC system has a unique identifier, the HIAC address. The HIAC address is a 32 bit identifier, whose value depends on its position and role in the system. The HIAC system has a hierarchical structure, which means that each device's role in the system determines with whom and in what level it can maintain connection. Since this connection belongs to a subnetwork part, it is enough to specify one subnetwork identifier, level address in the course of the addressing process. This identifier is part of the HIAC address, in other words the level address is the value which is masked by a so called HIAC Mask. We have to define the HIAC address and the HIAC mask by dividing these values into octets. These octets are represented by two hexadecimal digits, and they are separated by ':' characters from each other.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

E.g. If the HIAC address of a device is 3C:12:F8:00 and the value of the HIAC mask is 00:0F:FF:00, the level address is 2F8.

Every device gets its HIAC address and its HIAC mask from its employer. Thus, it will know with which devices it can communicate. In a certain subnetwork, the devices above certain other devices have always 0 level address and every other value implies the addresses of devices of the same rank. Besides the 0 level address of the subnetwork, its highest valued level address also has a special role. This is a so called broadcast message, which has to be interpreted by every participant as if it had been meant directly to them.

The address distribution also follows the hierarchical structure of the system (see fig. 3). This means that an employee may only have such an address which is derived from the employer, i.e. the employees constitute one ore more subnets of the employer. For example, if an employer's address is 20:C4:D2:00 (e.g. with mask 00:00:3F:00) employees' addresses can be 20:C4:D2:01-20:C4:D2:FE (e.g. with mask 00:00:00:FF). 20:C4:D2:FF is the broadcast address of the subnet.

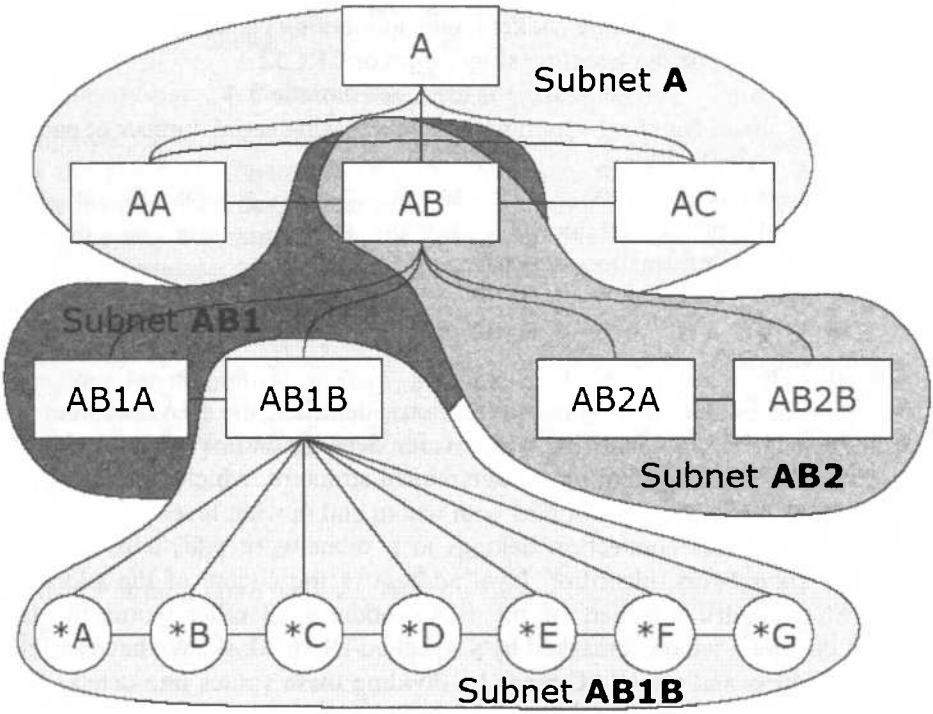


Figure 3: Hierarchical architecture of HIAC addressing

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In the current version of HIAC employers are "directly" connected with their employees (it may also be a TCP/IP connection). Colleagues are normally not connected, so if two colleagues want to communicate with each other, they send their messages through the employer. Note that this simple gateway functionality of the employer is a "low level" task, so the employer does not care about the message.

3.5. Packet Types

The content of the data field depends on the packet type in any case. Within this article we cannot give a detailed description of each type and the information transfered by them. However the following table briefly summarizes the available types.

Table 1: Packet types

TYPE	*Name*		*Description*
1	SIG	signal	sending quick warnings
2	SMPH	semaphore	testing for free resources
3	MSG	message	sending commands, appeals, information
4	SHMW	writing shared memory	storing general/common information
5	SHMR	reading shared memory	
8	SYNC	synchronization	synchronization, parallelization
14	SYS	system packet	system specific data transfer

3.5.1. System packets

System packets fulfill the fairly important tasks which are necessary for operation of HIAC, e.g. passing over data transfer parameters, distributing hierarchical arrangement, gateway functions, etc. We have to specify a subtype within the type of the system packet – this subtype will determine the exact task. The following subtypes are the most important ones:

Package Confirmation (PKGCNF): Response to a packet. May refuse, indicate failures or may accept.

Initialize Connection (INCON): The communication initializing partner passes over the communication parameters supported by it, and the HIAC data (address, mask) of the target device.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Accept Connection (ACCON): The communication receiver partner passes over the communication parameters supported by it, indicating that it is ready to communicate.

Refuse Connection (REFCON): If for some reason the device is not ready for communication (e.g. The employer wants to establish an employer-employee connection, but the employee has already an employer)

Reinitialize Connection (RECON): Necessary if the position of the device in the hierarchical system changes; therefore, its HIAC data and even the abilities of the communication partner change.

Employee Transfer (EMPTRF): If an employer does not need its employee it may pass it over to another device. Passing over is indicated by EMPTRF.

Workgroup Transfer (WKGTRF): Similar to EMPTRF, but a whole workgroup can be passed over.

Employee Request (EMPREQ): Request for an employee. An employer may instruct an employee to pass over one of its own employees. This can help if a device starts to have defective behavior. Its workgroup(s) may work onwards. (If there is a greater fault – and the device is out of order –, directly calling the subemployee is also possible, but this is not the question of EMPREQ.)

Workgroup Request (WKGREQ): Similar to EMPTRF, but a whole workgroup can be asked for.

4. APPLICATIONS

HIAC was developed mainly for robotics related applications, i.e. multi-agent heterogeneous robotic systems; however, it is also possible to use the protocol stack in other areas using hierarchical architecture of coupled devices, applications, etc.

The first system on which this communication system is tested is the Balaro [3] project. The Balaro (Balaton Rover) is being developed for mapping the basin of Hungary's largest lake, the Balaton, in order to find and pull out dangerous objects, relics from the second world war, etc. There is a simple hierarchy in this system, but it contains all the possible communication types, so it is ideal for testing and using HIAC. The Balaro system comprises a mobile platform (later there will be more of them) wandering on the ice cover of the lake and a station on the coast. The highest level of this hierarchy is the coast-station, all the lower levels are built in the mobile platform itself. The mobile platform has a control unit, which is responsible for the proper operation of Balaro, and some lower level intelligent

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

devices: the sensor system, the GPS with track-recording, the driving, etc. The devices are connected through TCP/IP connection (wireless with the coast station). The coast-station is the employer of the control unit, which is the employer of the lower level devices, so the devices are the subemployees of the coast station. The devices are colleagues of each other. The employer, employee and subemployee communication need no further explanation. Communication between colleagues is also necessary, e.g. the sensor system can "control" the driving directly in case of danger.

Other applications are in view in the field of microrobotics and other multiagent mobile systems, and also in the field of robotics on the game-market.

5. FUTURE PROSPECTS

Working out the specification we did not pursue to create an all-comprehensive system. There are several further development possibilities that can be integrated in later versions of HIAC. Some of the development prospects are the followings:

- *Secure transfer*
  A data transfer is secure if unauthorized devices cannot access data and even they cannot send data in the name of a partner in the system. SSL or other public-key encryption systems can be taken into account. If secure data transfer is needed now EDCP can also work over an encrypted connection.
- *Gateway function*
  Basic gateway functionality is already given in HIAC, but it needs several further supplements. If a device receives a message and it knows where to forward it (i.e. there is a direct connection with the target), it must forward it. If it does not know the target device, it has to forward it to its direct employee. This way of gateway functionality is mostly enough in communication with direct connection. In some cases this is not so. If using a system where the system's devices are not in a hierarchical arrangement (e.g. token ring), or a mobile system, where devices are moving continuously, the gateway theory must be expanded.
- *Moving systems*
  In mobile robot systems, often the devices cannot communicate on-line with each other, only when they meet somehow. Answering these questions remains to the later versions of HIAC

ACKNOWLEDGMENTS

The author gratefully acknowledges the contributions of the Hungarian Scientific Research Fund (OTKA T 042634).

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

REFERENCES

[1] VAJDA, F.: *$HIAC_{VI}$ Specifikáció ($HIAC_{VI}$ Specification)*, MoMic, dept. of CEIT, BUTE, 2004

[2] VAJDA, F.: *Robys – a Hierarchical Robot Programming System (Developed Mainly for Multiagent Microrobotic Environments)*, CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby, Vienna, 2004

[3] VOGEL, M: *Balaro: Semi-autonomous Ice Rover For Localization of Unknown Objects in Frozen Lakes*, CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby, Vienna, 2004

[4] VAJDA, F., URBANCSEK, T.: *High-Level Object-Oriented Program Language for Mobile Microrobot Control*, Proc. of the INES 2003 International IEEE Conference on Intelligent Engineering Systems, Assiut-Luxor, Egyiptom, 2003

[5] ADAMS, J. A.: *Human management of a hierarchical system for the control of multiple mobile robots*, Dissertation, University of Pennsylvania, 1995

[6] POSTEL, J.: *Internet Protocol – DARPA Internet Program Protocol Specification*, RFC 791, USC/Information Sciences Institute, 1981

[7] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 7498-1:1994 Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*, Ed. 2, 1994

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 143-158

SCALABLE ADDRESSING AND ROUTING IN LARGE-SCALE WIRELESS NETWORKS

GERGELY BICZÓK
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
biczok@tmit.bme.hu

NORBERT ÉGI
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
egi@tmit.bme.hu

PÉTER FODOR
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
fodorp@tmit.bme.hu

BALÁZS KOVÁCS
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
kovacsb@tmit.bme.hu

ROLLAND VIDA
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
vida@tmit.bme.hu

[Received December 2004 and accepted February 2005]

Abstract. One of the main characteristics of future networks will be the considerable increase in the number of communicating entities; PCs and laptops, but also PDAs, 3G phones, sensors, wearable devices, etc., will all connect to and communicate with each other, to form large-scale intelligent networks, integrating different technologies. On the other hand, there will be a considerable shift from fixed, wired devices to mobile, wireless ones. In these conditions, the current IP based mechanisms are foreseen not to fit these new network structures. Therefore, alternative addressing and routing solutions are needed to handle these large and highly mobile, dynamic, wireless networks. In this paper we propose a new, scalable approach to the addressing and routing problems that emerge in the context of the envisioned future network architecture.

Keywords: stateless routing, addressing, mobility, scalability, wireless communication

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

1. INTRODUCTION

Researchers in the field of networking have to continuously monitor the current trends related to the evolution of technologies, the changing user requirements or the emergence of new services, in order to design efficient new solutions that take into account these tendencies. In some cases, small modifications of existing approaches are enough to adapt to the evolving requirements. On the other hand, from time to time radical paradigm changes are needed. Currently there are several factors that indicate that such a radical change is inevitable, if we are to cope with the foreseen evolution of future networks.

One of the most important issues that will have to be handled in these networks is the significant increase of the number of communicating entities. We do not refer here to traditional, fix-installed nodes, but rather to new generation cell phones, wirelessly communicating notebooks, PDAs, personal identification cards, sensors, wearable devices etc. Integrating all these entities into a common networking infrastructure will generate serious scalability concerns regarding the current IP addressing mechanism and the routing schemes that are based on it.

Besides these scalability concerns that are due to the increasing number of communicating devices, it is important to note the associated shift of the ratio of mobile and fixed entities that will be involved in future network architectures. The main part of today's Internet is composed of fixed, wired devices, while wireless, mobile entities appear only at the periphery of the network. Current technologies, such as Mobile IP for example, are able to adequately handle small-scale mobility that appears at the periphery; nevertheless, they are not able to provide scalable and efficient mobility and addressing support, should the before-mentioned radical ratio change become a reality.

On the other hand, if mobility appears not only at the periphery of the network, a re-thinking of routing mechanisms will be needed. The current routing protocols were mainly designed to highly static, wired environments, where communication paths are broken rarely, and mainly due to errors, such as node failures or congestion. Therefore, they react very slowly to network changes, with long convergence times. As opposed to this, in a dynamic environment, where mobile hosts might reach the network through other mobile nodes and communication paths have to be continuously reshaped due to the enhanced mobility of the participants, it is hard to see these routing protocols performing well.

In order to underline these tendencies, let us provide now some concrete examples of future applications and communication environments. On the one hand, let us consider the case of a shopping mall, or an airport, where there are thousand of people that might have tens of thousands of wireless electronic devices able to communicate with each other. These devices, together with embedded sensors or other intelligent equipments, will group together in large and highly dynamic,

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

mobile ad-hoc networks. Similarly, let us consider the case of a highway, with sensors embedded in the road or the traffic signs, and with cars equipped with different wireless communication devices. In such an environment, a road sensor can alert the driver of a car about icy road conditions, information which can be then relayed through the ad-hoc wireless network to the other cars approaching. In all these cases nodes are highly mobile, the network has a rather large size, and the traditional addressing and routing techniques seem to be unable to ensure an efficient operation.

Thus, according to the tendencies that are to shape the networks of the future, new, scalable addressing and routing mechanisms are needed to provide continuous and consistent connectivity in a large and highly mobile network that uses different telecommunication technologies and devices.

In our paper we present a new, scalable addressing and routing architecture, designed so as to cope with these requirements. The rest of this paper is organized as follows. First, in section II, we present the main components of the proposed architecture and describe the corresponding addressing schemes. Then, in section III, we explain the node lookup and routing mechanisms. Section IV analyzes the advantages and drawbacks of our novel system, while section V presents some related work that our proposal can be compared with. Finally, we describe some future directions we intend to follow, and draw conclusions.

2. THE ARCHITECTURE

The goal of this paper is to propose an addressing and routing solution that fits the envisioned future network architecture. Our aim was not to develop another fully ad-hoc routing protocol, but to design an efficient and scalable solution that exploits the fact that in future networks, fixed network infrastructure will be available to support mobile and ad hoc wireless communication. In the following we briefly present the components of the architecture.

2.1. Components

There are three main architecture components in our approach. Wireless domains (WD) are meant to be ad hoc networks that include a fixed access point to the core network; these access points are called domain edges. Besides these edges, WDs are composed of nodes, which connect to the core network either directly, through the access points, or indirectly, through other nodes. WDs are radio networks featuring a wireless broadcast medium, which can be well utilized in inter-neighbor communication instead of sequence of unicast messages. Domain edges communicate with each other on a wired core network. In this paper, we do not focus on the core network architecture; we only assume that it supports the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

necessary addressing and routing primitives to enable communication among its components.

2.2. Addressing

Every wireless domain has a fixed domain ID in the network. Each node has a unique identifier, called Global Node Identifier (GNI), which is fixed as well. Moreover, each node has a Local Node Address (LNA), which reflects the position of the node in a domain. Inside a domain, nodes create a parent child hierarchy which is based on the hop distance from the domain edge. The LNA is a chain of numbers that correspond to the parent nodes of a certain node. For example, the parent of a node with an LNA of 10.2.4.5 has an LNA of 10.2.4, his parent has an LNA of 10.2, and so on. The addresses can be structured into a tree that is derived from the network graph. The vertices of the graph are nodes in the network, while an edge represents a parent-child relation between two nodes.

