RAIRO-Oper. Res. 48 (2014) 303–324
DOI: 10.1051/ro/2014009

RAIRO Operations Research
www.rairo-ro.org

PROCESS PARAMETER PREDICTION *VIA* MARKOV MODELS OF SUB-ACTIVITIES

Lino G. Marujo[1] and Raad Y. Qassim[2]

Abstract. This work aims to fill a lacunae in the project-oriented production systems literature providing a formal analytic description of the rework effects formulae and the determination of the extended design time due to a certain degree of overlapping in a pair of activities. It is made through the utilization of concepts of workflow construction with hidden (semi) Markov models theory and establishing a way to disaggregate activities into sub-activities, in order to determine the activity parameters used by the project scheduling techniques. With the aim to make a correlation between the entropy of the state transitions and the probability of changes, the information theory is also used, and the concept of impact caused by the probability of changes is provided. Numerical examples are shown for the purpose to demonstrate the applicability of the concepts developed, and one example of overlapping of two activities is shown. The original contributions of this work are shown on the last section.

Keywords. Activity parameters, sub-activities Markov model, entropy, project scheduling parameters, rework estimation.

Mathematics Subject Classification. 90B15, 90B30, 68M20.

1. Introduction

The increasing importance of agility and flexibility to time-to-market required by the companies, has been reflected on the growth in the product development and

Received August 21, 2013. Accepted January 6, 2014.

[1] Department of Industrial Engineering, POLI, Federal University of Rio de Janeiro, CP 64548, 21941-972 Rio de Janeiro, Brazil. lgmarujo@ufrj.br

[2] Department of Ocean Engineering, COPPE, Federal University of Rio de Janeiro, CP 68508, 21941-972 Rio de Janeiro, Brazil

Article published by EDP Sciences

manufacturing processes' speed up techniques, in order to deliver new products or changing existing products faster than the competition, forcing the companies to establish a way to produce ordered products more efficiently and fast. These characteristics are more latent on engineer-to-order environment, where the starting mechanism to execution or manufacturing, is the execution agreement, instead of a replenishment point in inventory records, for example certain types of industries such as: aeronautics, naval or civil construction. Hence, new methods of project schedule acceleration have risen as overlapping and crashing. For example, as the development of concepts is actually finished before the beginning of construction, a certain degree of overlapping of these two activities results in the shortening of the total project lead-time. Therefore, starting the execution phase before the total completion of concept development, should also result in an augmented number of changes and, consequently, productivity looseness, increasing the costs and spending additional time to accommodate possible changes required. This latent rework necessity appears on a low level activity, or detailed one, where this activity is broken into small parts, called sub-activities, forming a workflow model, which describes the sequences and the dependence path of tasks, in an atomic view of the project [31], where the activity parameters can be evaluated, before and after the application of acceleration techniques.

1.1. Work outline

This work is organized as follows: in the second section, is explained the concepts of the transition of the activity level to sub-activity one, making an extended review of literature concerning the hidden (semi) Markov models applied to workflow constitution, followed by a review of the concepts of workflow model's construction under some type of restrictions. And finally, a review on the state-of-art of the overlapping strategy to shrink project lead-time is conducted. The problem statement is delineated in the third section. The fourth section treats the relationship existing between the macro level and the micro level of activities and are applied the concepts of hidden semi-Markov models to establish a way to make this transition. An algorithm is proposed to determine the best hidden semi-Markov model that describes the workflow, and the calculations of activity parameters. In the fifth section, a stochastic model to evaluating the rework fraction of a pair of overlapped activities is developed. In the sixth section, a numerical example is provided to demonstrate the validity of proposed model and its results. In the last section, some conclusions and statements are done.

2. Literature review

2.1. The transition of the activity level to the sub-activity level

The organizations, nowadays, have to manage their business process as a project-oriented view, therefore, a question arose from this procedure: how to

manage what is occurring on an operational level of each activity on projects-oriented process? Based on this question, a number of methods have been applied to aggregate and disaggregate activities to support the high level of management while managing the projects *via* milestones, or critical events to the development of these projects. However, a macro vision of the process conducts to an increased probability of hiding imperfections and deviations on the execution of the tasks, incurring on overruns of costs and time. These deviations do not occur on an aggregate level of management, excepting those caused by management errors, rather, they occur due to deviations of the execution level sub-activities's level on the operational level, which can be represented through workflow models. Moreover, to do this evaluation, a method to disaggregate and further aggregate activities' parameters must be established to support a better project-oriented process management.

An approach to perform that analysis described above, is to schedule all activities simultaneously, but, to large project-oriented systems, this approach is computationally prohibitive [16]. In this way, researchers have been explored approaches of activities' aggregation and disaggregating to reduce the size of allocation problems.

Agrawal and Gunopulos [1] presented an approach for a system that constructs the process from logs of past, unstructured executions of the given process. The graph produced, conforms to the dependencies and path executions present in the log. By providing models that capture the previous executions of the process, this technique allows an easier introduction of a workflow system, also the evaluation and evolution of the existing process models. Hence, the detailed level of an activity is modeled as a hidden Markov model (HMM), being represented as an acyclic graph $G = \{V, E\}$ with each node $v \in V$ being the emission symbol of the hidden state, representing the tasks that are being performed, and the set of edges $e \in E$, representing the state symbols transitions. Each sub-activity i is allocated exactly to one activity, namely $i \in p$. The activities are decomposed on sub-activities, with a source node and a finish node, both added to observed emission, establishing the beginning and the end of the activity, following a workflow that will deliver the same output required for the project activity.

Furthermore, the latent necessity of rework is shown only on a basic level, *i.e.* sub-activities, organized into workflows, describing the sequence and the interdependency of these, in an atomic view of the project [31], because the nature of the process interdependency [34] and the randomness of the symbol emissions. A hidden Markov model can be defined as:

Definition 2.1. Let be any set of states, not directly observed $(k_1, \ldots, k_n)$, with $k_1 k_2 \ldots k_n$, so $P[S_{k_n} \leqslant a_n | S_{n_1} = a_1, S_{k_2} = a_2 \ldots S_{k_{n-1}} = a_{n-1}] = P[S_{k_n} \leqslant a_n | S_{k_{n-1}} = a_{n-1}]$, and a set of symbols $V = \{v_i, \ldots, v_n\}$, each of them with it inherent probability of emission $B = (b_i(K_i))$, the stochastic process is called a Hidden Markov Model.

Therefore, the sub-activities workflow is not, actually, a linear series of states, but it can contain variations due to operational constraints that the workers or the

TABLE 1. Correlation between workflow model and HMM.

Workflow model	HMM	Mathematical representation
Sub-activity Node	State	K
Type of task performed	Emitted symbol	s
Graph (event trace)	Set of sequences (realizations observed) of emitted symbols	E
Task duration†	State durations	δ

† In models where the durations cannot be determined, it is utilized the semi-markovian model, *i.e.*, where the duration is a probability density function, actually following a triangular distribution.

actors faced at the moment of performing these tasks in the shop floor. Herbst [18] explains the reason of the difficult to obtain a workflow model that describes one operation is due to the fact of the knowledge needed to define it is spread throughout the organization. It is recorded into the actor's minds whom are actively involved with the execution of sub-activities, and these actors, focused on the shop floor activities and targets, have no time to spend on making a formal modeling of operations workflows.

The hidden Markov models (HMM), as introduced by Rabiner [32], are characterized by a stochastic state sequence, $K = \{1, \ldots, K_n\}$, where the probability of each state is dependent only on the immediately previous event, and the states are not directly observed, but through symbols that characterize it. The exact instant that this state occurs is hidden, *i.e.*, there is a probability of the state's occurrence into a determined time interval, associated to symbols emitted by each state, and the transitions probabilities between them. On Table 1 can be observed the existing relations between a workflow model and an HMM [20].

An HMM can be visualized as an oriented graph, with the nodes describing the states with the emitted symbols, and the edges representing the state symbol transitions. Hence, the behavioral aspects of sequential workflow models (which are the atomic views of the project activities), can be mapped as an HMM, where the model will describe the flow in which the work is performed (see [5,19,20,38]).

The relationship between a node k_i and the respective task v_i, is deterministic in each branch of the graph, but in the final workflow model, there are emission probabilities represented by the emission probability $b_i(k_i)$. However, to take into account the called multiple node problem, the alike task could occur more than once into the workflow model, *i.e.*, due to unnecessary forced repetitions, reworks, lack of information, and so on.

Although the HMM can be a simple and efficient model to identify sequential data, authors suggest some limitations when the activities become more complex, or the activities show a *long term temporal* dependency [12,27], but it is not relevant to the present study, because the short term of the activities studied. To deal

with these limitations, two extensive classes of HMM have been proposed. The first one introduces complementary models to the basics HMM with a hierarchical structure, with the purpose to explore the natural hierarchic organization of the human being (*e.g.* see [7,13,26]). The second extension, adopts semi-Markov models and introduces their hidden variables, namely the duration probability of each state [28,30]. In these models, is assumed that a state remains unaltered by any random time duration before changes to another state. For each state, a duration probability distribution is given to characterize the behavior of such duration.

2.2. Constrained workflow models of sub-activities

On another hand, these techniques are complemented by the introduction of some constraints on the execution processes of sub-activities. Many workflow applications often have time constraints such that each processing of a workflow of sub-activities needs to be finished within its deadline. Son and Kim [37] address a suitable scheme that can maximize the number of workflow instances satisfying the given deadline. They first present a method to find out a set of critical activities where an activity is one whose delay of completion directly affects the overall processing time of a workflow. Since each critical activity has a certain number of servers to be processed, for the sufficient processing capacity, we then develop a method to determine the minimum number of servers for the critical activity such that this activity should be finished without delay for a given input arrival rate. Li and Fan [25] provide a range of six-time constraints and the method for identify the critical path of a workflow process is given accordingly. However, the constraints of a workflow model of sub-activities are not restricted to time constraints. Crampton [10] provides a model of constrained workflow systems and develops a systematic algebraic method for combining constraints and authorization information. van Hee *et al.* [17] investigate a resource management policy that allocates resources based on the number of available resources only, and formulate a condition on resources requesting process, called solidity based on the use of Petri Nets.

Another kind of constraints can be seen in [36] where the HMM represents a robot navigation plan, with a framework that incorporating readily available odometric information and geometrical constraints. The automated workflow composition is analyzed by [11, 42] where they propose the use of Markov Decision Processes (MDPs) to model workflow composition, to account for the uncertainty over the environmental model.

2.3. Overlapping strategy of shortening project lead-time

The overlapping strategy of activities with a view to reducing the project lead time has been studied in the context of project scheduling and new-product development. Browning *et al.* [6] provide a review of work on modeling product development process (see also [3]). Krishnan [23] provides a framework to help designers or managers to decide *when* and *how* to overlap the activities reducing

product development lead time while ensuring that the adverse effects on product quality and development effort are minimized, and presents a way to determine how to disaggregate design information and overlap consecutive stages based on the evolution and sensitivity properties of the information exchanged [24].

The information dependencies between development tasks constitute the information-processing view of the development processes, and can be modeled as a Markov chain [2] and arranged into a Design Structure Matrix [9, 27]. The overlapping strategy differs from the sequential approach in that it allows the downstream project stages to start before preceding upstream stages have finalized their works [3]. In the way to make the project faster and cheaper, the managers have noticed important advances in project management, and one of the most useful and popular technique is overlapping. As a result, the duration of individual activities actually increases through overlapping, while the total project lead-time decreases because working concurrently on different activities. Thereby, overlapping utilizes incomplete information, it requires that project stages start their work assuming a certain amount of work done with a quality lower than what was specified, forcing some stages of the system to be reworked, which is often needed to accommodate unforeseen upstream stages. Time-cost trade-offs are extensively discussed in the project schedule literature, where activities can be shortened (crashed) at additional costs. Because both crashing of activities and overlapping aim reducing completion times, they can be considered to alter nativities or complements to each other.

Nicoletti and Nicolo [31] developed a linear programming model with a view to maximizing information flow in concurrent engineering projects. Chakravarty [8] makes analysis of single and multiple overlap properties and its impacts on cost functions.

Ford and Sterman [14] state that concurrent development not only increases the vulnerability of projects to changes and errors requiring to rework, but also increases the fraction of work released that will require changes. Roemer and Ahmadi [33] present a cost minimization model for the simultaneous crashing and overlapping of activities in a project consisting of activities in series, analyzing the impact of different evolution/sensitivity parameters. Zhang, Qiu and Zhang [41] establish a method to measure the coupled strength of tasks and to calculate the gross workload, determining the best sequence of coupled tasks based on task output influence ratio, parameter change ratio and parameter feedback.

Gerk and Qassim [15], provide a mixed integer nonlinear programming model for the acceleration of projects, employing the simultaneous crashing, overlapping, and substitution of project activities, with the assumption that the rework fraction caused by overlapping rates was previously known.

2.4. Information flow

In the Information Theory context, the idea of information flow aims to measure the amount of information that flows from a state X to a state Y during to the

execution of a certain process. In a general view, the presence of a problem is an uncertainty state. Hence, a problem can be treated as a term of its inherent uncertainty, defined as the uncertainty before it and the uncertainty after it.