When a node arrives in a domain, first it broadcasts an address query message, to select its parent node. Each node that hears the query proposes an address to the newcomer. This address is composed from the LNA of the responding parent, extended with an arbitrary identifier, which differentiates the requesting new child from the other children of the parent node. From the received addresses the node selects its new address and therefore its parent (Figure 1).

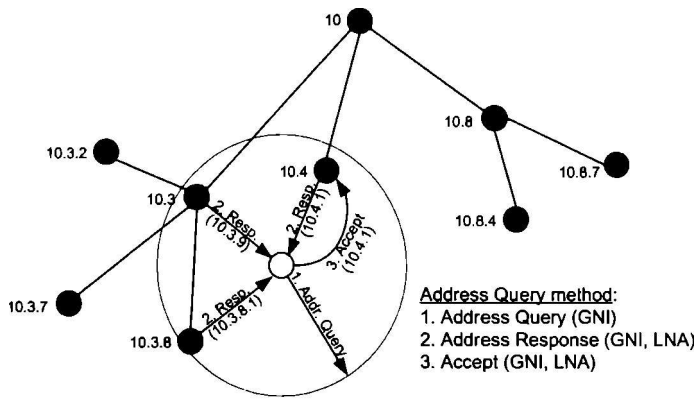


Figure 1: The address request procedure

The joining node always chooses the shortest address; by doing so, it tries to be as near as possible to the domain edge in the address tree. If the node receives several addresses with the same prefix length, it chooses the parent with the least children. This mechanism ensures a balanced address tree, which is important to achieve an efficient routing.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

3. THE ROUTING MECHANISM

Routing in our scheme is a process composed of several steps. Initially, the source node knows only the GNI of the destination. In order to be able to send packets, the source has to find out the other addresses (domain ID and LNA) that characterize the destination's current location. Then, routing can be completed. In the following we present the details of this process.

In the core network there is a distributed Global Node Locator (GNL) called Location Agency; it stores (GNI, domain ID) address pairs for each node in the network. These address pairs reflect a node's current domain location. The address lookup process is based on an algorithm that uses distributed hash tables (DHT); it stores GNIs and location related domain IDs in a distributed manner. The domain IDs are stored in so-called Location Agents (LA). Each Location Agent is responsible for a given interval from the global value space of the hash function; as such, it stores information (i.e., the current domain ID) about all nodes whose GNI hashes into its interval. These Location Agents form the distributed Location Agency.

As opposed to this, Domain Edges have a Local Address Directory function (LAD); they store (GNI, LNA) address pairs for every node in their wireless domain. The local address lookup function provided by a Domain Edge and the distributed global node locator function provided by a Location Agent are separate functionalities from the logical point of view. However, they can be physically collocated on the same machine. Therefore, in order to simplify our architecture, we consider in the following that a Domain Edge performs both functions.

Nodes in a domain maintain a Neighbor List (NL), where (GNI, LNA) address pairs are stored for neighbors within a given range. The Neighbor List is created and updated by periodical Hello messages exchanged among neighbors. Besides the direct neighbors, a node can store information related to nodes that are further away; these information can be exchanged by piggybacking them on the Hello messages. The range of the Neighbor List might vary, depending on the needs and the storing capabilities of the different nodes. The mandatory range of the list for the algorithm to operate correctly is one hop.

3.1. Address lookup

When a node wants to send a packet, it has to determine the current address of the target node from its GNI. First, it looks into its NL; if it cannot find the LNA of the destination, it initiates an address lookup message to its Domain Edge (Figure 2). In case of an unsuccessful lookup in the Edge's LAD table the edge node asks the global Location Agency about the current domain ID of the target. The address of the Location Agent where the requested domain ID may be stored is retrieved by hashing the GNI of the destination node. The responsible LA tells the domain ID of

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

target node; the ID is sent back to the source node, and can be used by core network routing to reach the target WD, where the destination node resides. Figure 2 presents the steps of this lookup process.

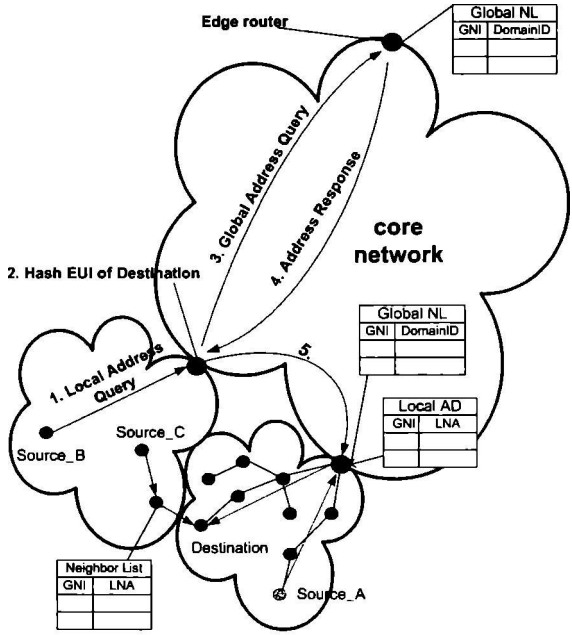


Figure 2: Node lookup and routing

3.2. Routing

According to the location of the source and the target node, intra- and inter domain routing can be differentiated. When intra domain routing is performed (Figure 2, Source_A), the source first looks for the target node's GNI in its neighbor list. If it finds a corresponding entry, it sends its packet directly to the destination. If not, the LNA of the target node can be obtained from the LAD service of the Domain Edge, by sending a LAD query upwards on the LNA tree.

The LAD lookup can be optimized by using the neighbor lists to reduce the number of hops the query has to travel. When initiating or forwarding a LAD lookup, a node looks into its NL to find an intra-domain neighbor that is closer to the domain edge (i.e., has a shorter LNA) than its own parent. If there is such a node, the lookup request is forwarded to it. As the root of the intra-domain LNA tree is the Domain Edge, it is assured that the request reaches it. In addition, some extra hops might be saved by shortcutting the LNA tree.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

If we are to deal with wireless domains that contain a large number of nodes, it might be reasonable to handle the LAD tables in a distributed manner as well, among several local nodes in the domain. Nevertheless, this functionality is not included yet in the current architecture.

If the source knows the LNA of the destination node, it searches for a prefix match between the target LNA and an LNA in its neighbor list. In case of success, the packet is sent to this neighbor. By doing so, a much shorter route can be found than the route obtained from the local address tree.

If there is no address prefix match, the packet is sent to the parent node. Each node obtains the LNA of its parent by simply cutting off the last component of its own LNA; no routing table has to be maintained.

In the worst case, this mechanism continues until the packet reaches the Domain Edge; then, the packet is sent through the parents of the target node, until it reaches the destination. Again, no routing table is needed. All the children of a parent node receive the packet. Among them, only the node with the LNA included in the target LNA forwards it.

In inter domain routing (Figure 2, Source_B) the source node knows the domain ID of the target node's current area from the Location Agency. First, the packet is sent to the Domain Edge; then, it is forwarded to the target domain. Here, the Domain Edge determines the target LNA from its LAD table; the packet is then sent down in the address hierarchy to the target node.

In some cases the packet might not have to travel through the core network, even if the communicating parties are situated in different wireless domains. This can happen when the source and the target are near to each other (typically the source is near the border of its domain). In this case, efficient packet forwarding can be obtained by the use of the Neighbor List (Figure 2, Source_C).

3.3. Multiple LNA trees

The above described routing mechanism has the drawback of putting a significant load on upper part of the LNA tree inside a domain; if no shortcut is found in the neighbor lists of the intermediate nodes, an intra-domain packet has to travel all the way up to the Domain Edge, and then down again on the appropriate branch of the LNA tree, towards the destination. Therefore, a significant amount of traffic will be handled by the nodes neighboring the Domain Edge. Another drawback is the lack of robustness of the approach, which pops up in the case of node mobility; if one of the nodes on the LNA tree moves out of the range of its parent (or child) node, the LNA-based routing path is broken.

To alleviate these drawbacks, we propose to use multiple LNA addresses and multiple trees, instead of only one. When entering the wireless domain, a new node

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

requests LNA addresses on all of these trees. Each neighboring node that receives the query (and that already has an LNA address on all of the trees) proposes to be the parent of the newcomer on all these trees. Then, the new node can select an LNA address for each tree, and reject the other proposals. Note that a node can have LNA addresses of different lengths on the different trees. Therefore, some of its proposals might be attractive to the newcomer, some not. Thus, the new node can choose to have different parents on the different LNA trees.

Once the new node has chosen its addresses on the different LNA trees, an update message is sent to the LAD service; all these addresses will be stored along with the GNI of the node. Therefore, when an address lookup is initiated by a source node, all the LNA addresses of the target will be returned.

Using multiple LNA trees in the same wireless domain has several advantages. On the one hand, when routing a packet, all these trees can be used to find the best path to the target node; an intermediate node that receives a packet on one of the LNA trees can decide to forward it over a different tree, if on that latter tree it has a neighboring node that is close to the target's corresponding LNA. Thus, routing can be accelerated.

On the other hand, robustness is enhanced through the use of multiple trees; if a node looses one of its parents (or child) – e.g., because it moves out from it's communication range – it can still use the other LNA trees to forward the packets. Thus, the routing paths are not broken.

The drawback of having multiple LNA trees is the increased amount of signaling and stored states. Both in the LAD tables and in the neighbor lists nodes have to be stored with several LNAs. Also, creating and maintaining multiple LNA trees requires additional signaling. However, this might represent an acceptable tradeoff if routing efficiency and robustness can be enhanced.

3.4. Alternative Routing

Besides the LNA-based routing mechanism, we also propose an alternative routing scheme that can improve the performance of the approach in several cases. Although the address tree based routing model provides a stateless solution for packet delivery, it suffers of some significant drawbacks. First, it lacks protection against link failures caused by node mobility, which is not affordable in a large-scale, highly dynamic networking environment that we envisaged. Second, in case of intra-domain routing it barely supports the optimal routes, as the majority of the packets are sent upwards to the Domain Edge, before being redistributed along the LNA-based tree branches. Thus, network traffic concentrates near the Domain Edge, which could lead to congestion and packet loss. In these conditions, there is a need for distributing the traffic load, for finding optimal routes and for

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

eliminating faults that are due to node mobility. Alternative routes could provide an answer to these issues.

The alternative routing solution that we propose can be summarized as a hash based probability routing. Its main goal is to find alternative routes that lead to the destination node, or to one of the address sub-trees that contains the destination node. The basic idea of the algorithm is to split up, at every node, the global value space of a hash function that is applied on the GNIs of the target nodes, among its neighbors. Thus, the number of sub-spaces depends on the number of a node's neighbors; this can be obtained from the node's Neighbor List, and it will vary from node to node. The hash function used for alternative routing can be similar to, or different from the hash function used for the Global Node Locator service.

When a packet arrives to a given node (A), the hash function is used to map the GNI of the destination into a sub-space that is assigned to one of node A's neighbors (node B). Then, B will be the next node to forward the packet from node A. It is important to mention that node B is not aware of the local assignment of sub-spaces done at node A; B will locally split the global value space among its own neighbors, apply the hash function on the destination's GNI, and choose the next hop (node C) to forward the packet according to its own assignment table.

The advantage of the hash based forwarding is that it performs stateless, semi-random packet forwarding, since only the GNI of nodes are taken into account; the packets of the same flow, sent to the same destination node, will be forwarded on the same alternative route, through the same intermediate nodes, without using any routing tables. The LNA, and therefore the current location of the target node, are not taken into account. Thus, when choosing the next hop on the alternative route, there is no guarantee that the packet will get physically closer to the destination. However, this might happen. In order to detect such a case, a node that receives a packet on an alternative route will always check its Neighbor List for the destination GNI; the packet will be further forwarded on the alternative route only if no matching entry is found in the NL. Moreover, the alternative route might cross the LNA-based routing path, in which case the cross point node can choose to stop forwarding on the alternative path, and send the packets down the LNA-branch. To do so, the receiving node first checks whether it is in the destination's LNA branch itself (its own LNA is included in the destination's LNA) or not. The hash-based forwarding is continued only if there's no match in the LNAs.

The two optimization possibilities can be applied together as well. That is, a node on the alternative route might fail in finding the destination node in its Neighbor List, but might discover a neighbor whose LNA is included in the LNA of target node. In that case, that neighbor will be chosen as next hop, without applying the hash function again on the target GNI. The efficiency of this mechanism depends on the range of the Neighbor List; the larger the range, the higher the probability of

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

finding a node in the neighborhood of the target. On the other hand, there is a tradeoff between the low-state nature of the routing scheme, expressed by the allowed size of the Neighbor List, and the possible optimization gains. Simulations in future work will give answers on the positive and negative effects of Neighbor List size.

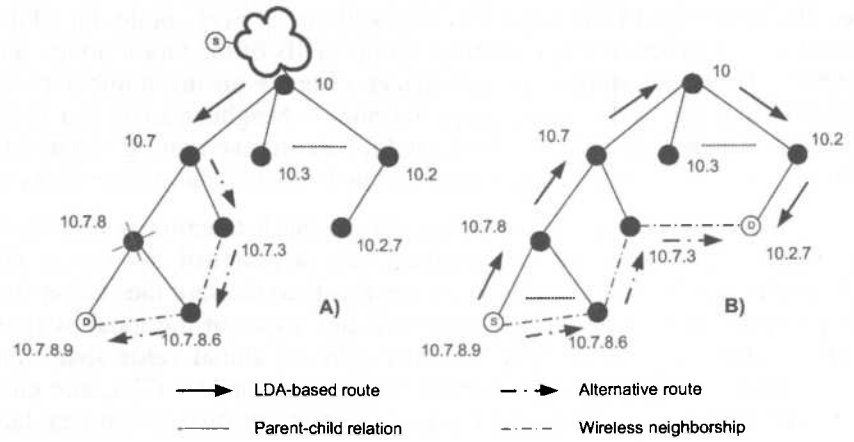


Figure 3: Using alternative routes in case of A) route failure B) route optimization

As stated before, the alternative routing scheme has a semi-random nature, which might result in routes that do not converge towards the destination. In order to limit the length of an alternative route, a TTL (Time To Live) field can be used; when the packet has crossed a number of alternative hops without reaching the target or without crossing the corresponding LNA-branch, it is dropped. Also, the lack of traditional routing tables and associated routing primitives (e.g., reverse path forwarding check) can result in the alternative routes containing loops. They can be avoided by using sequence numbers for the packets; a packet that was already handled by a node will not be processed again.

The alternative routing scheme can be used to handle several possible issues, such as fault tolerance, load balancing, or route optimization. For example, in case of a node failure in the middle of an LNA-branch, due to battery depletion or mobility, alternative routing can be used to get around the failure. This situation can typically arise when routing packets of a source located in a different wireless domain; after entering the destination's domain through the Domain Edge, packets are normally forwarded along the LNA-based tree. Figure 3.A presents an example for this scenario.

Alternative routing can also be applied to optimize routing paths in case of intra-domain communication. The source node starts to send packets along two possible paths (Figure 3.B). One corresponds to the LNA address tree, while the other is an

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

alternative path obtained through the above described hash routing scheme. In the second case route lookup can be constrained by the TTL field of the packet. The maximum number of alternative hops we allow can be obtained from the lengths of the source and destination addresses. For example, the route from 10.7.8.9 to 10.2.7 will consists of 4 intermediate nodes along the LNA address tree. As the destination is 5 hops away along this tree, alternative routes shorter than 6 hops are desirable.