The process uncertainty of information is expressed by $H(X, Y)$, *i.e.* the conjoint entropy and can be determined as defined by Shannon [35] as:

$$H(X, Y) = -\sum_{x, y} p_{x, y} \log(p_{x, y}).$$

The entropy measures the internal degree of ordination in the message's structure produced by the state X to state Y. As greater as these ordination, as low will be the randomness, and, therefore, lowest the H value. Hence, the information flow is characterized by the communication channel, the intrinsic information flowed into these channel and its entropy measure. The communication channel can be represented by a communication network topology, or by a minor part of it, *i.e.* a sub-net, actually described by an oriented graph $G = \{V, E\}$, where V is the set of vertices and E is the set of arcs, defining the information flow directions and the nodes where it was generated or where it must reach [2].

3. Problem statement

In this Section, a precise problem statement is provided for a pair of problems under consideration. It is assumed that an activity may be described within a triangular distribution pattern at the macroscopical level, with which a number of stochastic sub-activities realisations may be associated at the microscopic level. The principal objective is the prediction of activity attributes at the macroscopic level, such as duration and rework times, from probabilistic properties of the constituent sub-activity set at the microscopic level. In the following, the framework presented by Nicoletti and Nicolò [31] is adopted for the decomposition of an activity into its constituent sub-activities, with one important difference: whilst the authors assume that for each activity there exists one and only one set of sub-activities, in this paper this assumption is relaxed by allowing a set of microscopic realisations of sub-activities for each macroscopic activity.

3.1. Single activity duration

Given an activity at the macroscopic level and the corresponding sub-activity constituent set and its realization at the microscopic level, find the maximum, minimum, and the most probable duration time of the activity at the macroscopic level.

3.2. Rework time due to activity overlapping

Given a pair of activities and the overlapping times of the activity pair at the macroscopic level, and the corresponding respective constituent sets and their

realizations at the microscopic level, find the maximum, minimum, and most probable rework time due to the overlapping of the activity pair.

3.3. Premises

There are some premises adopted in this work, as follows:

- The activities on a macro level, have deterministic characteristics.
- Each activity can only be concluded, when a certain number of discrete states has been completed, which are the sub-activities, but they are not observable in a direct way, but through a set of emitted symbols, characterizing a Hidden Semi-Markov Model.
- The duration time of each sub-activity is a weighted average, defined as the relation with the number of emitted symbols, which characterize the states, and has three dimensions: optimistic, most likely and pessimistic.
- The hidden semi-Markov model of the downstream activity does not affect the upstream activity, indeed, there is no feedback influencing the probability of change, nor the impact of change into the current aggregate activity.
- The overlapping only occurs between two adjacent activities.
- The physical and financial resources are unbounded, but they constitute an important area of study of project management theory.

4. Single activity model

4.1. Notations

- p denotes the aggregate upstream activity;
- q denotes the aggregate downstream activity;
- i denotes the disaggregate anterior task;
- j denotes the disaggregate posterior task;
- $K = \{1, \ldots, |K_n|\}$ denotes the set of (hidden) states;
- $V = \{v_1, \ldots, v_n\}$ denotes the set of tasks performed or emissions symbols;
- $A = (a_{ij}) = P[S_t + 1 = k_j | S_t = k_i]$, with $1 \leqslant i, j \leqslant N$ denotes the matrix of transitions probabilities;
- $B = (b_i(k)) = P[\text{emission of } V_k \text{ on time } t | S_t = K_i]$, with $1 \leqslant i \leqslant N$ are the probabilities of emitting each symbol;
- $\pi = \pi_i$, $\pi_i = P[S_i = k_i]$ with $1 \leqslant i \leqslant N$ are the initial probabilities;
- S_t denotes the state on time t;
- $P(V|M)$ denotes the probability of the sequence V follow the model M;
- $P^*(V|M)$ denotes the most likely sequence path of V in the model M;
- $G = \{V, E\}$ denotes the graph G with the edges $E = \{e_{1,2}, \ldots, e_{n-1,n}\}$ and the nodes V, representing the set of accomplishments observed;
- $\beta_{\text{inc}} = \{V, E\}$ denotes the incidence matrix of graph G, where:

$$\exists e_k = (i, j) \text{if } \{e_{i,k} = +1, e_{r,k} = 0, \forall r \neq i, j, e_{j,k} = -1$$

- $Dur(V|M) = (d_{ij})$ denotes the duration parameters of the sequence of tasks;
- $NumObs$ denotes the number of observed sequences of symbols related to the sub-activities level;
- τ_p denotes the completion state of activity p;
- T_p denotes the total length of time of activity p;
- δ_i denotes the duration of each task, in a micro level;
- D_p denotes the state which task i is completed;
- p_{ij} denotes the probability of change for each state transition $e_{i,j}$, related to the entropy of the process;
- $s_{ij}(K_i)$ represents the amount of impact due to changes in the previous task i, in amount of time;
- Edt_q denotes the amount of extended design time, in time units;
- L_q denotes the fraction of rework needed in activity q, due to overlapping.

On an elementary level, where the activities are detailed, described as sub-activities i, j, even though hidden to the project managers, it follows a strict precedence relation, following a defined order, in which the set of tasks when aggregated represents the activity on a macro level, called activity p, q.

Assumption 4.1. The relationship between the activity and the sub-activities, exists when these ones have in sub-activities in a determined order, produces the same targeted output, and, for this instance, the work only flows to another activity if the output of that sub-activity has had reached.

It must be observed, however, that in the environment described previously, there is no overlapping of activities p, q, the view is focused on one project activity isolated.

Assumption 4.2. The macro activity is deterministic and can contain various sub-activities, but one sub-activity only can be allocated into one macro activity.

The detailed level of an activity is modeled as a hidden Markov model (HMM), being represented as an acyclic graph $G = \{V, E\}$ with each node $v \in V$ being the emission symbol of the hidden state, representing the tasks are being performed, and the set of edges $e \in E$, representing the state symbol transitions. Each sub-activity i is allocated to exactly one activity, namely $i \in p$. The activities are decomposed on sub-activities, with a source node and a finish node, both added to observed emission, establishing the beginning and the finish of the activity, following a workflow model.

To establish an execution model of sub-activities, an inductive approach to construct workflows is proposed, following the four phases below:

1. *Detailed activities execution phase* – in this phase the sub-activities are performed using a provided generic model to guide the activities' development. In situations not described by the general model, the actors execute the sub-activities needed and make registrations of the sequences generated, that are the learning of each one about the process.