Even if there are no shorter routes than the one based on the LNA-tree, we could use alternative routes to provide load balancing. Thus, considering the scenario in Figure 3.B we could accept 5–hop-long alternative routes to shift network load off the root nodes.

Since alternative routes continuously change according to node mobility and stateless packet forwarding, routes must be monitored and compared periodically to the reference route obtained from the LNA address tree. Upon deciding on the optimal route between the alternative and the LNA-based paths, packets will be sent only along that optimal route, until the next periodic evaluation.

The proposed alternative routing solution does not ensure shortest-path delivery; packet forwarding is rather semi-randomized, the actual location (LNA) of the target node not being taken into account. However, it provides a stateless solution to support fault tolerance, load balancing, or route optimization, with a certain probability, which depends on the range of the Neighbor List or the density of the neighboring nodes inside a wireless domain. The stateless nature of the approach is due to the hash-based forwarding that avoids the use of routing tables.

3.5. Mobility Handling

Since future networks are thought to be highly mobile, efficient mobility handling is a key issue in novel routing mechanisms.

There are a number of issues that should be addressed when handling mobility. If a node moves to another physical location, the topology should heal itself, and the addressing should remain consistent. Moreover, if a new node arrives at a domain, it should be easily attached to the system, and the corresponding lookup tables should be properly modified. In the following we give a brief overview on how the architecture deals with the mobility of nodes.

There are two basic kinds of mobility in our system. The first type is intra-domain mobility (the so-called "micro-mobility"): the node moves to another location inside the domain. The second type is inter-domain mobility (the so-called "macro-mobility"): the node moves outside the boundaries of its former domain, and joins a different ad hoc network (wireless domain).

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In case of intra-domain mobility, a mobile node should acquire a new LNA, according to its current location inside the wireless domain. However, address changes do not have to spread outside the specific domain. Since the distributed Location Agency only deals with the domain memberships of the nodes (it stores (GNI, domain ID) tuples), the change in the LNA of the mobile node does not affect the lookup tables in the LAs. Therefore, only the Domain Edge should be informed about the new LNA, in order to update its LAD table accordingly. Further advantage of keeping address changes local is that in case of ongoing inter-domain communication, neither the correspondent node of the moving node nor the Location Agency has to be informed of the address change, as Location Agency only bothers with domain ID changes. Message redirection due to node micro-mobility is done at the Domain Edges.

If a node moves out of the range of its parent node, the parent should delete it from its Neighbor List and redistribute the global GNI-based hash value space among its remaining neighbors. On the other hand, at its new location the mobile node acts as a newcomer, as described in Section II/B; it acquires a new LNA from its new neighbors, and registers it at the Domain Edge.

Inter-domain mobility requires different actions to be taken, as shown in Figure 5. If the moving node requests attachment in a new domain, its new parent node should inform the Domain Edge about the newcomer. The Domain Edge notices that there is no entry for the newcomer's GNI in its LAD table; thus, it creates an entry for it, it hashes the new node's GNI, and finds out which LA is responsible for it. Then, it initiates an update message towards that LA.

Upon reception of the update, the LA refreshes its GNL table. Moreover, if there was already an entry for that specific GNI associated with a different domain ID (i.e., the mobile node is not totally new to the network), it informs the Domain Edge of the node's former domain about the change. At last, the former Domain Edge refreshes its local lookup table, i.e., it deletes the corresponding entry. In case of ongoing communication, the former domain may notify the correspondent node of the mobile node, that a domain change occurred. To ensure a more sophisticated domain handoff procedure, soft-handover and buffering methods can be used.

Since nodes in a wireless domain are likely to communicate and move together in groups, the addressing method may be driven by policy considerations. These groups are controlled by a master device that represents the parent of the group sub-tree. In case of group mobility the master is the only one that has to interact with the environment on behalf of group address change. Only the group identifier address prefix has to be changed and updated at the Domain Edge when acquiring a new group parent.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

4. PERFORMANCE ANALYSIS

One of the most important goals of the proposed architecture is to be scalable, even in the presence of a large number of nodes in the network. With regard to scalability, it is not advisable to maintain routing tables in every router. Moreover, in ad hoc networks, all nodes act as routers; if there are a large number of nodes in the network, the routing tables consist of a large number of entries, which compromises scalability.

As opposed to traditional solutions, our approach provides stateless routing: the LNA-based hierarchical addressing and the hash-based forwarding mechanisms eliminate the need for states in the nodes. Besides this being a fully scalable method, the lack of routing tables can also speed up the routing decision process, as nodes do not have to perform searches in large tables.

Our stateless routing scheme is also fault tolerant, in the sense that there are no corrupted states in the routers (since there are no routing states at all). From the architecture's point of view, fault tolerance can also be achieved both at the upper and the lower levels of the hierarchy. Considering the upper level, the distributed Global Node Locator service in the core network ensures that if one of the Location Agents goes down there will be no stoppage in the network's operation. On the other hand, at the lower level the Neighbor Lists provide some flexibility in routing; if a node on the LNA-based path goes down, routing can be performed based on NLs.

The performance of our algorithm can also be enhanced through the use of multiple LNA trees, as presented earlier. If several trees are deployed, routing efficiency and robustness increases. However, multiple trees generate an increased signaling overhead as well. Therefore, the right balance should be found between the number of used LNA addresses and the overhead that is still acceptable to the wireless network. It is straightforward to think that the relative performance enhancements will decrease with the number of LNA trees (i.e., using two LNA trees instead of only one, changes the performance of the algorithm in much more drastically than using ten LNA trees instead of nine). However, this will depend on several factors, such as the size of the domain or the mobility patterns of the nodes. Evaluating the usefulness of the multiple LNA trees for different scenarios will necessitate a thorough simulation-based analysis.

Alternative routes back up the regular, LNA-based routing. The significance of the alternative routing mechanism is twofold: it does not only provide routes in case of node failures, radio transmission problems (e.g., noise, interference) and mobility, but can also be used as a load balancing method, shifting traffic load off the nodes close to the root.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Handling mobility is of great importance in future networking scenarios, as the number of mobile nodes is strongly increasing. The architecture supports both intra- and inter-domain mobility. Note that the alternative routing mechanism is suitable for a highly mobile scenario as well.

5. RELATED WORK

In this section we overview some of the related work in the literature that uses similar concepts to solve addressing or routing problems. We will briefly present the advantages and drawbacks of our solution when compared to these existing proposals.

Landmark [1] applies hierarchical addressing to build scalable network architecture. It defines Landmark nodes, which are routers whose neighbors within a certain number of hops contain routing entries for that router. The address of a node is a chain of different hierarchical level Landmarks. Every node stores the Landmarks in a hierarchical manner; and stores a neighbor list. The address distribution and storage method used by the Landmark Hierarchy is also a scalable solution, and resembles to our addressing method. However, Landmark was mainly proposed for wired networks; therefore, it does not consider the physical properties of the wireless networks, as opposed to our proposal.

If we are to distribute many different values in a balanced form, hash functions can be used to map the addresses into a hash value interval. This interval can be partitioned, with the portions of the interval being assigned to the different nodes; each node is responsible for the addresses that are mapped into its hash interval portion. This method is especially used in DHT based routing systems, such as Content Addressable Networks (CAN) [2], Chord [3], or Tribe [4].

In our solution, the distribution of the GNL database uses a similar method; the global GNI-based hash value space is partitioned among a set of Location Agents (which we considered to be collocated with the Domain Edges, to simplify the architecture). Moreover, during alternative routing, a given node partitions the global GNI hash value space among all the nodes in its immediate vicinity.

In overlay networks, such as those created through the use of DHTs, it is useful for nodes to know the addresses of the nodes in their immediate vicinity in the overlay. The same holds for wireless environments, where nodes might store information about their neighbors in the wireless network space. To store a so-called neighbor list needs some extra resources in nodes, but it can provide a more efficient routing. A well known system, in which the nodes use a list to store the addresses of the nodes in their vicinity, is Pastry [5]; it provides a scalable, decentralized object location and routing solution.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

In Pastry, the size N of the neighborhood set is determined; at all times, the neighborhood set is completely filled with the address of the N closest nodes. As opposed to this, in our proposal the size of the Neighbor List is not determined. It contains entries for all the nodes that are one or a given number of hops away from the node; thus, it has a variable size that depends on the size of the network, the mobility of its elements, etc.

Finally, a routing method that is in a way similar to our alternative routing scheme is presented in [6]. Rumor routing is based on a routing query being sent on a random walk, until it reaches an already established path. If such a path is found, further routing is done along that path. If not, the routing query is resubmitted, in the hope of the new random walk leading to a useful path. The alternative routing scheme we propose is different in the sense that the hash-based forwarding scheme ensures the packets for the same destination being sent over the same alternative path, as opposed to the ever changing random walks in Rumor routing.

6. CONCLUSIONS AND FUTURE WORK

The increase in the number of communicating entities in today's networks, the need for supporting small and large scale mobility, and the justification of ad hoc networks have raised new challenges that routing protocols have to face. In the present paper we introduced a novel architecture that aims to provide IP-independent addressing and a scalable routing mechanism, which operates in a stateless manner. An alternative routing scheme is proposed to handle node failures, load balancing, and mobility support.

The goal of the paper was only to demonstrate the need for new, scalable addressing and routing mechanism, and to lay down the basic principles of a novel architecture that can efficiently handle the requirements of future networks. However, there are a lot of points in the proposed approach, where optimization is needed.

One important issue to optimize is the balancing of the address tree. A well balanced LNA assignment can provide better connectivity, and enable shorter and faster routes. Keeping the address tree balanced implies re-addressing functions that have to be designed in a scalable manner as well.

Moreover, re-addressing is needed when a parent node of a sub-tree moves away; the "orphan" sub-tree has to be "adopted" by another node. Our hierarchical addressing scheme provides aggregation opportunities, as the entire sub-tree can be easily re-addressed. If the root node of the sub-tree finds a new possible parent, re-addressing means replacing the LNA of the former parent node with the LNA of the new parent in the LNAs of all the sub-tree nodes. Moreover, a single update message is enough to refresh the LAD entries that correspond to all the sub-tree nodes in the Domain Edge. However, these aggregated re-addressing steps can lead

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

to unbalanced LNA trees. Finding an efficient optimization scheme for this issue is subject of future work.

The issue of security is also a problem that needs to be addressed. Our goal in this article was to introduce an architecture that simplifies node lookup, and a routing solution that exhibits the properties of wireless medium and keeps stored states low. Security problems arising in our networking context can be caused by address spoofing, that effect on the structure of address-tree, or denial of packet forwarding that may block best-effort forwarding sequences. Our goal in the future is to design the functions of our architecture keeping in mind the mentioned problems, but security extends beyond the scope of this paper.

Finally, implementing a prototype system, and performing simulations to test the efficiency of the architecture is also subject of future work.

REFERENCES

[1] TSUCHIJA, P. F.: *The Landmark hierarchy: a new hierarchy for routing in very large networks.* In Proceedings of the ACM SIGCOMM, Stanford, CA, August 1988, pp. 35-42.

[2] RATNASAMY, S., FRANCIS, P., HANDLEY, M. KARP, R., SHENKER, S.: *A Scalable Content Addressable Network.* In Proceedings of the ACM SIGCOMM, San Diego, August 2001, pp. 161-172.

[3] STOICA, I., MORRIS, R., KARGER, D. KAASHOEK, M F., BALAKRISHNAN, H.: *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications.* In Proceedings of the ACM SIGCOMM, San Diego, August 2001

[4] VIANA, A. C.., Amorim, M. D., Fdida, S., Rezende, J. F..: *Indirect Routing Using Distributed Location Information.* In Proceedings of IEEE International Conference on Pervasive Computing and Communications, PerCom'03

[5] DRUSCHEL, P., ROWSTORN, A.: *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.* In Proceedings of the 18th IFIP/ACM International on Distributed Systems Platform, (Middleware 2001), November 2001.

[6] BRAGINSKY, D., ESTRIN, D.: *Rumor routing algorithm for sensor networks.* In Proceedings of the First ACM Workshop on Sensor Networks and Applications, Atlanta, GA, USA, October 2002, pp. 22-31.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 159-175

TP MODEL TRANSFORMATION BASED CONTROL OF THE TORA SYSTEM

ZOLTÁN PETRES, BARNA RESKÓ, AND PÉTER BARANYI
Integrated Intelligent Systems Japanese–Hungarian Laboratory,
Budapest University of Technology and Economics
H–1521 Budapest, Műegyetem rakpart 2.

Computer and Automation Research Institute,
Hungarian Academy of Sciences
H–1111 Budapest, Kende u. 13–17.
petres@tmit.bme.hu,{resko,baranyi}@sztaki.hu

[Received October 2004 and accepted February 2005]

Abstract. This paper presents a case study of the TP (Tensor Product) model transformation in the control of a nonlinear benchmark problem. We design a nonlinear controller of translational oscillation with an eccentric rotational proof mass actuator (TORA) system via TP model transformation and LMI (Linear Matrix Inequality) based controller design technique that is also capable of the reference signal tracking control. The main contribution of the paper is to show that both numerical methods the TP model transformation and the LMI can readily be executed computer independently on the given problem and without analytical derivations, that, hence, lead to a fast way of controller designs for a class of engineering control problems. Numerical simulation is used in the paper to provide empirical validation of the control results.

Keywords: nonlinear control design, tensor product model, linear matrix inequalities, parallel distributed compensation, TORA system

1. INTRODUCTION

Recently a control design method was proposed for the stabilization of parameter varying nonlinear state-space models [2–4]. This method is based on two numerical steps. In the first step the TP model transformation [4] is executed, while in the second step LMIs are solved under the PDC (Parallel Distributed Compensation) framework, that also includes the feasible solution of LMIs' The book [20] refers to a great number of related papers dealing with PDC design framework. The first

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

step is capable of transforming a given state-space model into a tensor product form (which is identical with a class of the Takagi–Sugeno inference operator based fuzzy model, see in Section 5.1) whereupon design techniques of the PDC framework, can immediately be executed. The second step results in a controller according to various different control specifications.
It is worth noticing here that both steps are executed numerically by computers. This implies two advantages such as:

1. the controller can be derived automatically, without analytic derivations;

2. the identified model which the control design method starts with can be defined either by analytical equations or by other soft-computing techniques, for instance by neural networks, fuzzy logic systems, or algorithms based on *Rudas*-type generalized operators [17, 18].

The main goal of this paper is to study, via the control of the TORA system example, how to execute the TP model transformation based control design method and to show its performance in case of reference signal tracking control. This control problem has a great comparative literature related to different control theories, and also a special issue of the *International Journal of Robust and Nonlinear Control* was devoted to describe the control problem and to present several control design methods including optimal control theory, Lyapunov backstepping, passivity theory, fuzzy logic, computing with words, etc. The overview of this literature is behind the scope of this paper, but we refer the reader to [1, 5, 12, 14, 20]
The rest of the paper is organized as follows. Section 2 introduces the notation being used in this paper. Section 3 illustrates the case study of this paper, the TORA system. Section 4 briefly summarizes some preliminaries and defines the convex state-space TP model. Section 5 presents the TP model transformation and Section 6 describes the LMI based controller design. Section 7 is devoted for evaluation of the derived controllers, and finally Section 8 concludes the paper.

2. NOMENCLATURE

This section is devoted to introduce the notations being used in this paper.

- $\{a, b, \ldots\}$ = scalar values
- $\{\mathbf{a}, \mathbf{b}, \ldots\}$ = vectors

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

- $\{\mathbf{A}, \mathbf{B}, \ldots\}$ = matrices
- $\{\mathcal{A}, \mathcal{B}, \ldots\}$ = tensors
- $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ = vector space of real valued $(I_1 \times I_2 \times \quad \times I_N)$-tensors
- $\diamond_{i,j,n,\ldots}$ = indices, they define lower order: for example, an element of matrix $\mathbf{A}$ at row-column number i, j is symbolized as $(\mathbf{A})_{i,j} = a_{i,}$ Systematically, the ith column vector of $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e. $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots \end{bmatrix}$
- $\diamond_{I,J,N,\ldots}$ = index upper bound: for example: $i = 1 \ldots I$, $j = 1 \ldots J$, $n = 1 \ldots N$ or $i_n = 1 \ldots I_n$
- $\mathbf{A}_{(n)}$ = n-mode matrix of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$
- $\text{rank}_n(\mathcal{A})$ = n-mode rank of tensor $\mathcal{A}$
- $\mathcal{A} \times_n \mathbf{U}$ = n-mode matrix-tensor product
- $\mathcal{A} \otimes_n \mathbf{U}_n$ = multiple product as $\mathcal{A} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \quad \times_N \mathbf{U}_N$
- $\mathbf{A}^+$ = the pseudo inverse of matrix $\mathbf{A}$

Detailed discussion of tensor notations and operations is given in [13].

3. TORA SYSTEM

The Translational Oscillations with a Rotational Actuator (TORA) system[1] was developed as a simplified model of a dual-spin spacecraft [13]. Later, Bernstein and his colleagues at the University of Michigan, Ann Arbor, turned it into a benchmark problem for nonlinear control [5, 7, 8].

The system shown in Fig. 1 represents a translational oscillator with an eccentric rotational proof-mass actuator. The oscillator consists of a cart of mass M connected to a fixed wall by a linear spring of stiffness k. The cart is constrained to have one-dimensional travel. The proof-mass actuator attached to the cart has mass m and moment of inertia I about its center of mass, which is located at distance e from the point about which the proof mass rotates. The motion occurs in a horizontal plane, so that no gravitational forces need to be considered. In Fig. 1, N denotes the control torque applied to the proof mass, and F is the disturbance force on the cart.

[1] Also referred to as the rotational/translational proof-mass actuator (RTAC) system.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

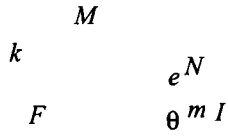


Figure 1: TORA system

Let q and $\dot{q}$ denote the translational position and velocity of the cart, and let θ and $\dot{\theta}$ denote the angular position and velocity of the rotational proof mass, where $\theta = 0$ deg is perpendicular to the motion of the cart, and $\theta = 90$ deg is aligned with the positive q direction. The equations of motion are given by

$$\begin{aligned} (M+m)\ddot{q} + kq \qquad & -me(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) + F \\ (I+me^2)\ddot{\theta} \qquad & -me\ddot{q}\cos\theta + N \end{aligned}$$

With the normalization

$$\begin{aligned} \xi &\triangleq \sqrt{\tfrac{M+m}{I+me^2}}\,q, \qquad \tau \triangleq \sqrt{\tfrac{k}{M+m}}\,t, \\ u &\triangleq \tfrac{M+m}{k(I+me^2)}N, \quad w \triangleq \tfrac{1}{k}\sqrt{\tfrac{M+m}{I+me^2}}\,F, \end{aligned}$$

the equation of motion become

$$\begin{aligned} \ddot{\xi} + \xi \qquad & \varepsilon\left(\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta\right) + w \\ \ddot{\theta} \qquad & -\varepsilon\ddot{\xi}\cos\theta + u \end{aligned}$$

where θ is the normalized cart position, and w and u represent the dimensionless disturbance and control torque, respectively. In the normalized equations, the symbol $(\cdot)$ represents differentiation with respect to the normalized time τ. The coupling between the translational and rotational motions is represented by the parameter ε which is defined by

$$\varepsilon \triangleq \frac{me}{\sqrt{(I+me^2)(M+m)}}$$

Letting $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}^T = \begin{pmatrix} \xi & \dot{\xi} & \theta & \dot{\theta} \end{pmatrix}^T$ the dimensionless equations of motion in first-order form are given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u + \mathbf{d}(\mathbf{x})w, \tag{1}$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Table 1: Parameters of the TORA system

Description	Parameter	Value	Units
Cart mass	M	1.3608	kg
Arm mass	m	0.096	kg
Arm eccentricity	e	0.0592	m
Arm inertia	I	0.0002175	kg m^2
Spring stiffness	k	186.3	N/m
Coupling parameter	ε	0.200	—

where

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-1}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 & 0 & 0 & 1 \\ \frac{\varepsilon\cos x_3}{1-\varepsilon^2\cos^2 x_3} & 0 & 0 & \frac{-\varepsilon x_4 \sin x_3}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix}$$

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} 0 \\ \frac{-\varepsilon\cos x_3}{1-\varepsilon^2\cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix} \qquad \mathbf{d}(\mathbf{x}) = \begin{pmatrix} 0 \\ \frac{1}{1-\varepsilon^2\cos^2 x_3} \\ 0 \\ \frac{-\varepsilon\cos x_3}{1-\varepsilon^2\cos^2 x_3} \end{pmatrix}$$

Note that u, the control input, is the normalized torque N and w, the disturbance, is the normalized force F In the followings consider the case of no disturbance. The parameters of the simulated system are given in Table 1.

4. BASIC CONCEPTS

4.1. Parameter-varying state-space model

Consider parameter-varying state-space model:

$$\begin{aligned} s\mathbf{x}(t) \quad & \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \\ \mathbf{y}(t) \quad & \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t), \end{aligned} \tag{2}$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state vector $\mathbf{x}(t)$. The system matrix

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{pmatrix} \in \mathbb{R}^{O\times I} \tag{3}$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

is a parameter-varying object, where $\mathbf{p}(t) \in \Omega$ is time varying N-dimensional parameter vector, where $\Omega = [a_1, b_1] \times [a_2, b_2] \times \quad \times [a_N, b_N] \subset \mathbb{R}^N$ is a closed hypercube. $\mathbf{p}(t)$ can also include some elements of $\mathbf{x}(t)$. Further, for a continuous-time system $s\mathbf{x}(t) = \dot{\mathbf{x}}(t)$; and for a discrete-time system $s\mathbf{x}(k) = \mathbf{x}(k+1)$ holds.

4.2. Convex state-space TP model

Eq. (3) can be approximated for any parameter $\mathbf{p}(t)$ as a convex combination of the R linear time-invariant (LTI) system matrices $\mathbf{S}_r$, $r = 1 \ldots R$. Matrices $\mathbf{S}_r$ are also termed as vertex system matrices. Therefore, one can define basis functions $w_r(\mathbf{p}(t)) \in [0,1] \subset \mathbb{R}$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ belongs to the convex hull of $\mathbf{S}_r$ as $\mathbf{S}(\mathbf{p}(t)) = \mathrm{co}\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}$, where vector $\mathbf{w}(\mathbf{p}(t))$ contains the basis functions $w_r(\mathbf{p}(t))$ of the convex combination. This kind of approximation is termed as, for instance, basis function based approximation, B-spline approximation, or tensor product approximation, see Chapter 3.2 of [16] and [9], and one can find the above model as *polytopic* model in control theories. The control design methodology, to be applied in this paper, uses univariate basis functions. Thus, the explicit form of the convex combination in terms of tensor product becomes:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \quad \sum_{i_N=1}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n(t)) \mathbf{S}_{i_1,i_2,\ldots,i_N} \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \tag{4}$$

The (4) is termed as TP model in this paper. Function $w_{n,j}(p_n(t)) \in [0,1]$ is the jth univariate basis function defined on the nth dimension of Ω, and $p_n(t)$ is the nth element of vector $\mathbf{p}(t)$. The I_n $(n = 1, \ldots, N)$ is the number of univariate basis functions used in the nth dimension of the parameter vector $\mathbf{p}(t)$. The multiple index $(i_1, i_2, \ldots, i_N)$ refers to the LTI system corresponding to the i_nth basis function in the nth dimension. Hence, the number of LTI vertex systems $\mathbf{S}_{i_1,i_2,\ldots,i_N}$ is obviously $R = \prod_n I_n$.

Remark 1 *Eq. (4) is also known as the explicit inference form of the Takagi–Sugeno inference operator based fuzzy model (TS fuzzy model for brevity). For instance, (4) is defined by fuzzy rules:*

$$\begin{aligned} \textbf{IF} \quad & w_{1,i_1}(p_1(t)) \quad \textbf{AND} \quad w_{2,i_2}(p_2(t)) \\ & w_{N,i_N}(p_N(t)) \quad \textbf{THEN} \quad \mathbf{S}_{i_1,i_2,\ldots,i_N}, \end{aligned}$$

where functions $w_{n,i_n}(p_n(t))$ represent the antecedent fuzzy sets and $\mathbf{S}_{i_1,i_2,\ldots,i_N}$ represents the consequent systems.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

One can rewrite (4) in the concise TP form as:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \underset{\delta}{\approx} \left(\mathcal{S} \underset{n=1}{\overset{N}{\otimes}} \mathbf{w}_n(p_n(t)) \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{5}$$

that is

$$\mathbf{S}(\mathbf{p}(t)) \underset{\varepsilon}{\approx} \mathcal{S} \underset{n=1}{\overset{N}{\otimes}} \mathbf{w}_n(p_n(t)).$$

Here, row vector $\mathbf{w}_n(p_n) \in \mathbb{R}^{I_n}$ contains the basis functions $w_{n,i_n}(p_n)$, the $N+2$-dimensional coefficient tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times O \times I}$ is constructed from the LTI vertex system matrices $\mathbf{S}_{i_1,i_2,\ldots,i_N} \in \mathbb{R}^{O \times I}$ The first N dimensions of $\mathcal{S}$ are assigned to the dimensions of Ω. The convex combination of the LTI vertex systems is ensured by the conditions:

Definition 1 *The TP model (5) is convex if:*

$$\forall n, i, p_n(t) \quad w_{n,i}(p_n(t)) \in [0,1]; \tag{6}$$

$$\forall n, p_n(t) \quad \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \tag{7}$$

This simply means that $\mathbf{S}(\mathbf{p}(t))$ is within the convex hull of LTI vertex systems $\mathbf{S}_{i_1,i_2,\ldots,i_N}$ for any $\mathbf{p}(t) \in \Omega$.

Remark 2 $\mathbf{S}(\mathbf{p}(t))$ *has finite TP model representation in many cases* ($\varepsilon = 0$ *in (5)). However, one should face that exact finite element TP model representation does not exist in general* ($\varepsilon > 0$ *in (5)), see [21]. In this case* $\varepsilon \to 0$*, when the number of LTI systems involved in the TP model goes to* ∞.

We define here a further characteristic of the convex TP model.

Definition 2 *The LTI vertex systems form a tight convex hull if their corresponding basis functions have the following feature:*

$$\forall n, i_n; \max_{p_n(t)} (w_{n,i_n}(p_n(t))) \underset{\delta_{n,i}}{\approx} 1, \tag{8}$$

where $\forall \delta_{n,i_n}$ *as small as possible. For instance, the basis functions are determined subject to*

$$\text{minimize}(\|\delta\|_{L_2}),$$

where vector δ *consists of all* δ_{n,i_n}.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

5. TRANSFORMATION OF THE TORA SYSTEM TO TP MODEL FORM

First we give a brief introduction to the TP model transformation based on papers [2,4].

5.1. TP model transformation

The goal of the TP model transformation is to transform a given state-space model (2) into convex TP model, in which the LTI systems form a tight convex hull. Namely, the TP model transformation results in (5) with conditions (6) and (7), and searches the LTI systems as a points of a tight convex hull of $\mathbf{S}(\mathbf{p}(t))$, see (8).
The TP model transformation is a numerical method and has three key steps. The first step is the discretization of the given $\mathbf{S}(\mathbf{p}(t))$ via the sampling of $\mathbf{S}(\mathbf{p}(t))$ over a huge number of points $\mathbf{p} \in \Omega$. The sampling points are defined by a dense hyper rectangular grid. In order to loose minimal information during the discretization we apply as dense grid as possible. The second step extracts the LTI vertex systems from the sampled systems. This step is specialized to find the minimal number of LTI vertex systems as the vertex points of the tight convex hull of the sampled systems. The third step constructs the TP model based on the LTI vertex systems obtained in the second step. It defines the continuous basis functions to the LTI vertex systems.

Method 1 (TP model transformation)
Step 1) Discretization

- *Define the transformation space* Ω *as:* $\mathbf{p}(t) \in \Omega \quad [a_1, b_1] \times [a_2, b_2] \times \quad \times [a_N, b_N]$.
- *Define a hyper rectangular grid by equidistantly located grid-lines:* $g_{n,m_n} = a_n + \frac{b_n - a_n}{M_n - 1}(m_n - 1)$, $m_n = 1 \ldots M_n$. *The numbers of the grid lines in the dimensions are* M_n.
- *Sample the given function* $\mathbf{S}(\mathbf{p}(t))$ *over the grid-points:*

$$\mathbf{S}^s_{m_1,m_2,\ldots,m_N} = \mathbf{S}(\mathbf{p}_{m_1,m_2,\ldots,m_N}) \in \mathbb{R}^{O \times I},$$

where $\mathbf{p}_{m_1,m_2,\ldots,m_N} = \begin{pmatrix} g_{1,m_1} & & g_{N,m_N} \end{pmatrix}$. *Superscript "s" means "sampled".*

- *Store the sampled matrices* $\mathbf{S}^s_{m_1,m_2,\ldots,m_N}$ *into the tensor* $\mathcal{S}^s \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N \times O \times I}$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Step 2) Extracting the LTI vertex systems
This step uses Higher-Order Singular Value Decomposition (HOSVD), and transformations Non-negativeness (NN), Sum Normalization (SN) and Normalization (NO). The studies of HOSVD can be found in a large varieties of publications. This paper uses the concept and tensor notation of HOSVD as discussed in [13]. The SN, NN and NO transformations are introduced in [22] and [23].
This step executes HOSVD, extended with NN, SN and NO transformation, on the first N dimensions of tensor $\mathcal{S}^s$ During performing the HOSVD we discard all zero or small singular values σ_k and their corresponding singular vectors in all dimensions. As a result we have

$$\mathcal{S}^s \underset{\gamma}{\approx} \mathcal{S} \underset{n}{\otimes} \mathbf{U}_n,$$

where the error γ is bounded as:

$$\gamma = \left(\left\| \mathcal{S}^s - \mathcal{S} \underset{n}{\otimes} \mathbf{U}_n \right\|_{L_2} \right)^2 \leq \sum_k \sigma_k^2. \tag{9}$$

The resulting tensor $\mathcal{S}$, with the size of $(I_1 \times I_2 \times \quad \times I_N \times O \times I)$, where $\forall n \quad I_n \leq M_n$, contains the LTI vertex systems, and is immediately substitutable into (5). The NN and SN transformations guarantee that the resulting LTI vertex systems form a convex hull of the sampled systems in $\mathcal{S}^s$ When the transformation NO is executed the resulting LTI systems form the tight convex hull of the sampled systems.
The software implementations of HOSVD, NN, SN and NO are rather simple, for instance, in MATLAB.
Step 3) Constructing continuous basis system