2. *Workflow induction phase* – in this phase a machine learning compound inducts the workflow, with the symbols of tasks recorded in previous phase, describing consistently the observations. The workflow is, hence, a description of how actually the job is performed, and not how the job should be done, according to the specific goals of that process.
3. *Analysis and improvement of the workflow* – as the common practice is not necessarily the better practice, the model must be analyzed and improved. This is performed by experts on the process, who can verify the conjoint probabilities of each sequence generated into the same workflow, what can be named as a *modified Viterbi algorithm*, introduced by Stolcke [38].
4. *Determination of the project activity duration and best path* – after finding the most likely sequence provided by the workflow inducted by the Stolcke's algorithm, or merge algorithm for an HMM, now the evaluation of the shorter duration, or optimistic duration of activity $Dur_{\text{opt}}(V^*|M)$, the most likely duration of aggregate activity, $Dur(V|M)$, as well as the longer duration or pessimistic $Dur_{\max}(V|M)$, and the probability of rework inherent to this activity, based on the hidden semi-Markov models, using the durations of the states.

4.2. The merge algorithm for a single activity HMM

In this section, an algorithm has been developed to merge state symbols, based on graph theory, where the incidence matrix β_{inc} gives the path to joint identical emission symbols that represent the same task. Stolcke [38] have described the algorithm that inducts the structure and the transition probabilities of the emissions in an HMM, for a given observation series of shop floor realizations. The algorithm inputs are the sequential observations arranged into a graph, where the incidence matrix can be read (see Fig. 1), where each $NumObs$ represents the number of observed realizations in shop floor by the process actors, so the graph sequence is divided proportionally $(1/NumObs)$ in the quantity of observations made.

Applying the merge algorithm on the incidence matrix, the HMM that better describes the hidden workflow is given by the algorithm resulting on the graph shown in Figure 2.

The merge algorithm works as follows:

1. First, for a pair of emissions i, j, let an arc being connecting it, *i.e.* $i \to j$, it will be marked with $a+1$ for i, and from i to j will be marked with $a-1$ in j;
2. For each symbol emitted, the $+1$ are summed and stored on a variable called $NumPos$, and for the same symbol the -1 are summed and stored on a variable called $NumNeg$;
3. After these phases, the new probability of transition, $NewProbTrans$, between two emissions, can be obtained dividing the original probability, *i.e.* the initial probability 1.0, by the number of positive arcs leaving each symbol;
4. With these results, the transition matrix is updated through a multiplication of the $NewProbTrans$ by the $NumNeg$;

Algorithm 1 Merge

Require: Incidence matrix β_{inc}

For $i = 1$ to m

  For $j = 1$ to n

    NumPos = $\Sigma_i \Sigma_j b_{i,k}$; for all $b_{i,k} = 1$

    NewProbTrans = 1/NumPos;

    NumNeg = $\Sigma_i \Sigma_j b_{j,k}$; for all $b_{j,k} = -1$;

  End For

End For

For $i = 1$ to m

  For $j = 1$ to n

  $a_{i,j}$ = NewProbTrans × NumNeg;

  End For

End For

FIGURE 1. The description of the Merge Algorithm.

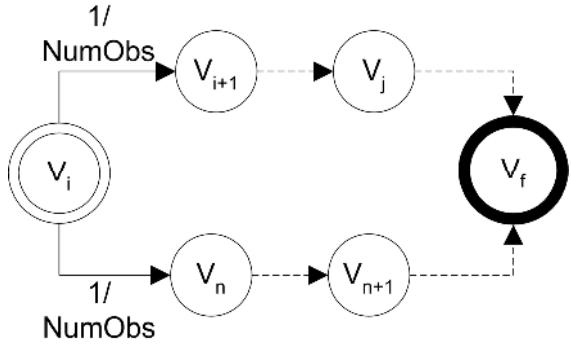

FIGURE 2. Graph describing the sequence of observed symbols.

5. The updated transition matrix explains how the emissions are observed throughout time, and the emission's probability for each state, that is hidden, not directly observed.

After the transitions matrix has been obtained from a series of observations, we can evaluate the activity progress, the rework probability fraction, and the probability of a given sequence, as the expression (4.1):

$$Pr(rework)_p = \prod_i \prod_j a_{ij} \quad \forall i \leqslant j. \tag{4.1}$$

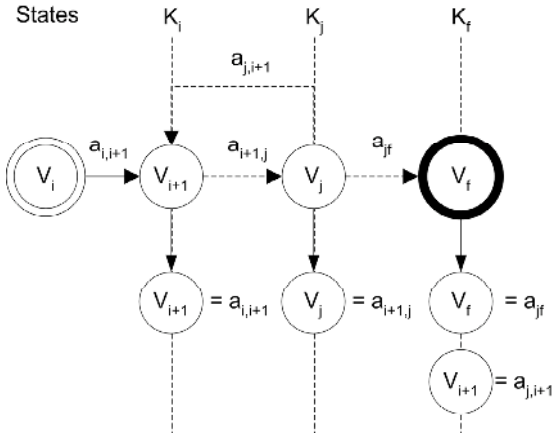

FIGURE 3. Graph describing the workflow of an activity.

Expression (4.1) performs the evaluation of lower-triangular section of the transitions matrix, and returns the conjoint probability of each state emits the same output symbol task more than once.

In order to calculate the probability of a given sequence of states, we must follow the expression:

$$P(v_i, v_{i+1} \dots v_k \dots v_j, v_f|M) = p(v_i \to v_{i+1}) \cdot p(v_{i+1} \cdots \to v_k) \times p(v_k \cdots \to v_j) \cdot p(v_j \to v_f). \tag{4.2}$$

4.3. The parameter determination of the hidden semi-Markov model for a single activity

The model calculates the optimistic duration of an activity $Dur_{\text{opt}}(V^*|M)$, the most likely duration of aggregate activity, $Dur(V|M)$, represented by the weighted average of probabilities with the inner duration of each emission. Furthermore, can be calculated the longest duration, or pessimistic, $Dur_{\max}(V|M)$, that is the duration of activity and the impact of the probable rework on the sequence, obeying the expressions (4.3)–(4.5).

$$Dur_{\text{opt}}(V^*|M) = \sum_i \sum_j a_{ij} \cdot \delta_i \quad \forall i, j \in V^* \tag{4.3}$$

$$Dur(V|M) = \frac{\sum_i \sum_j a_{ij} \cdot \delta_i}{\sum_i \sum_j a_{ij}} \quad \forall j = i + 1 \tag{4.4}$$

$$Dur_{\max}(V|M) = Dur(V|M) + \frac{\sum_i \sum_j a_{ij} \cdot \delta_i}{\sum_i \sum_j a_{ij}} \quad \forall j \leqslant i. \tag{4.5}$$