- *One can determine the discretized points of the basis easily from matrices $\mathbf{U}_n$. The i_nth column vector $\mathbf{u}_{n,i_n=1\ldots I_n}$ of matrix $\mathbf{U}_n \in \mathbb{R}^{M_n \times I_n}$ determines one discretized basis function $w_{n,i_n}(p_n(t))$ of variable $p_n(t)$. The values u_{n,m_n,i_n} of column i_n define the values of the basis function $w_{n,i_n}(p_n(t))$ over the grid-lines $p_n(t) = g_{n,m_n}$:*

$$w_{n,i_n}(g_{n,m_n}) = u_{n,m_n,i_n}.$$

- *The basis functions can be determined over any points by the help of the given $\mathbf{S}(\mathbf{p}(t))$. In order to determine the basis functions in vector $\mathbf{w}_d(p_d)$, let p_k be fixed to the grid-lines as:*

$$p_k = g_{k,1} \quad k = 1 \ldots N, \quad k \neq d.$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Then for p_d:

$$\mathbf{w}_d(p_d) = (\mathbf{S}(\mathbf{p}))_{(3)} \left(\left(\mathcal{S} \underset{k}{\otimes} \mathbf{u}_{k,1} \right)_{(n)} \right)^{+}$$

where vector $\mathbf{p}$ *consists of elements* p_k *and* p_d *as* $\mathbf{p} = \begin{pmatrix} g_{1,1} & \cdots & p_d & \cdots & g_{N,1} \end{pmatrix}$, *and superscript "+" denotes pseudo inverse and* $\mathbf{u}_{k,1}$ *is the first row vector of* $\mathbf{U}_k$. *The third-mode matrix* $(\mathbf{S}(\mathbf{p}))_{(3)}$ *of matrix* $\mathbf{S}(\mathbf{p})$ *is understood such that matrix* $\mathbf{S}(\mathbf{p})$ *is considered as a three-dimensional tensor, where the length of the third dimension is one. This practically means that the matrix* $\mathbf{S}(\mathbf{p})$ *is stored into one row vector by placing the rows of* $\mathbf{S}(\mathbf{p})$ *next to each other, respectively.*

5.2. Determination of the convex state-space TP model form of the TORA system

Observe that the nonlinearity is caused by $x_3(t)$ and $x_4(t)$. For the TP model transformation we define the transformation space as $\Omega = [-a,a] \times [-a,a]$ ($x_3(t) \in [-a,a]$ and $x_4(t) \in [-a,a]$), where $a = \frac{55}{180}\pi$rad (note that these intervals can be arbitrarily defined). Let the density of the sampling grid be 100×100. The sampling results in $\mathbf{A}^s_{i,j}$ and $\mathbf{B}^s_{i,j}$, where $i,j = 1 \quad 100$. Then we construct the matrix $\mathbf{S}^s_{i,j} = \begin{pmatrix} \mathbf{A}^s_{i,j} & \mathbf{B}^s_{i,j} \end{pmatrix}$, and after that the tensor $\mathcal{S}^s \in \mathbb{R}^{100\times100\times4\times4}$ from $\mathbf{S}^s_{i,j}$. If we execute HOSVD on the first two dimensions of $\mathcal{S}^s$ then we find that the rank of $\mathcal{S}^s$ on the first two dimensions are 4 and 2 respectively. This means that the TORA system can be exactly given as convex combination of $4 \times 2 = 8$ linear vertex model. In the present case the fourth singular value of the first dimension is very small comparing to the other three ($\sigma_1 = 200.28, \sigma_2 = 2.0191, \sigma_3 = 0.93035, \sigma_4 = 0.0036238$), therefore we discard it. Consequently, we reduce the rank of the first dimension to three, which causes a dispensable error. In conclusion, the TP model transformation describes TORA system as:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{3} \sum_{j=1}^{2} w_{1,i}(x_3(t)) w_{2,j}(x_4(t)) \left(\mathbf{A}_{i,j}\mathbf{x}(t) + \mathbf{B}_{i,j}u(t) \right). \tag{10}$$

The basis functions $w_{1,i}(x_3(t))$ and $w_{2,j}(x_4(t))$ are depicted in Fig. 2.

6. DETERMINATION OF CONTROLLERS FOR THE TORA SYSTEM VIA PDC DESIGN FRAMEWORK

In the previous section we transformed the TORA system (1) to TP model form whereupon LMI design under the PDC framework can immediately be executed.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

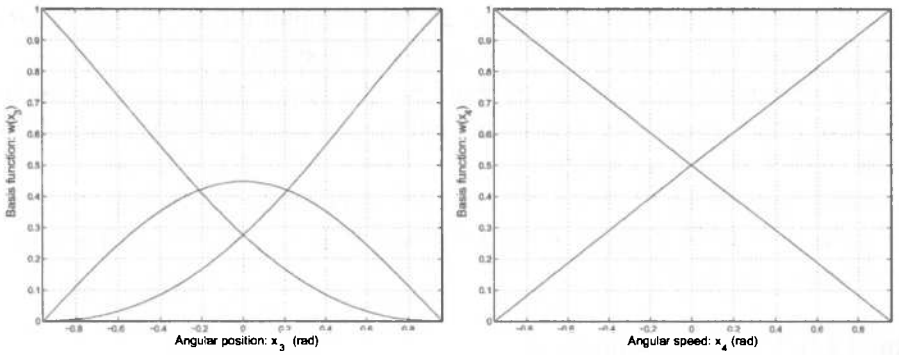


Figure 2: Basis functions on dimension $x_3(t)$ and $x_4(t)$

This section briefly introduces the main concept of the LMI design and calls LMI design theorems involving different control purposes. These LMI design theorems will be applied in the second part of this section to the TORA system.

As a result of the dramatic and continuing growth in computer power, and the advent of very powerful algorithms (and associated theory) for *convex optimization*, we can now solve very rapidly many *convex optimization* problems involving LMIs [15]. Many control problems and design specifications have LMI formulations [6, 11] that comes from the fact that LMI formulations have the ability to readily combine various design constraints or objectives in a numerical tractable manner. This is especially true for Lyapunov-based analysis and design.

As an alternative way of LMI based control design the PDC framework was introduced by Tanaka and Wang [20]. The PDC design framework determines one LTI feedback gain to each LTI vertex systems of a given convex TP model. The framework starts with the LTI vertex systems $\mathcal{S}$, and results in the vertex LTI gains $\bar{\mathcal{K}}$ of the controller. The $\mathcal{K}$ is computed by the LMI based stability theorems. After having the $\mathcal{K}$, the control value $\mathbf{u}(t)$ is determined by the help of the same basis functions as used in (5):

$$\mathbf{u}(t) = -\left(\mathcal{K} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t))\right)\mathbf{x}(t). \tag{11}$$

The LMI theorems, to be solved under the PDC framework, are selected according to the stability criteria and the desired control performance. For instance, the speed of response, constraints on the state vector or on the control value can be considered via properly selected LMI based stability theorems. The present control design ap-

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

plies different LMI theorems to achieve global asymptotic stability and to enforce constraint on the control value for the present TORA system.

In order to complete the paper let us recall briefly those LMI theorems, which will be applied in this paper. The derivations and the proofs of these theorems are fully detailed in [20].

Before dealing with the LMI theorems, we introduce a simple indexing technique in order to have direct link between the TP model form and the typical form of LMI formulations.

Method 2 (Index transformation) *Let*

$$\mathbf{S}_r = \begin{pmatrix} \mathbf{A}_r & \mathbf{B}_r \\ \mathbf{C}_r & \mathbf{D}_r \end{pmatrix} = \mathbf{S}_{i_1,i_2,\ldots,i_N},$$

where $r = \text{ordering}(i_1, i_2, \ldots, i_N)$ *(*$r = 1 \ldots R = \prod_n I_n$*). The function "ordering" results in the linear index equivalent of an N-dimensional array's index* $i_1, i_2, \ldots, i_N$*, when the size of the array is* $I_1 \times I_2 \times \quad \times I_N$*. Let the basis functions be defined according to the sequence of r:*

$$w_r(\mathbf{p}(t)) = \prod_n w_{n,i_n}(p_n(t)).$$

First we call one of the simplest LMI design theorems. The controller design can be derived from the Lyapunov stability theorems for global and asymptotic stability as shown in [19,20]:

Theorem 1 (Global and asymptotic stabilization of the convex TP model (5)) *Assume a given state-space model in TP form (5) with conditions (6) and (7).*
Find $\mathbf{X} > 0$ *and* $\mathbf{M}_r$ *satisfying eq.*

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} + \mathbf{M}_r^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_r > 0 \tag{12}$$

for all r and

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} - \mathbf{X}\mathbf{A}_s^T - \mathbf{A}_s\mathbf{X} + \tag{13}$$

$$+\mathbf{M}_s^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_s + \mathbf{M}_r^T\mathbf{B}_s^T + \mathbf{B}_s\mathbf{M}_r \geq 0.$$

for $r < s \leq R$*, except the pairs* (r,s) *such that* $w_r(\mathbf{p}(t))w_s(\mathbf{p}(t)) = 0, \forall \mathbf{p}(t)$.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Since the above conditions (12) and (13) are LMI's with respect to variables $\mathbf{X}$ and $\mathbf{M}_r$, we can find a positive definite matrix $\mathbf{X}$ and matrix $\mathbf{M}_r$ or determine that no such matrices exist. This is a convex feasibility problem. This numerical problem can be solved very efficiently by means of the most powerful tools available in the mathematical programming literature e.g. MATLAB LMI Control Toolbox [10]. The feedback gains can be obtained form the solutions $\mathbf{X}$ and $\mathbf{M}_r$ as

$$\mathbf{K}_r = \mathbf{M}_r \mathbf{X}^{-1} \tag{14}$$

Then, by the help of $r = \text{ordering}(i_1, i_2, \ldots, i_N)$ in Method 2 one can define feedbacks $\mathbf{K}_{i_1,i_2,\ldots,i_N}$ from $\mathbf{K}_r$ obtained in (14) and store into tensor $\mathcal{K}$ of (11).
In order to set constraints on the control value we add the following LMIs to (12) and (13):

Theorem 2 (Constraint on the control value) *Assume that* $\|\mathbf{x}(0)\| \leq \phi$, *where* $\mathbf{x}(0)$ *is unknown, but the upper bound* ϕ *is known. The constraint* $\|\mathbf{u}(t)\| \leq \mu$ *is enforced at all times* $t > 0$ *if the LMIs*

$$\begin{aligned} \phi^2 \mathbf{I} &\leq \mathbf{X} \\ \begin{pmatrix} \mathbf{X} & \mathbf{M}_i^T \\ \mathbf{M}_i & \mu^2 \mathbf{I} \end{pmatrix} &\geq 0 \end{aligned}$$

hold. We obtain the feedback gains as above (14) by solving all the LMIs.

7. EVALUATION OF THE DERIVED CONTROLLERS

To demonstrate the performance of the controlled system numerical experiments are presented in this section. The control values are computed by (11) as

$$u(t) = -\left(\sum_{i=1}^{3} \sum_{j=1}^{2} w_{1,i}(x_3(t)) w_{2,j}(x_4(t)) \mathbf{K}_{i,j} \right) \mathbf{x}(t).$$

in all cases of the simulations. Vectors $\mathbf{K}_{i,j}$ are resulted by LMIs discussed above.

7.1. Controller 1: Global and asymptotic stabilization of the TORA system

Let the resulting LTI vertex systems be substituted into the LMIs of the Theorem 1. The LMI solver shows that eq. (12) and (13) are feasible in the present case. Eq. (14) yields 6 LTI feedback gains $\mathbf{K}_{i,j}$.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

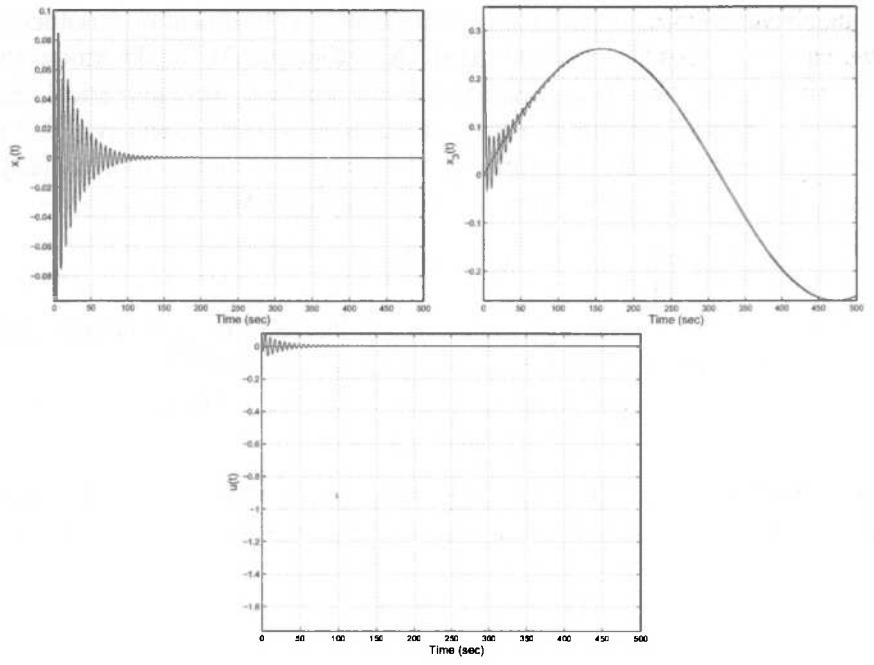


Figure 3: Controller 1: Global and asymptotic stabilization of the TORA system

In order to show the performance of the controller we generated a sinusoidal reference signal $f(t)$ with the following parameters: amplitude $\frac{15}{180}\pi$ rad and frequency 0.01 $\frac{\text{rad}}{\text{sec}}$. Thus, the input x_3 of the controller became $x_3(t) - f(t)$. The response of the reference signal tracking control is shown in Fig. 3. They show the state values $x_1(t)$, $x_3(t)$ (solid line) and $f(t)$ (dashed line), and the control value $u(t)$ for the initial conditions $x_1(0) = 0.1\,\text{m}$, $x_3(0) = \frac{20}{180}\pi\,\text{rad}$.

7.2. Controller 2: Constraint on the control value

In order to be capable of bounding the control values we apply Theorem 2. In the case of Controller 2 we define the minimal control value whereas the LMIs are still feasible. The response of the resulting controller is presented in Fig. 4. The control value in the second case ($\max(\|u\|) = 0.0759$) is significantly smaller in the first case ($\max(\|u\|) = 1.75$) while only a slight difference can be seen on the simulation results.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

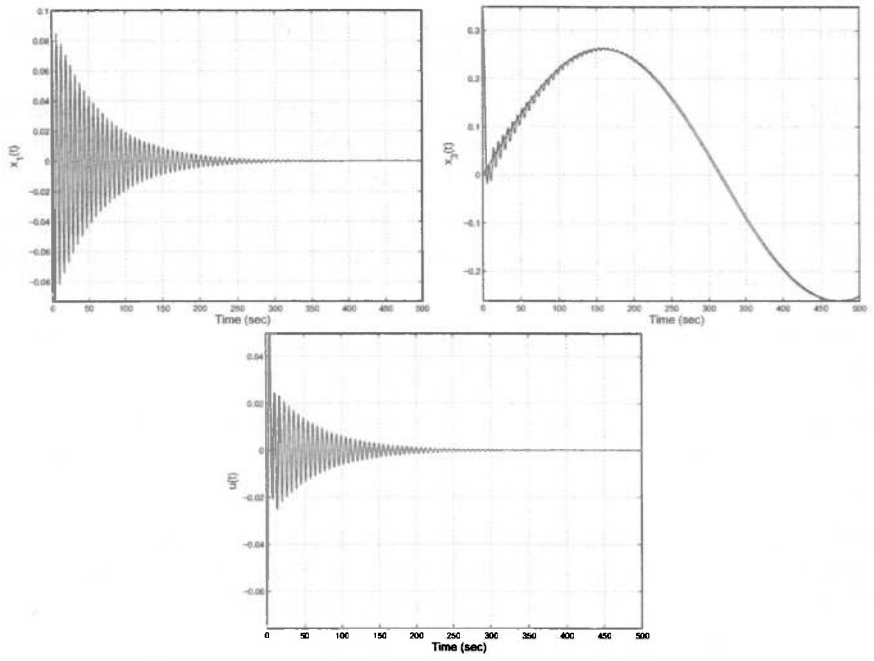


Figure 4: Controller 2: Constraint on the control value

8. CONCLUSION

This paper shows that once we have a computer program, for instance in MATLAB, of the TP model transformation and an LMI solver (MATLAB LMI Control Toolbox [10]) then the control design method, studied in this paper, can easily and automatically be executed. This paper shows an example when we want to achieve more than the global and asymptotic stability but also we want to define some constraint on the control value. The derived controllers' performance is shown in a reference signal tracking case. This paper applied rather simple LMI theorems in the controller design, but by applying more advanced theorems other control specifications can be taken into consideration during the controller design.