5. Multiple activities model

Given two activities, which have been modeled as a hidden Markov model, with a certain grade of overlapping y_{pq}, we define:

1. **Completion state of an activity** (D_p). It determines with how many states the activity will be concluded. It can be represented also, by the final state number, that has been added to the observations' records.
2. **Evolution state of an activity** τ_p. It denotes the evolution of a given activity related to a certain degree of overlapping, and it is measured in terms of quantity of states. To be more conservative, it must be represented by the lower integer of the fraction.
$$\tau_p = \left\lfloor \left(1 - \frac{y_{pq}}{Dur(V|M_p)}\right) \cdot D_p \right\rfloor \tag{5.1}$$
where y_{pq} denotes the fraction of overlapping of q on p.
3. **Probability of changes** (P_{pq}). It expresses the probability of a given sub-activity contained in p suffers any parameter modification, when the states are being visited $[(k_{i-1} \cdot \delta, k_i \cdot \delta)]$ affecting the succeeding activity q.
Let the activity p be a set of states $\{k_1 \ldots k_p\}$, the probability of change of each pair of states can be derived from the information entropy concept, introduced by Shanon (see [4,21,22,29,35]), where there is an intrinsic relationship between the information entropy H and the probability of change $p_{ij}(k_i)$, so,
$$H(k_i) = -\sum_{a}^{n} p_i(k_i) \log p_i(k_i) \cdot \tag{5.2}$$
The summation only has validity of within the state D_{p-1}, because on last state, D_p, whole information is available, and to a hidden state transition, where time is a continuous function, let
$$H(k_i\, k_j) = \sum_{k=1}^{D_{p-1}} H(k_i) \tag{5.3}$$
and
$$P_{pq}\,(k_i) = \frac{H(k_i)}{\sum\limits_{k=1}^{D_{p-1}} H(k_i)} \cdot \tag{5.4}$$
4. **Amount of impact of changes** $(S_{pq}(k_i))$. It determines the quantity of time units needed to accommodate the changes of parameters with probability $p_{ij}(k_i)$.
$$S_{pq}(k_i) = P_{pq}(k_i) \cdot \sum_{i}^{n} \delta_i \cdot p_{ij}, \quad k_i \in \{1 \ldots D_i - 1\}. \tag{5.5}$$

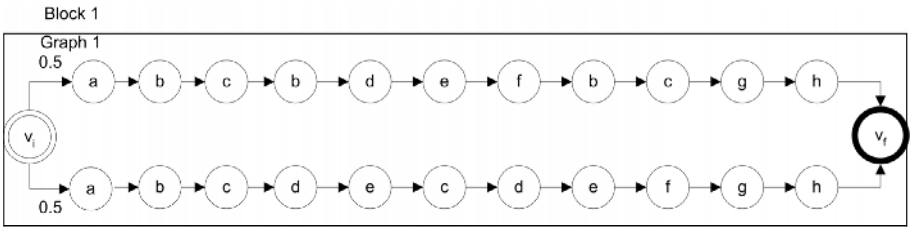

FIGURE 4. A pair of observations sequences of Block 1.

5. **Extended design time** (Edt_q). It is influenced by the information entropy transmitted between activities. It can be determined as the summation of the amount of impact of changes of each state, in time units.

$$Edt_q = \sum_{k_i=\tau_p}^{k_j=D_p} s_{ij}(k_i), \quad \tau_p \leqslant D_p. \tag{5.6}$$

6. **Fraction of rework** (L_q). It determines the fraction of rework needed by the activity q that has overlapped the previous activity p in a certain amount of overlapping y_{pq}. It is a fraction of the extended design time and the duration of most likely duration $Dur(V|M_q)$ of activity q.

$$L_q = \frac{Edt_q}{Dur(V|M_q)}. \tag{5.7}$$

For some authors, the fraction of rework is obtained through experts interviews or in an empiric way [39–41].

6. RESULTS

6.1. OBTAINING A WORKFLOW MODEL FROM A SET OF OBSERVATIONS OF A SINGLE ACTIVITY

Let an activity called Block 1, where the observations graph has two branches of recorded realizations, called sub-activities, as viewed in Figure 4. After the input graph of the block, the merge algorithm defines the new workflow model, through the incidence matrix of Graph 1, β_{inc_1} (Fig. 5).

To apply the merge algorithm to find the HMM that better describes the sequences of sub-activities, the input of the model, *i.e.* the new transition matrix, can be viewed on Table 2.

For the first macro activity, the calculus is demonstrated in the Appendix 1. Obtaining the transition probability matrix shown at Table 2.

		e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	e14	e15	e16	e17	e18	e19	e20	e21	e22	e23	e24
	vi	1												1											
	a	-1	1											-1	1										
	b		-1	1	-1	1			-1	1					-1	1									
	c			-1	1					-1	1					-1	1		-1	1					
Binc_1	d					-1	1										-1	1		-1	1				
	e						-1	1										-1	1		-1	1			
	f							-1	1													-1	1		
	g										-1	1											-1	1	
	h											-1	1											-1	1
	vf												-1												-1

FIGURE 5. A spreadsheet format of the incidence matrix β_{inc_1}.

TABLE 2. The probability transitions for Block 1.

		v_i	a	b	c	d	e	f	g	h	v_f
	v_i		1.00								
	a			1.00							
	b				0.75	0.25					
	c			0.25		0.50			0.25		
$A_1 =$	d						1.00				
	e				0.33			0.67			
	f			0.50					0.50		
	g									1.00	
	h										1.00
	v_f										

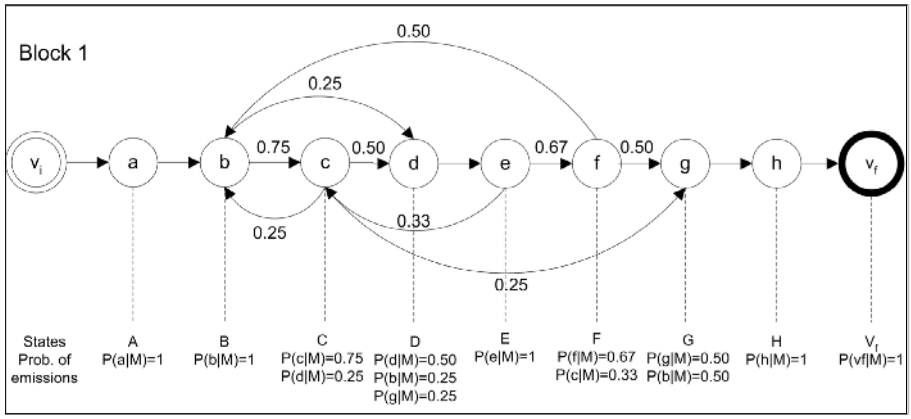

FIGURE 6. The HMM that represents the workflow for Block 1.

The HMM that describes the workflow model for Block 1 is shown on Figure 6. It shows the hidden states ($K_1 = \{V_i, A, B, C, D, E, F, G, H, V_f\}$) and the symbols emitted with its probability of occurrence.

After the workflow model has been found, the rework probability and the time parameters of activity Block 1 and Block 2 must be calculated. The rework

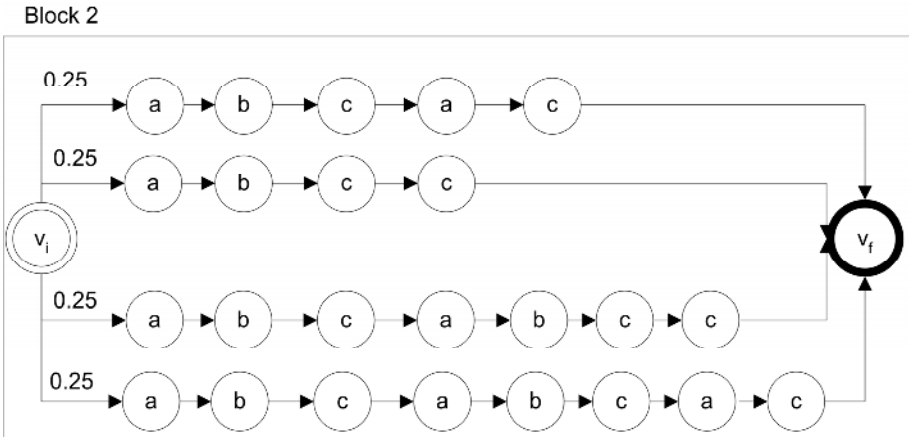

FIGURE 7. A series of observations realized in shop floor for Block 2.

Binc_2	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26	c27	c28
vi	1						1					1								-1								
a	-1	1		-1	1		-1	1				-1	1		-1	1				-1	1		-1	1		-1	1	
b		-1	1					-1	1				-1	1		-1	1				-1	1		-1	1			
c										-1									-1									
c			-1	1	-1	1			-1	1	1			-1	1		-1	1	1			-1	1		-1	1	-1	1
vf						-1					-1								-1									-1

FIGURE 8. A spreadsheet format of the incidence matrix β_{inc2}.

probability is calculated with the lower-triangular matrix (see Tab. 2), from the transitions probability matrix A_1.

Given the duration parameters for each activity, $\Delta = (2.08, 1.97, 1.98, 1.90, 2.03, 1.96, 1.86, 2.02, 2.13)$ in time units obtained by Monte Carlo method, the most likely duration of the activity Block 1. The maximum duration can be obtained with the sum of the most probable duration and the multiplication of the lower-triangular matrix.

For a second activity called Block 2 (see Fig. 7), let be four series of observations grouped into graph G_2, the same calculations must be done in order to evaluate the rework probability and the duration parameters.

With these sequences of observations, can be extracted the incidence matrix, β_{inc2}, shown in Figure 8.

Applying the merge algorithm the transition probability matrix can be found, and describes the workflow model to Block 2 realizations, as shown on Table 3.

The HMM that describes the workflow model for Block 2 is shown on Figure 9. It shows the hidden states ($K_2 = \{V_i, A, B, C, V_f\}$) and the symbols emitted with its probability of occurrence.

After finding the workflow model, the rework probability and the time parameters of this activity Block 2 should be determined. The rework probability is

TABLE 3. The probability transitions for Block 2.

		v_i	a	b	c	v_f
	v_i		1.00			
	a			0.75	0.25	
$A_2 =$	b				1.00	
	c		0.40		0.20	0.40
	v_f					

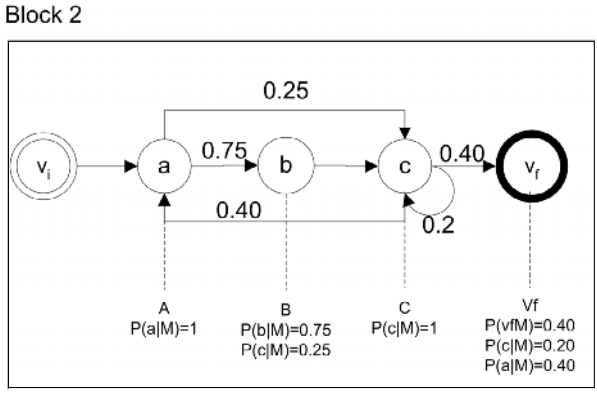

FIGURE 9. The HMM that represents the workflow for Block 2.

TABLE 4. The model results for Block 1, 2.

Activity Parameter	Block 1	Block 2
$Pr(rework)_i$	0.04	0.08
$P(best_path\|M_i)$	0.13	0.30
$Dur(V\|M_i)$	16.05 t.u.	15.69 t.u.
$Dur_{\max}(V\|M_i)$	18.03 t.u.	19.59 t.u.
$Dur_{\text{opt}}(V\|M_i)$	14.90 t.u.	15.64 t.u.

calculated with the lower-triangular matrix, from the transition probability matrix A_2 (Tab. 3).

The results are shown on Table 4.

6.2. ANALYSIS OF THE OVERLAPPING EFFECTS FOR MULTIPLE ACTIVITIES

Let a certain grade of overlapping of Block 2 on Block 1, $y_{12} = 5$ time units (t.u.), the impact of changes on Block 2 should be calculated, as the Block 1 has not been finished yet, and the information to flow to Block 2 is incomplete. Table 5 shows the results of the multiple activity model.

TABLE 5. The multiple activity model results for an overlapping grade of $y_{12} = 5$ t.u.

Activity parameter	Value	Unit
Overlapping length	6–9	states
τ_1	6	states
D_1	9	states
$H(6)$	0.2754	
$H(7)$	0.3010	
$H(8) = H(9)$	0	
$H(6,9)$	0.5764	
$P(6)$	0.4778	
$P(7)$	0.5222	
$P(8) = P(9)$	0	
δ_6	1.9666	t.u.
δ_7	1.9150	t.u.
$s(6)$	0.9396	t.u.
$s(7)$	1.0000	t.u.
Edt_2	1.9396	t.u.
$Dur(V\|M_2)$	15.698	t.u.
L_2	0.1236	

7. SENSITIVITY ANALYSIS

This session investigates the interdependencies between the main variables involved in overlapping activities and evaluate the theory developed. In Figure 10 it can be seen that as the degree of overlap between two activities increases, decreases the amount of activity supplemented by the predecessor states.