REFERENCES

[1] ALLEYNE, A.: *Physical insights on passivity-based TORA control designs*, IEEE Transaction on Control System Technology, **6** (1998), 436–439.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[2] BARANYI, P.: *TP model transformation as a way to LMI based controller design*, IEEE Transaction on Industrial Electronics, **51** (2004), No. 2, 387–400.

[3] BARANYI, P., KORONDI, P., PATTON, R. J., AND HASHIMOTO, H.: *Global asymptotic stabilization of the prototypical aeroelastic wing section via TP model transformation*, Asian Journal of Control, **7** (2004), No. 2, (to be printed).

[4] BARANYI, P., TIKK, D., YAM, Y., AND PATTON, R. J.: *From differential equations to PDC controller design via numerical transformation*, Computers in Industry, Elsevier Science, **51** (2003), 281–297.

[5] BERNSTEIN, D. S.: *Special issue: A nonlinear benchmark problem*, International Journal of Robust and Nonlinear Control, **8** (1998).

[6] BOYD, S. GHAOUI, L. E. FERON, E., AND BALAKRISHNAN, V.: *Linear matrix inequalities in system and control theory*, Philadelphia PA:SIAM, ISBN 0-89871-334-X, (1994).

[7] BUPP, R. T., BERNSTEIN, D. S., AND COPPOLA, V. T.: *A benchmark problem for nonlinear control design*, International Journal of Robust and Nonlinear Control, **8** (1998), 307–310.

[8] BUPP, R. T., BERNSTEIN, D. S., AND COPPOLA, V. T.: *Experimental implementation of integrator backstepping and passive nonlinear controllers on the RTAC testbed*, International Journal of Robust and Nonlinear Control, **8** (1998), 435–457.

[9] FARIN, G.: *Curves and surfaces for computer aided geometric design*, Academic Press Professional, San Diego, USA, 1990.

[10] GAHINET, P., NEMIROVSKI, A., LAUB, A. J., AND CHILALI, M.: *LMI Control Toolbox*, The MathWorks, Inc., 1995.

[11] GHAOUI, L. E. AND NICULESCU, S. I.: *Advances in linear matrix inequality methods in control*, in: *Advances in Design and Control* (L. E. Ghaoui and S. I. Niculescu, eds.), SIAM Books, Philadelphia, 2000

[12] JANKOVIC, M., FONTAINE, D., AND KOKOTOVIC, P. V.: *TORA example: Cascade and passivity control design*, IEEE Transaction on Control System Technology, **4** (1996), 292–297.

[13] LATHAUWER, L. D., MOOR, B. D., AND VANDEWALLE, J.: *A multi linear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications, **21** (2000), No. 4, 1253–1278.

[14] MARGALIOT, M. AND LANGHOLZ, G.: *Fuzzy control of a benchmark problem: A computing with words approach*, IEEE Transaction on Fuzzy Systems, **12** (2004), No. 2, 230–235.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[15] NESTEROV, Y. AND NEMIROVSKY, A.: *Interior-point polynomial methods in convex programing: Theory and Applications*, SIAM Books, Philadelphia, 1994.

[16] RAMSAY, J. O. AND SILVERMAN, B. W.: *Functional Data Analysis*, Springer series in statistics, Springer-Verlag, 1997.

[17] RUDAS, I. J. AND KAYNAK, O.: *Entropy-based operations on fuzzy sets*, IEEE Transactions on Fuzzy Systems, **6** (1998), 33–40.

[18] RUDAS, I. J. AND KAYNAK, O.: *Minimum and maximum fuzziness generalized operators*, Fuzzy Sets and Systems, **98** (1998), 83–94.

[19] SCHERER, C. W. AND WEILAND, S.: *Linear Matrix Inequalities in Control*, DISC course lecture notes, 2000, http://www.cs.ele.tue.nl/sweiland/lmi.htm.

[20] TANAKA, K. AND WANG, H. O.: *Fuzzy Control Systems Design and Analysis — A Linear Matrix Inequality Approach*, John Wiley and Sons, Inc., 2001.

[21] TIKK, D., BARANYI, P., AND PATTON, R. J.: *Polytopic and TS models are nowhere dense in the approximation model space*, IEEE International Conference on System, Man, and Cybernetics (SMC'02), (2002), proc. on CD.

[22] YAM, Y., BARANYI, P., AND YANG, C. T.: *Reduction of fuzzy rule base via singular value decomposition*, IEEE Trans. Fuzzy Systems, **7** (1999), No. 2, 120–132.

[23] YAM, Y. YANG, C. T., AND BARANYI, P.: *Singular Value Based Fuzzy Reduction with Relaxed Normalization Condition*, vol. 128 of *Studies in Fuzziness and Soft Computing, Interpretability Issues in Fuzzy Modeling*, Springer-Verlag, 2003 pp. 325–354.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Production Systems and Information Engineering
Volume 2 (2004), pp. 177-192

OBSERVER DESIGN VIA TP MODEL TRANSFORMATION

PÉTER BARANYI
Computer and Automation Research Institute,
Hungarian Academy of Sciences
H–1111 Budapest, Kende u. 13–17.
baranyi@sztaki.hu

YEUNG YAM
Chinese University of Hong Kong
Dept. Automation and Computer Aided Engineering
Shatin N.T., Hong Kong

[Received October 2004 and accepted February 2005]

Abstract. This paper presents a case study how to apply the recently proposed TP model transformation technique, that has been introduced for nonlinear state-feedback control design, to nonlinear observer design. The study is conducted through an example. This example treats the question of observer design to the prototypical aeroelastic wing section with structural nonlinearity. This type of model has been traditionally used for the theoretical as well as experimental analysis of two-dimensional aeroelastic behavior. The model investigated in the paper describes the nonlinear plunge and pitch motion of a wing, and exhibits complex nonlinear behavior. In preliminary works this prototypical aeroelastic wing section was stabilized by a state-feedback controller designed via TP model transformation and linear matrix inequalities. Extending this control strategy with the observer derived in this paper an output feedback strategy can be determined. Numerical simulations are used to provide empirical validation of the resulting observer.

Keywords: nonlinear control, linear parameter varying model, TP model transformation, parallel distributed compensation, linear matrix inequality

1. INTRODUCTION

The main goal of the paper is to study how to apply the TP (Tensor Product) model transformation to observer design. The motivation of this goal is that the TP model transformation was proposed under the Parallel Distributed Compensation (PDC) design framework [22] for nonlinear state feedback controller design [1, 5]. The TP model transformation is capable of transforming a given time varying (parameter dependent, where the parameters may include state variables) linear state-space model

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

into time varying convex combination of finite number of linear time invariant models. The resulting linear time invariant models can then be readily substituted into Linear Matrix Inequalities (LMI), available under the PDC design framework, to determine a time varying (parameter dependent, where the parameters may include state variables) nonlinear controller according to given control specifications. The whole above design can be executed numerically by computers and hence the controller can be determined without analytical derivations in acceptable time. In most cases not all of the state variables are available, but only some of them. This paper studies how to apply the result of the TP model transformation to observer design under the PDC design framework similarly to the controller design. The resulting observer can then be applied to estimate the unavailable state variables.

A few papers were printed in last years dealing with the state-feedback control design of the prototypical aeroelastic wing section via TP model transformation, for instance see [2–4]. This paper focuses attention on the observer design to the prototypical aeroelastic wing section since not all of the state variables of the prototypical aeroelastic wing section are available in reality. The combination of the state-feedback controller and the observer leads to the output feedback control of the prototypical aeroelastic wing section.

2. BASIC NOTATION

This section is devoted to introduce the notations being used in this paper: $\{a,b,..\}$: scalar values. $\{\mathbf{a},\mathbf{b},\ldots\}$: vectors. $\{\mathbf{A},\mathbf{B},\ldots\}$: matrices. $\{\mathcal{A},\mathcal{B},..\}$: tensors. The $\mathcal{R}^{I_1\times\cdots\times I_N}$: vector space of real valued $(I_1\times I_2\times\quad\times I_N)$-tensors. Subscript defines lower order: for example, an element of matrix $\mathbf{A}$ at row-column number i,j is symbolized as $(\mathbf{A})_{i,j}=a_{i,}$ Systematically, the ith column vector of $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e. $\mathbf{A}=\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots\end{bmatrix}$. $\diamond_{i,j,n},\ldots$: are indices. $\diamond_{I,J.N},\ldots$: index upper bound: for example: $i=1..I$, $j=1..J$, $n=1..N$ or $i_n=1..I_n$. $\mathbf{A}_{(n)}$: n-mode matrix of tensor $\mathcal{A}\in\mathcal{R}^{I_1\times I_2\times\cdots\times I_N}$ $\mathcal{A}\times_n\mathbf{U}$: n-mode matrix-tensor product. $\mathcal{A}\otimes_n\mathbf{U}_n$: multiple product as $\mathcal{A}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\quad\times_N\mathbf{U}_N$. Detailed discussion of tensor notations and operations is given in [16].

3. BASIC CONCEPTS

The detailed description of the TP model transformation and PDC design framework is beyond the scope of this paper and can be found in [1,2,5,22]. In the followings a few concepts are presented being used in this paper, for more details see [1,2,5,22].

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

3.1. Linear parameter-varying state-space model

Consider parameter-varying state-space model:

$$s\mathbf{x}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \tag{1}$$
$$\mathbf{y}(t) = \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t),$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state vector $\mathbf{x}(t)$. The system matrix

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{pmatrix} \in \mathcal{R}^{O\times I} \tag{2}$$

is a parameter-varying object, where $\mathbf{p}(t) \in \Omega$ is time varying $N-$dimensional parameter vector, where $\Omega = [a_1, b_1] \times [a_2, b_2] \times \quad \times [a_N, b_N] \subset \mathcal{R}^N$ is a closed hypercube. $\mathbf{p}(t)$ can also include some (or all) elements of $\mathbf{x}(t)$. Further, for a continuous-time system $s\mathbf{x}(t) = \dot{\mathbf{x}}(t)$; and for a discrete-time system $s\mathbf{x}(k) = \mathbf{x}(k+1)$ holds.

3.2. Convex state-space TP model

Equ. (2) can be approximated for any parameter $\mathbf{p}(t)$ as a convex combination of the R LTI system matrices $\mathbf{S}_r$, $r = 1..R$. Matrices $\mathbf{S}_r$ are also termed as vertex system matrices. Therefore, one can define basis functions $w_r(\mathbf{p}(t)) \in [0,1] \subset \mathcal{R}$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ belongs to the convex hull of $\mathbf{S}_r$ as $\mathbf{S}(\mathbf{p}(t)) = co\{\mathbf{S}_1, \mathbf{S}_2, .., \mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}$, where vector $\mathbf{w}(\mathbf{p}(t))$ contains the basis functions $w_r(\mathbf{p}(t))$ of the convex combination. The control design methodology, to be applied in this paper, uses univariate basis functions. Thus, the explicit form of the convex combination in terms of tensor product becomes:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \tag{3}$$
$$\left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \quad \sum_{i_N=1}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n(t)) \mathbf{S}_{i_1,i_2,..,i_N} \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}$$

(3) is called as TP model in this paper. Function $w_{n,j}(p_n(t)) \in [0,1]$ is the j-th univariate basis function defined on the n-th dimension of Ω, and $p_n(t)$ is the n-th element of vector $\mathbf{p}(t)$. I_n (n=1,...,N) is the number of univariate basis functions used in the n-th dimension of the parameter vector $\mathbf{p}(t)$. The multiple index $(i_1, i_2, ..., i_N)$ refers to the LTI system corresponding to the i_n-th basis function in the n-th dimension. Hence, the number of LTI vertex systems $\mathbf{S}_{i_1,i_2,..,i_N}$ is obviously $R = \prod_n I_n$. One can rewrite (3) in the concise TP form as:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \mathcal{S} \underset{n=1}{\overset{N}{\otimes}} \mathbf{w}_n(p_n(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{4}$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

that is

$$\mathbf{S}(\mathbf{p}(t)) \underset{\varepsilon}{\approx} \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)).$$

Here, ε represents the approximation error, and row vector $\mathbf{w}_n(p_n) \in \mathcal{R}^{I_n}$ contains the basis functions $w_{n,i_n}(p_n)$, the $N+2$ -dimensional coefficient tensor $\mathcal{S} \in \mathcal{R}^{I_1 \times \cdots \times I_N \times O \times I}$ is constructed from the LTI vertex system matrices $\mathbf{S}_{i_1,i_2,\ldots,i_N} \in \mathcal{R}^{O \times I}$ The first N dimensions of $\mathcal{S}$ are assigned to the dimensions of Ω. The convex combination of the LTI vertex systems is ensured by the conditions:

Definition 1. The TP model (4) is convex if:

$$\forall n, i, p_n(t) \quad w_{n,i}(p_n(t)) \in [0,1]; \tag{5}$$

$$\forall n, p_n(t) \quad \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \tag{6}$$

This simply means that $\mathbf{S}(\mathbf{p}(t))$ is within the convex hull of LTI vertex systems $\mathbf{S}_{i_1,i_2,..,i_N}$ for any $\mathbf{p}(t) \in \Omega$.

Remark 1. $\mathbf{S}(\mathbf{p}(t))$ has finite TP model representation in many cases ($\varepsilon = 0$ in (4)). However, one should face that exact finite element TP model representation does not exist in general ($\varepsilon > 0$ in (4)), see [25,26]. In this case $\varepsilon \mapsto 0$, when the number of LTI systems involved in the TP model goes to ∞. In the present observer design, the state-space dynamic model of the prototypical aeroelastic wing section can be exactly represented by a finite convex TP model.

4. MODEL OF THE PROTOTYPICAL AEROELASTIC WING SECTION

In the last few years various studies of aeroelastic systems have emerged. [14] presents a detailed background and refers to a number of papers dealing with the modelling and control of aeroelastic systems. The following provides a brief summary of this background.

Regarding the properties of aeroelastic systems one can find the study of free-play non-linearity by Tang and Dowell in [23,24], by Price et al. in [21] and [20], by Lee et al. in [17], and a complete study of a class of non-linearities is in [28], [20]. O'Neil et al. [18] examined the continuous structural non-linearity of aeroelastic systems. These papers conclude that an aerolesatic system may exhibit a variety of control phenomena such as *limit cycle oscillation, flutter* and even *chaotic vibrations.*

Control strategies have also been derived for aeroelastic systems. [6] shows that controllers, capable of stabilizing structural non-linearity over flow regimes, can be derived via classical multivariable control methods. However, while several authors

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

have investigated the effectiveness of linear control strategies for aeroelastic systems, experimental evidence has shown that linear control methods may not be reliable when non-linear effects predominate. For example in the case of large amplitude limit cycle oscillation behaviour the linear control methodologies [6] do not stabilize aeroelastic systems consistently. [12] and [6] proposed non-linear feedback control methodologies for a class of non-linear structural effects of the wing section [18]. Papers [12, 14, 15] develop a controller, capable of ensuring local asymptotic stability, via partial feedback linearization. It has been shown that by applying two control surfaces global stabilization can be achieved. For instance, adaptive feedback linearization [13] and the global feedback linearization technique were introduced for two control actuators in the work of [14]. TP model transformation based control design was introduced in [2–4]. This control design ensures global asymptotic stability with one control surface and is capable of involving various control specification beyond stability.