Related to the degree of overlap and the amount of entropy, which can be observed that increases as it grows, because the greater the amount of states not yet completed increase the uncertainty in the overlap and hence the greater the entropy. It should be noted that in Figure 11 in some states where only one symbol is given no increased entropy, because it is zero.

The entropy decreases due to the state of completion of an activity as the higher the available information to be passed on to the successor activity, the greater the likelihood of success of the process, or less need for rework.

It is concluded that the degree of overlap decreases if the state of completion increases, as the entropy value influences, through which the quantity of emitted symbols in each state affects the need to rework to accommodate changes due to the uncertainty of the process. As the degree of overlap decreases and increasing the state of completion of the activity, decreases the need for rework, as shown in Figure 12.

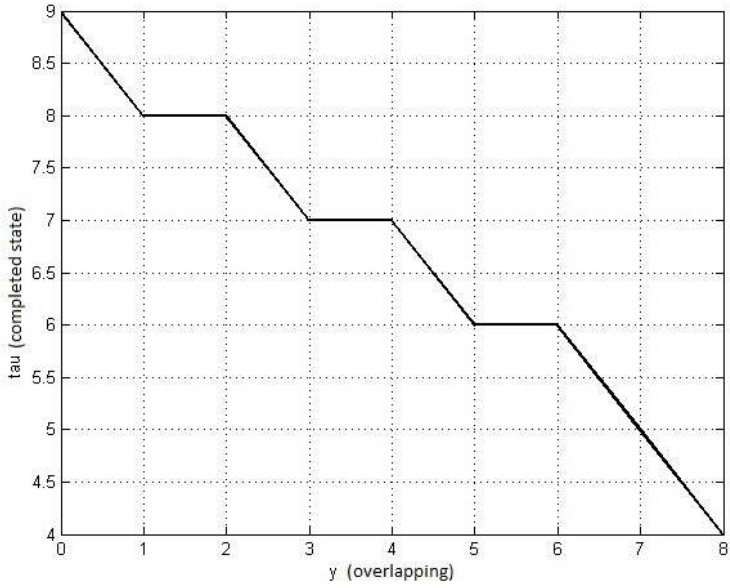

FIGURE 10. Relationship between the overlapping amount (y_{pq}) and the completed state (τ_{p}).

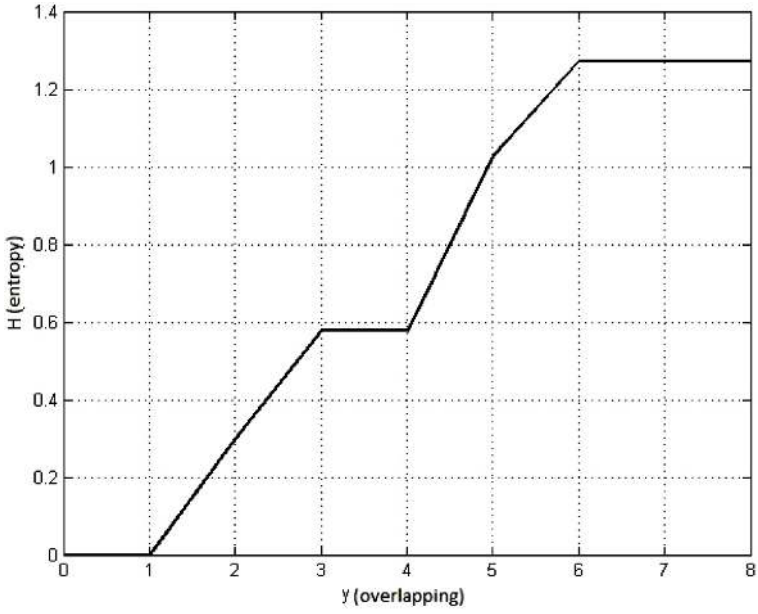

FIGURE 11. Relation between overlapping and the entropy.

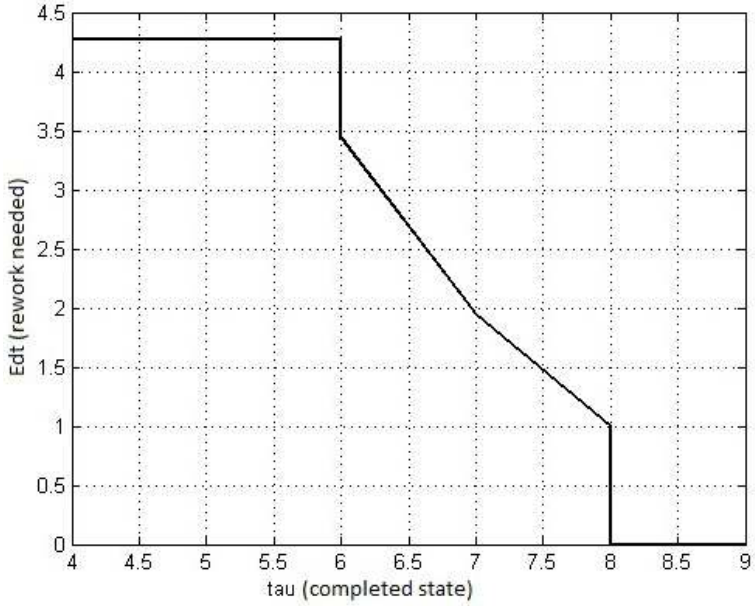

FIGURE 12. The decreasing of rework necessity as the the evolution of the completed states.

8. Conclusions and suggestions

In the present work has been presented that the project management area, more specifically the shortening techniques of project schedule are already an increasing field of research, based on the last researches [41]. Furthermore, could be observed that the project activities had been considered on a macro level, instead of its micro level, where the tasks have been realized, characterized by state sequence emissions, described as a hidden semi-Markov model.

A disaggregation model was developed to project-oriented production systems, using the observations done in the sub-activities developments by the actors of the process. The model has been obtained from a framework of steps to acquire and analyze data of the observations and derived through operations on the incidence matrix of a graph that describes each set of observations that produce a same output.

The aggregate activities of a project are formed by sub-activities, that give the information needed to establish a way to determine the optimistic, most likely and pessimistic duration of an activity. Furthermore, provides a model to establish the most likely sequence of operations, or tasks, that should be performed in order to accomplish the optimistic duration, and also the probability of inherent rework.