4.1. Equations of Motion

In this paper, we consider the problem of flutter suppression for the prototypical aeroelastic wing section as shown in Figure 1. The aerofoil is constrained to have two degrees of freedom, the plunge h and pitch α. The equations of motion of the system have been derived in many references (for example, see [10], and [9]), and can be written as

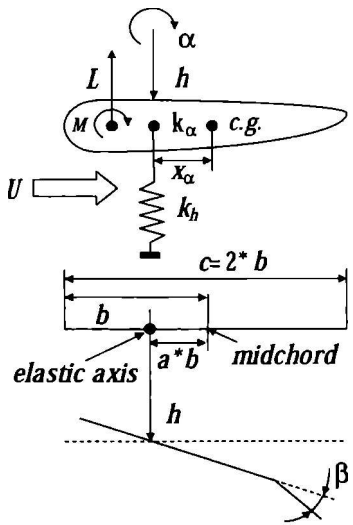


Figure 1: Aeroelastic model

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$\begin{pmatrix} m & mx_\alpha b \\ mx_\alpha b & I_alpha \end{pmatrix} \begin{pmatrix} \ddot{h} \\ \ddot{\alpha} \end{pmatrix} + \begin{pmatrix} c_h & 0 \\ 0 & c_\alpha \end{pmatrix} \begin{pmatrix} \dot{h} \\ \dot{\alpha} \end{pmatrix} + \tag{7}$$

$$+ \begin{pmatrix} k_h & 0 \\ 0 & k_\alpha(\alpha) \end{pmatrix} \begin{pmatrix} h \\ \alpha \end{pmatrix} = \begin{pmatrix} -L \\ M \end{pmatrix},$$

where

$$L = \rho U^2 b c_{l_\alpha} \left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a \right) b \frac{\dot{\alpha}}{U} \right) + \rho U^2 b c_{l_\beta} \beta \tag{8}$$

$$M = \rho U^2 b^2 c_{m_\alpha} \left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a \right) b \frac{\dot{\alpha}}{U} \right) + \rho U^2 b c_{m_\beta} \beta,$$

and where x_α is the non-dimensional distance between elastic axis and the centre of mass; m is the mass of the wing; I_α is the mass moment of inertia; b is semi-chord of the wing, and c_α and c_h respectively are the pitch and plunge structural damping coefficients, and k_h is the plunge structural spring constant. Traditionally, there have been many ways to represent the aerodynamic force L and moment M, including steady, quasi-steady, unsteady and non-linear aerodynamic models. In this paper we assume the quasi-steady aerodynamic force and moment, see work [10]. It is assumed that L and M are accurate for the class of low velocities concerned. Wind tunnel experiments are carried out in [6]. In the above equation ρ is the air density, U is the free stream velocity, c_{l_α} and c_{m_α} respectively, are lift and moment coefficients per angle of attack, and c_{l_β} and c_{m_β}, respectively are lift and moment coefficients per control surface deflection, and a is non-dimensional distance from the mid-chord to the elastic axis. β is the control surface deflection.

Several classes of non-linear stiffness contributions $k_\alpha(\alpha)$ have been studied in papers treating the open-loop dynamics of aeroelastic systems [8,23,27,28]. For the purpose of illustration, we now introduce the use of polynomial non-linearities. The non-linear stiffness term $k_\alpha(\alpha)$ is obtained by curve-fitting the measured displacement-moment data for non-linear spring as [19]:

$$k_\alpha(\alpha) = 2.82(1 - 22.1\alpha + 1315.5\alpha^2 + 8580\alpha^3 + 17289.7\alpha^4).$$

The equations of motion derived above exhibit limit cycle oscillation, as well as other non-linear response regimes including chaotic response [8, 19, 28]. The system parameters to be used in this paper are given in [1] and are obtained from experimental models described in full detail in works [14, 19].

With the flow velocity $u = 15(m/s)$ and the initial conditions of $\alpha = 0.1(rad)$ and $h = 0.01(m)$, the resulting time response of the non-linear system exhibits limit cycle oscillation, in good qualitative agreement with the behaviour expected in this class

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

of systems. Papers [18, 19] have shown the relations between limit cycle oscillation, magnitudes and initial conditions or flow velocities.
Let the equations (7) and (8) be combined and reformulated into state-space model form:

$$\mathbf{x}(t) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{pmatrix} \quad \text{and} \quad \mathbf{u}(t) = \beta.$$

Then we have:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) = \mathbf{S}(\mathbf{p}(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{9}$$

where

$$\mathbf{A}(\mathbf{p}(t)) = \begin{pmatrix} x_3 \\ x_4 \\ -k_1 x_1 - (k_2 U^2 + p(x_2))x_2 - c_1 x_3 - c_2 x_4 \\ -k_3 x_1 - (k_4 U^2 + q(x_2))x_2 - c_3 x_3 - c_4 x_4 \end{pmatrix}$$

$$\mathbf{B}(\mathbf{p}(t)) = \begin{pmatrix} 0 \\ 0 \\ g_3 U^2 \\ g_4 U^2 \end{pmatrix}$$

where $\mathbf{p}(t) \in \mathcal{R}^{N=2}$ contains values x_2 and U. The new variables are tabulated in Table 1. One should note that the equations of motion are also dependent upon the elastic axis location a.

5. OBSERVER DESIGN

The recently proposed very powerful numerical methods (and associated theory) for *convex optimization* involving Linear Matrix Inequalities (LMI) help us with the analysis and the design issues of dynamic systems models in acceptable computational time [7, 11]. One direction of these analysis and design methods is based on LMI's under the PDC design framework [22]. In this paper we apply the TP model transformation in combination with the PDC based observer design technique to derive viable observer methodologies for the prototypical aeroelastic wing section defined in the previous section. The key idea of the proposed design method is that the TP model transformation is utilized to represent the model (9) in convex TP model form with specific characteristics, whereupon PDC controller design techniques can immediately be executed.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Table 1: System variables

$$
\begin{aligned}
d &= m(I_\alpha - mx_\alpha^2 b^2)\\
k_1 &= \frac{I_\alpha k_h}{d}\\
k_2 &= \frac{I_\alpha \rho b c_{l_\alpha} + mx_\alpha b^3 \rho c_{m_\alpha}}{d}\\
k_3 &= \frac{-mx_\alpha b k_h}{d}\\
k_4 &= \frac{-mx_\alpha b^2 \rho c_{l_\alpha} - m\rho b^2 c_{m_\alpha}}{d}\\
p(\alpha) &= \frac{-mx_\alpha b}{d} k_\alpha(\alpha)\\
q(\alpha) &= \frac{m}{d} k_\alpha(\alpha)\\
c_1(U) &= \frac{I_\alpha(c_h + \rho U b c_{l_\alpha}) + mx_\alpha \rho U^3 c_{m_\alpha}}{d}\\
c_2(U) &= \frac{I_\alpha \rho U b^2 c_{l_\alpha}(\frac{1}{2} - a) - mx_\alpha b c_\alpha + mx_\alpha \rho U b^4 c_{m_\alpha}(\frac{1}{2} - a)}{d}\\
c_3(U) &= \frac{-mx_\alpha b c_h - mx_\alpha \rho U b^2 c_{l_\alpha} - m\rho U b^2 c_{m_\alpha}}{d}\\
c_4(U) &= \frac{mc_\alpha - mx_\alpha \rho U b^3 c_{l_\alpha}(\frac{1}{2} - a) - m\rho U b^3 c_{m_\alpha}(\frac{1}{2} - a)}{d}\\
g_3 &= \frac{1}{d}(-I_\alpha \rho b c_{l_\beta} - mx_\alpha b^3 \rho c_{m_\beta})\\
g_4 &= \frac{1}{d}(mx_\alpha b^2 \rho c_{l_\beta} + m\rho b^2 c_{m_\beta})
\end{aligned}
$$

5.1. TP model form of the prototypical aeroelastic wing section

5.1.1. TP model transformation

The goal of the TP model transformation is to transform a given state-space model (1) into convex TP model [1,2,5], in which the LTI systems form a tight convex hull. Namely, the TP model transformation results in (4) with conditions (5) and (6), and searches the LTI systems as a points of a tight convex hull of $\mathbf{S}(\mathbf{p}(t))$.

The detailed description of the TP model transformation is discussed in [1,2,5]. In the followings only the main steps are briefly presented. The TP model transformation is a numerical method and has three key steps. The first step is the discreatisation of the given $\mathbf{S}(\mathbf{p}(t))$ via the sampling of $\mathbf{S}(\mathbf{p}(t))$ over a huge number of points $\mathbf{p} \in \Omega$, where Ω is the transformation space. The sampling points are defined by a dense hyper rectangular grid. In order to loose minimal information during the discretisation we apply as dense grid as possible. The second step extracts the LTI vertex systems from the sampled systems. This step is specialized to find the minimal number of LTI vertex systems as the vertex points of the tight convex hull of the sampled systems. The third step constructs the TP model based on the LTI vertex systems obtained in the second step. It defines the continuous basis functions to the LTI vertex systems.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

5.1.2. Determination of the convex TP model form of the aeroelastic model

We execute the TP model transformation on the model (9). We used the following parameters: $b = 0.135m$; $span = 0.6m$; $k_h = 2844.4N/m$; $c_h = 27.43Ns/m$; $c_\alpha = 0.036Ns$; $\rho = 1.225kg/m^3$; $c_{l_\alpha} = 6.28$; $c_{l_\beta} = 3.358$; $c_{m_\alpha} = (0.5+a)c_{l_\alpha}$; $c_{m_\beta} = -0.635$; $m = 12.387kg$; $x_\alpha = -0.3533 - a$; $I_\alpha = 0.065kgm^2$; $c_\alpha = 0.036$.
First of all, according to the three steps of the TP model transformation, let us define the transformation space Ω. We are interested in the interval $U \in [14,25](m/s)$ and we presume that the interval $\alpha \in [-0.2,0.2](rad)$ is sufficiently large enough. Therefore, let: Ω $[14,25] \times [-0.1,0.1]$ in the present example (note that these intervals can arbitrarily be defined). Let the grid density be defined as $M_1 \times M_2$, $M_1 = 300$ and $M_2 = 300$. Step 2 of the TP model transformation yields 6 vertex LTI systems:

$$\mathbf{A}_{1,1} = 10^3 \begin{pmatrix} 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \\ -0.2314 & -0.0095 & -0.0034 & -0.0001 \\ 0.2780 & -1.1036 & 0.0071 & -0.0000 \end{pmatrix} \quad \mathbf{B}_{1,1} = \begin{pmatrix} 0 \\ 0 \\ -8.5825 \\ -32.4370 \end{pmatrix}$$

$$\mathbf{A}_{2,1} = \begin{pmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -231.3804 & -46.3063 & -4.3776 & -0.2573 \\ 277.9906 & -966.7931 & 10.6520 & 0.4104 \end{pmatrix} \quad \mathbf{B}_{2,1} = \begin{pmatrix} 0 \\ 0 \\ -27.3677 \\ -103.4344 \end{pmatrix}$$

$$\mathbf{A}_{3,1} = 10^3 \begin{pmatrix} 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \\ -0.2314 & -0.0227 & -0.0039 & -0.0002 \\ 0.2780 & -1.0543 & 0.0089 & 0.0002 \end{pmatrix} \quad \mathbf{B}_{3,1} = 10^3 \begin{pmatrix} 0 \\ 0 \\ -0.0154 \\ -0.0580 \end{pmatrix}$$

$$\mathbf{A}_{1,2} = \begin{pmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -231.3804 & -16.5786 & -3.4333 & -0.1425 \\ 277.9906 & 23.0842 & 7.1447 & -0.0157 \end{pmatrix} \quad \mathbf{B}_{1,2} = \begin{pmatrix} 0 \\ 0 \\ -8.5825 \\ -32.4370 \end{pmatrix}$$

$$\mathbf{A}_{2,2} = \begin{pmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -231.3804 & -53.4094 & -4.3776 & -0.2573 \\ 277.9906 & 159.8695 & 10.6520 & 0.4104 \end{pmatrix} \quad \mathbf{B}_{2,2} = \begin{pmatrix} 0 \\ 0 \\ -27.3677 \\ -103.4344 \end{pmatrix}$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$\mathbf{A}_{3,2} = \begin{pmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -231.3804 & -29.8524 & -3.9054 & -0.1999 \\ 277.9906 & 72.3823 & 8.8983 & 0.1974 \end{pmatrix} \quad \mathbf{B}_{3,2} = \begin{pmatrix} 0 \\ 0 \\ -15.3526 \\ -58.0244 \end{pmatrix}$$

The third steps results in basis functions $w_{1,i}(U)$ and $w_{2,j}(\alpha)$ depicted in Figure 2. When we numerically check the error between the model (9) and the resulting TP model, we find that the error is about 10^{-11} that is caused by the numerical computation.

In conclusion, the aeroelastic model (9) can be described exactly in finite convex TP form of 6 vertex LTI models, also see [2]. Note that, one may try to derive the basis functions analytically from (9). The basis functions of α can be extracted from $k_\alpha(\alpha)$. Finding the basis functions of U, however, seems to be rather complicated. In spite of this, the computation of the TP model transformation takes a few seconds.

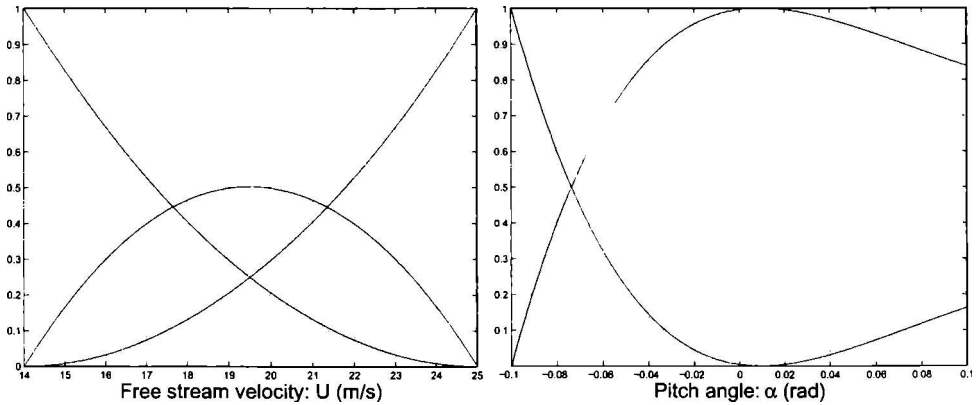


Figure 2: Basis functions on the dimensions U and α.