To support the development of the stochastic model that aims to evaluate the probability of changes and its impact on the duration of activity, two aspects have been analyzed:

- The first one is to determine the state, or sub-activity, where the activity overlapped is when the succeed activity starts with a certain degree of overlapping, so the evaluation of the probability of changes is done.
- The second evaluates the impact of change in each state and how to quantify this impact onto macro activity, obtaining the extended design time and the fraction of rework.

The model was demonstrated through a numerical example with two activities, analyzing its parameters of duration, its workflow models and the interaction between them.

References

[1] R. Agrawal and D. Gunopulos, *Mining Process Models from Workflow Logs*. IBM Research Center (1998).
[2] R. Ahlswede, N. Cai, S.R. Li and R.W. Yeung, Network Information Flow. *IEEE Trans. Inf. Theor.* **46** (2000) 1204–1216.
[3] R. Ahmadi, T. Roemer and R.H. Wang, Structuring product development model. *Eur. J. Oper. Res.* **130** (2001) 539–558.
[4] Y. Bard, Estimation of state probabilities using the maximum entropy principle. *IBM J. Res. Dev.* **24** (1980) 563–569.
[5] P.O. Boaventura, *Grafos: teoria, modelos, algoritmos*. E. Blucher (1996).
[6] T.R. Browning, E. Fricke and H. Negele, Key concepts in modeling product development processes. *Syst. Eng.* **9** (2006) 104–128.
[7] H.H. Bui, D.Q. Phung and S. Venkatesh, Hierarchical hidden Markov models with general state hierarchy. In *AAAI* (2004).
[8] A.K. Chakravarty, Overlapping design and build cycles in product development. *Eur. J. Oper. Res.* **134** (2001) 392–424.
[9] S.H Cho and S.D. Eppinger, A simulation-based process model for managing complex design projects. *IEEE Trans. Eng. Manage.* **52** (2005) 316–327.
[10] J. Crampton, *On the satisfiability of authorization constraints in workflow systems*. Department of Mathematics, Royal Holloway, University of London (2004).
[11] P. Doshi, R. Goodwin, R. Akkiraju and K. Verma, Dynamic workflow composition using Markov decision processes. *Int. J. Web Serv. Res.* **2** (2005) 1–17.
[12] T.V. Duong, H.H. Bui, D.Q. Phung and S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *IEEE* (2005).
[13] S. Fine, Y. Singer and N. Tishby, The hierarchical hidden Markov model: analysis and applications. *Mach. Learn.* **32** (1998) 41–62.
[14] D.N. Ford and J.D. Sterman, The Liar's Club: concealing rework in concurrent development. *Concurr. Eng.: Res. Appl.* **11** (2003) 211–119.
[15] J.E.V. Gerk and R.Y. Qassim, Project Acceleration *via* Activity Crashing, Overlapping and Substitution. *IEEE Trans. Eng. Manage.* **55** (2008) 509–601.
[16] S.T. Hackman and R.C. Leachman, An aggregate model of project-oriented production. *IEEE Trans. Syst. Man Cybern.* **19** (1989) 220–231.
[17] K.V. Hee *et al.*, Scheduling-free resource management. *Data Knowl. Eng.* **61** (2007) 59–75.
[18] J. Herbst, *An inductive approach to the acquisition and adaptation of workflow models*. Daimler Chrysler AG Research and Technology (1999).

[19] J. Herbst, A machine learning approach to workflow management. *Lect. Notes Comput. Sci.* **1810** (2000) 183–194.
[20] J. Herbst and D. Karagiannis, Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. In *IEEE* (1998).
[21] E.T. Jaynes, Information theory and statistical mechanics. *Phys. Rev.* **106** (1957) 620–630.
[22] D. Kimber, *Notes on Statistical Mechanics, Information Theory and Thermodynamics*. Xerox Palo Alto Research Centre (1994).
[23] V. Krishnan, *Design process improvement: sequencing and overlapping activities in product development* (1993).
[24] V. Krishnan, S.D. Eppinger and D.E. Whitney, A model-based framework to overlap product development activities. *Manag. Sci.* **43** (1997) 437–451.
[25] W. Li and Y. Fan, Time constraints in workflow models. In *ICAM* (2003).
[26] S. Luhr, S. Venkatesh, G. West and H.H. Bui, *Duration abnormality detection in sequences of human activity*. Dept. of Computing, Curtin University of Technology (2004).
[27] J.U. Maheswari and K. Varghese, Project scheduling using dependency structure matrix. *Int. J. Project Management* **23** (2005) 223–230.
[28] C. Mitchell, M. Harper and L. Jamieson, On the complexity of explicit duration HMM's. *IEEE Trans. Speech Audio Proc.* **3** (1995) 213–217.
[29] S.K. Mitter, *Statistical inference, statistical mechanics and the relationship to information theory*. Lecture Notes, MIT (2004).
[30] K.P. Murphy, *Hidden semi-Markov models (HSMMs)*. University of California at Berkeley (2002).
[31] S. Nicoletti and F. Nicoló, A concurrent engineering decision model: management of the project activities information flows. *Int. J. Prod. Econ.* **54** (1998) 115–127.
[32] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77** (1989) 257–286.
[33] T.A. Roemer, R. Ahmadi and R.H. Wang, Time-cost trade-offs in overlapped product development. *Oper. Res.* **48** (2000) 858–865.
[34] T.L. Saaty and J.M. Alexander, *Thinking with Models: Mathematical Models in the Physical, Biological and Social Sciences*. Pergamon-Press (1981).
[35] C.E. Shannon, A mathematical theory of communications. *The Bell Systems Technical Journal* **27** (1948) 379–423, 623–656.
[36] H. Shatkay and L.P. Kaelbling, Learning geometrically-constrained hidden Markov models for robot navigation: bridging the topological-geometrical gap. *J. Artificial Intelligence Res.* **16** (2002) 167–207.
[37] J.H. Son and M.H. Kim, Improving the performance of time-constrained workflow processing. *J. Syst. Softw.* **58** (2001) 211–119.
[38] A. Stolcke, *Bayesian Learning of Probabilistic Language Models* (1994).
[39] A.A. Yassine, R.S. Sreenivas and J. Zhu, Managing the exchange of information in product development. *Eur. J. Oper. Res.* **184** (2008) 311–326.
[40] A.A. Yassine, D.E. Whitney and T. Zambito, Assessment of rework probabilities for simulating product development processes using the design structure matrix. In *ASME* (2001).
[41] H. Zhang, W. Qiu and H. Zhang, An approach to measuring coupled tasks strength and sequencing of coupled tasks in new product development. *Concurr. Eng.: Res. Appl.* **14** (2006) 305–311.
[42] H. Zhao and P. Doshi, Composing nested web processes using hierarchical semi-Markov decisioxn processes. In *AAAI* (2006).