5.2. Observer design to the prototypical aeroelastic wing section

5.2.1. Method for observer design under PDC framework

In reality not all the state variables are readily available in most cases. Unavailable state variables should be estimated in the case of state-feedback control strategy. Under this circumstances, the question arises whether it is possible to determine the state from the system response to some input over some period of time. Namely, the

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

observer is required to satisfy:

$$\mathbf{x}(t) - \hat{\mathbf{x}}(t) \to 0 \quad \text{as} \quad t \to \infty,$$

where $\hat{\mathbf{x}}(t)$ denotes the state vector estimated by the observer. This condition guaranties that the steady-state error between $\mathbf{x}(t)$ and $\hat{\mathbf{x}}(t)$ converges to 0. We use the following observer structure:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(\mathbf{p}(t))\hat{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) + \mathbf{K}(\mathbf{p}(t))(\mathbf{y}(t) - \hat{\mathbf{y}}(t))$$

$$\hat{\mathbf{y}}(t) = \mathbf{C}(\mathbf{p}(t))\hat{\mathbf{x}}(t),$$

That is in TP model form:

$$\dot{\hat{\mathbf{x}}}(t) = \mathcal{A} \underset{n}{\otimes} \mathbf{w}(p_n(t))\hat{\mathbf{x}}(t) + \mathcal{B} \underset{n}{\otimes} \mathbf{w}_n(p_n(t))\mathbf{u}(t) + \tag{10}$$

$$+ \mathcal{K} \underset{n}{\otimes} \mathbf{w}(p_n(t))(\mathbf{y}(t) - \hat{\mathbf{y}}(t))$$

$$\hat{\mathbf{y}}(t) = C \underset{n}{\otimes} \mathbf{w}(p_n(t))\hat{\mathbf{x}}(t).$$

At this point we should emphasize that in our example the vector $\mathbf{p}(t)$ does not contain values form the estimated state-vector $\hat{\mathbf{x}}(t)$, since $p_1(t)$ equals U and $p_2(t)$ equals the pitch angle ($x_2(t)$). These variables are observable. We estimate only state-values $x_3(t)$ and $x_4(t)$. Consequently the goal, in the present case, is to determine gains in tensor $\mathcal{K}$ for (5.1). For this goal the following LMI theorem can be find in [22]. Before dealing with this LMI theorem, we introduce a simple indexing technique, in order, to have direct link between the TP model form (3.4) and the typical form of LMI formulations:

Method 1. (Index transformation) Let

$$\mathbf{S}_r = \begin{pmatrix} \mathbf{A}_r & \mathbf{B}_r \\ \mathbf{C}_r & \mathbf{D}_r \end{pmatrix} = \mathbf{S}_{i_1,i_2,..,i_N},$$

where $r = ordering(i_1, i_2, .., i_N)$ ($r = 1..R = \prod_n I_n$). The function "ordering" results in the linear index equivalent of an N dimensional array's index $i_1, i_2, .., i_N$, when the size of the array is $I_1 \times I_2 \times \quad \times I_N$. Let the basis functions be defined according to the sequence of r:

$$w_r(\mathbf{p}(t)) = \prod_n w_{n,i_n}(p_n(t)).$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

Theorem 1. (Globally and asymptotically stable observer)
In order to ensure

$$\mathbf{x}(t) - \hat{\mathbf{x}}(t) \to 0 \quad \text{as} \quad t \to \infty,$$

in the observer strategy (5.1), find $\mathbf{P} > 0$ and $\mathbf{N}_r$ satisfying equ.

$$-\mathbf{A}_r^T\mathbf{P} - \mathbf{P}\mathbf{A}_r + \mathbf{C}_r^T\mathbf{N}_r^T + \mathbf{N}_r\mathbf{C}_r > 0 \tag{11}$$

for all r and

$$-\mathbf{A}_r^T\mathbf{P} - \mathbf{P}\mathbf{A}_r - \mathbf{A}_s^T\mathbf{P} - \mathbf{P}\mathbf{A}_s + \tag{12}$$
$$+\mathbf{C}_r^T\mathbf{N}_s^T + \mathbf{N}_s\mathbf{C}_r + \mathbf{C}_s^T\mathbf{N}_r^T + \mathbf{N}_r\mathbf{C}_s > 0.$$

for $r < s \leq R$, except the pairs (r,s) such that $w_r(\mathbf{p}(t))w_s(\mathbf{p}(t)) = 0, \forall \mathbf{p}(t)$.

Since the above equations are LMI's with respect to variables $\mathbf{P}$ and $\mathbf{N}_r$, we can find a positive definite matrix $\mathbf{P}$ and matrix $\mathbf{N}_r$ or determine that no such matrices exist. This is a convex feasibility problem. Numerically, this problem can be solved very efficiently by means of the most powerful tools available in the mathematical programming literature e.g. MATLAB-LMI toolbox [11].
The observer gains can then be obtained as:

$$\mathbf{K}_r = \mathbf{P}^{-1}\mathbf{N}_r. \tag{13}$$

Finally, by the help of $r = ordering(i_1, i_2, .., i_N)$ in Method 1 one can define $\mathbf{K}_{i_1,i_2,..,i_N}$ from $\mathbf{K}_r$ obtained in (5.4) and store into tensor $\mathcal{K}$ of (5.1).

5.2.2. Observer design to the prototypical aeroelastic wing section

This section applies Theorem 1 to the TP model of the aeroelastic wing section. We define matrix $\mathbf{C}$ for all r from:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t),$$

that is in present case:

$$\mathbf{C}_r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The LMIs of Theorem 1, applied to result of the TP model transformation, are feasible and yields 6 observer feedbacks:

$$\mathbf{K}_{1,1} = 10^3 \begin{pmatrix} 0.0000 & 0.0001 \\ 0.0001 & 0.0008 \\ -0.0432 & 0.0001 \\ 0.5674 & -3.8791 \end{pmatrix}$$

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

$$\mathbf{K}_{2,1} = 10^3 \begin{pmatrix} 0.0000 & 0.0002 \\ 0.0001 & 0.0008 \\ -0.0429 & -0.0356 \\ 0.5670 & -3.7432 \end{pmatrix}$$

$$\mathbf{K}_{3,1} = 10^3 \begin{pmatrix} 0.0000 & 0.0001 \\ 0.0001 & 0.0008 \\ -0.0430 & -0.0127 \\ 0.5672 & -3.8302 \end{pmatrix}$$

$$\mathbf{K}_{1,2} = 10^3 \begin{pmatrix} 0.0000 & -0.0001 \\ 0.0002 & 0.0010 \\ -0.0430 & -0.0442 \\ 0.5677 & 2.8271 \end{pmatrix}$$

$$\mathbf{K}_{2,2} = 10^3 \begin{pmatrix} 0.0000 & -0.0001 \\ 0.0002 & 0.0010 \\ -0.0431 & -0.0785 \\ 0.5672 & 2.9650 \end{pmatrix}$$

$$\mathbf{K}_{3,2} = 10^3 \begin{pmatrix} 0.0000 & -0.0001 \\ 0.0002 & 0.0010 \\ -0.0430 & -0.0566 \\ 0.5675 & 2.8768 \end{pmatrix}$$

In conclusion the state values $x_3(t)$ and $x_4(t)$ are estimated by (5.1) as:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(\mathbf{p}(t))\hat{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p}(t))u(t) +$$

$$\left(\sum_{i=1}^{3} \sum_{j=1}^{2} w_{1,i}(U) w_{2,j}(\alpha) \mathbf{k}_{i,j} \right) (\mathbf{y}(t) - \hat{\mathbf{y}}(t)),$$

where

$$\mathbf{y}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \quad \text{and} \quad \hat{\mathbf{y}}(t) = \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix} \quad \text{and} \quad \mathbf{p}(t) = \begin{pmatrix} U \\ \alpha \end{pmatrix},$$

($x_1(t) = h$, plunge, and $x_2(t) = \alpha$, pitch). In order to demonstrate the accuracy of the observer, numerical experiments are presented in the next section.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

5.2.3. *Simulation results*

We simulate the observer for initials $\mathbf{x}(0) = \begin{pmatrix} 0.01 & 0.1 & 0.1 & 0.1 \end{pmatrix}^T$ and $\hat{\mathbf{x}}(0) = \begin{pmatrix} -0.01 & -0.1 & -0.1 & -0.1 \end{pmatrix}^T$, for the open loop case. Figure 3 shows how the observer is capable of converging to the unmeasurable state values $x_3(t)$ and $x_4(t)$.

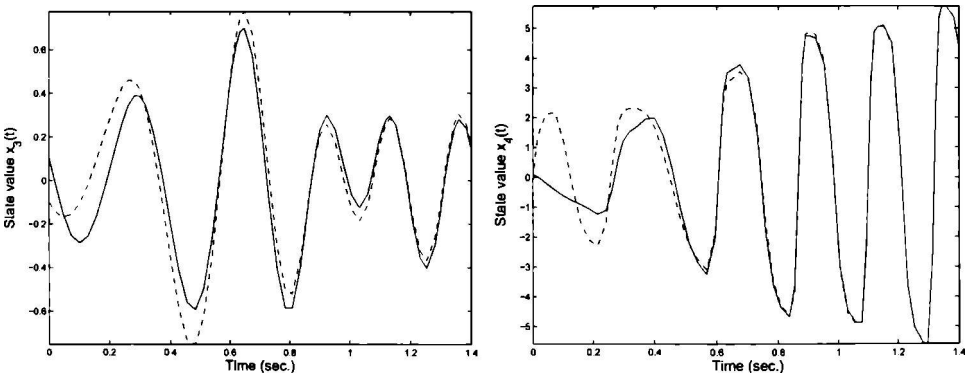


Figure 3: State values $x_3(t)$, $x_4(t)$ (solid line) and estimated values $\hat{x}_3(t)$, $\hat{x}_4(t)$ (dashed line) for open loop response.
($U = 20m/s$, $a = -0.4$, initials: $\mathbf{x}(0) = \begin{pmatrix} 0.01 & 0.1 & 0.1 & 0.1 \end{pmatrix}^T$
$\hat{\mathbf{x}}(0) = \begin{pmatrix} -0.01 & -0.1 & -0.1 & -0.1 \end{pmatrix}^T$)

6. CONCLUSION

First message of the paper is that the TP model transformation method under the PDC design framework can be used for observer design in the same way as for controller design. The second message is that the paper shows how to determine observer for the prototypical aeroelastic wing section.

REFERENCES

[1] BARANYI, P.: TP *model transformation as a way to* LMI *based controller design*, IEEE Transaction on Industrial Electronics, **51** (2004), No. 2.

[2] BARANYI, P., KORONDI, P., PATTON, R., AND HASHIMOTO, H.: *Global asymptotic stabilisation of the prototypical aeroelastic wing section via tp model transfromation*, Asian Journal of Control, **(to be printed in Vol. 7, No. 2, 2004)**.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[3] BARANYI, P., MICHEKBERGER, P., AND VÁRKONYI-KÓCZY, A.: *Numerical control design for aeroelastic systems*, in: *2nd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence (SAMI 2004)*, Herl'any, Slovakia, 16-17 January, 2004 pp. 43–50.

[4] BARANYI, P. AND PATTON, R.: *A numerical control design method for prototypical aeroelastic wing section with structural non-linearity*, in: *European Control Conference (ECC'03)*, University of Cambridge, UK, 2003

[5] BARANYI, P., TIKK, D., YAM, Y., AND PATTON, R. J.: *From differential equations to* PDC *controller design via numerical transformation*, Computers in Industry, Elsevier Science, **51** (2003), 281–297.

[6] BLOCK, J. J. AND STRGANAC, T. W.: *Applied active control for nonlinear aeroelastic structure*, Journal of Guidance, Control, and Dynamics, **21** (1998), No. 6, 838–845.

[7] BOYD, S. GHAOUI, L. E. FERON, E., AND BALAKRISHNAN, V.: *Linear matrix inequalities in system and control theory*, Philadelphia PA:SIAM, (1994).

[8] DOWELL, E. H.: *Nonlinear aeroelasticity*, Proc. of the 31th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 97-1024, (1990), 1497–1509.

[9] (EDITOR), E. H. D., CURTISS, H. C. J., SCANLAN, R. H., AND SISTO, F.: *A Modern Course in Aeroelasticity*, Stifthoff and Noordhoff, Alpen aan den Rijn, The Netherlands, 1978.

[10] FUNG, Y. C.: *An Introduction to the Theory of Aeroelasticity*, John Wiley and Sons, New York, 1955.

[11] GAHINET, P., NEMIROVSKI, A., A.J.LAUB, AND M.CHILALI: LMI *Control Toolbox*, The MathWorks, Inc., 1995.

[12] KO, J., KRIDULA, A. J. AND STRGANAC, T. W.: *Nonlinear control theory for a class of structural nonlinearities in a prototypical wing section*, Proc. of the 35th AIAA Aerospace Scinece Meeting and Exhibit, AIAA paper 97-0580, (1997).

[13] KO, J., KURDILA, A. J., AND STRGANAC, T. W.: *Adaptive feedback linearization for the control of a typical wing section with structural nonlinearity*, Nonlinear Dynamics, (1997).

[14] KO, J., KURDILA, A. J., AND STRGANAC, T. W.: *Nonlinear control theory for a class of structural nonlinearites in a prototypical wing section*, AIAA Journal of Guidance, Control, and Dynamics, **20** (1997), No. 6, 1181–1189.

[15] KO, J., KURDILA, A. J., AND STRGANAC, T. W.: *Nonlinear dynamics and control for a structurally nonlinear aeroelastic system*, Proc. of the 38th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 97-1024, (1997).

[16] LATHAUWER, L. D., MOOR, B. D., AND VANDEWALLE, J.: *A multi linear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications, **21** (2000), No. 4, 1253–1278.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM

[17] LEE, B. H. K. AND LEBLANC, P.: *Flutter analysis of two dimensional airfoil with cubic nonlinear restoring force*, National Aeronautical Istbalishment, Aeronautical Note-36, National Research Council, Ottava, Canada, (1986), No. 25438.

[18] O'NEIL, T. GILLIAT, H. C., AND STRGANAC, T. W.: *Investigatiosn of aeroelsatic response for a system with continuous structural nonlinearities*, Proc. of the 37th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 96-1390, (1996).

[19] O'NEIL, T. AND STRGANAC, T. W.: *An experimental investigation of nonlinear aeroelastic respons*, AIAA Journal of Aircraft (accepted).

[20] PRICE, S. J., ALIGHANBARI, H., AND LEE, B. H. K.: *The aeroeloasric response of a two dimensional airfoil with bilinear and cubic structural nonlinearities*, Proc. of the 35th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 94-1646, (1994), 1771–1780.

[21] PRICE, S. J., ALIGHANBARI, H., AND LEE, B. H. K.: *Postinstability behavior of a two dimensional airfoil with a structural nonlinearity of aircraft*, Journal of Aircraft, **31** (1994), No. 6, 1395–1401.

[22] TANAKA, K. AND WANG, H. O.: *Fuzzy Control Systems Design and Analysis - A Linear Matrix Inequality Approach*, Hohn Wiley and Sons, Inc., 2001, 2001.

[23] TANG, D. M. AND DOWELL, E. H.: *Flutter and stall response of a helicopter blade with structural nonlinearity*, Journal of Aircraft, **29** (1992), 953–960.

[24] TANG, D. M. AND DOWELL, E. H.: *Comparison of theory and experiment for nonlinear flutter and stall response of a helicopter blade*, Journal of Sound and Vibration, **165** (1993), No. 2, 251–276.

[25] TIKK, D., BARANYI, P., PATTON, R., AND TAR, J.: *Approximation capability of* TP *model forms*, Australian Journal of Intelligent Information Processing Systems (accepted for publication), (2004).

[26] TIKK, D., BARANYI, P., AND R.J.PATTON: *Polytopic and* TS *models are nowere dense in the approximation model space*, IEEE Int. Conf. System Man and Cybernetics (SMC'02), (2002), proc. on CD.

[27] YANG, Z. C. AND ZHAO, L. C.: *Analysis of limit cycle flutter of an airfoil in incompressible flow*, Journal of Sound and Vibration, **123** (1988), No. 1, 1–13.

[28] ZHAO, L. C. AND YANG, Z. C.: *Chaotic motions of an airfoil with nonlinear stiffness in incompressible flow*, Journal of Sound and Vibration, **138** (1990), No. 2, 245–254.

DIGITALIZÁLTA: MISKOLCI EGYETEM KÖNYVTÁR, LEVÉLTÁR, MÚZEUM