A dynamic critical path computation algorithm for enterprise process cooperative scheduling

Yang, F.; Tan, W.; Shen, W.; Ghenniwa, H.; Xue, Y.

https://doi.org/10.1109/CSCWD.2010.5471902

A dynamic critical path computation algorithm for enterprise process cooperative scheduling

NRCC-53264

Yang, F.; Tan, W.; Shen, W.; Ghenniwa, H.; Xue, Y.

May 2010

A version of this document is published in / Une version de ce document se trouve dans:
2010 14th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2010), Shanghai, P.R. China, April 14-16, 2010, pp. 606-610

A Dynamic Critical Path Computation Algorithm for Enterprise Process Cooperative Scheduling

Fujun Yang[1], Wenan Tan[2], Weiming Shen[1,3], Hamada H.Ghenniwa[1] , Yunjiao Xue[1]
[1]Department of Electrical and Computer Engineering, University of Western Ontario, London, Ontario, Canada
[2]School of Computer and Information, Shanghai Second Polytechnic University, Shanghai, P.R. China
[3]Centre for Computer-assisted Construction Technologies, National Research Council, London, Ontario, Canada
fyang73@ uwo.ca, twajsj@sohu.com, weiming.shen@nrc.gc.ca, hghenniwa@eng.uwo.ca

Abstract—Business process simulation plays an important role for enterprise business process re-engineering. This paper proposes a new algorithm to dynamically select a critical path with multiple constraints for ***enterprise process cooperative scheduling within*** *cooperative business simulation systems. The proposed algorithm is based on Dynamic Programming and Analytic Hierarchy Process (AHP), and has been validated through a prototype implementation with case studies.*

Keywords- Critical Path; Dynamic Critical Path Selection; Analytic Hierarchy Process; Business Process Simulation; Dynamic Programming

I. INTRODUCTION

As a result of advanced network technologies and the rapid development of economic globalization enterprises are facing tremendous pressure when it comes to competing on the world stage. An adaptive information management system is urgently required to enhance productivity, shorten the product development cycle, improve product quality, and that will allow for businesses to make rapid adjustments to deal with a competitive market.

Business process simulation based on dynamic modeling and intelligence diagnosis is the key technology for the coordination of various business processes [1, 2]. By setting up the process model and enacting it in the simulation server, a simulation system can help streamline the business process, deliver tasks and documents among users, and monitor the overall performance of the process [3].

There are various scheduling algorithms in the simulation engine domain, for example, Tailoring A Measurement Environment [4], Shark Engine [5], JBpm engine implementation mechanisms [6], YAWL engine scheduling mechanism [7], and Bossa Engine (based PN machine) [8]. In the simulation engine domain, the internal scheduling and implementation mechanisms have their own characteristics. Although the algorithms look similar, their focuses are different. For the development of workflow, finding a suitable and stable mechanism for scheduling and promoting enterprise cooperative work is the key to ensuring the quality of the simulation engine and ultimately the performance of business process simulations.

A favorable critical path is very important for a cooperative process dynamic simulation algorithm, which should be suitable to tackle some fuzzy constraints. Selection of the critical path is a hot topic of a simulation algorithm for a complex simulation system. Meanwhile, the critical path is the core of intelligent analysis and diagnosis. The process path algorithm decides the success or failure of the simulation system development.

In addition to the critical path, there are some paths whose conditions are very close to the critical path, known as the hypo-critical-paths, which are usually watched by the commanding managers at all levels. In process scheduling, particular attention should be paid to the dynamic critical path, because the main-critical-paths and the hypo-critical-paths usually could be transferred during the enterprise process simulation or process enactment [9]. The managers should pay attention to the management of the activities of the hypo-critical-paths. Once the working environment and the original plan change, the hypo-critical-path maybe become the main-critical-path.

This paper proposes an algorithm to dynamically select a critical path with multi-constraints for a cooperative simulation system. The algorithm proposed is based on Dynamic Programming and AHP.

The rest of this paper is organized as follows. Section 2 analyzes previous work in related research; Section 3 presents the architecture of process simulation system; Section 4 describes the computing process of the algorithm based on a use case in an application system. Finally Section 5 concludes the paper and proposes our future work.

II. RELATED WORK

A critical path with multi-constraints is commonly used in real life. Research on selecting a critical path with multi-constraints has a long history in many fields. It has been well studied in several diverse areas such as monitoring traffic flows, workflow management, decision-making, and AON (Activity on Node) [10]. In these research fields, a critical path measures the similarity degree of two activity networks. However, these researches are mainly focused on finding matches based on the pure short path (such as shortest flux, shortest time) or the geometry perspective without considering the conceptual semantics of the whole net in a knowledge context in real applications.

Suh et al. [11] studied the measurement of the cost constraints of the network measurement model for optimization problems. Without taking into account sampling frequency, circumstances, and the different optimization objectives of the formation of the different measurement models, these optimization problems are NP hard. Shmoys et al. [12] gave an algorithm which can obtain the approximation result with an approximate ratio of $1+\frac{1}{e}$. Khuller et al. [13] proposed a greedy strategy to the approximation with an approximate ratio of $1-\frac{1}{e}$. However, both algorithms cannot solve the fuzzy weights of some parameters. Chang et al. [14] proposed a critical path algorithm called ICSF, based on the controlling flow structure, according to the model time-constrained workflow model to determine the critical path. H. Pan [15] suggested the AON diagram of the topology gateway key path TCNA algorithm to determine the critical path. However, such a critical path calculating algorithm is not perfectly applicable in a dynamic process scheduling system with fuzzy conditions. Furthermore, those works did not handle cases of weight-changing schedules, which are common in highly competitive business environments.

To the best of our knowledge, no research has been done to measure the dynamic critical path based on a single algorithm for fuzzy conditions.

III. A Dynamic Critical Path Selection Algorithm

In a business process simulation system, the mechanism of activity scheduling is the key to enterprise business processes cooperative work. In practice, we found that the critical path may change following changing markets in a complex dynamic environment, which we call a dynamic critical path. This section presents an algorithm to find a dynamic critical path under multi-constraints.

The AHP (Analytic Hierarchy Process) was proposed by Saaty [16] of the United States in 1980, and was used to solve complicated multi-objective decision-making problems. An important feature of the AHP is its ability to integrate expertise to solve fuzzy questions.

A. Definitions

In this subsection some definitions to be used in this paper are given.

Definition 1. Set P is Processes set, $P=\{p_i\},(i=0,\cdot\cdot,n)$, in which, P_0 means business process beginning; n is the sum of the business processes.

Definition 2. Set Path is key path set of enterprise business processes. $Path_k=\{A_{k1},A_{k2},..A_{ki},i=1,2,\cdot\cdot n\}$ is the k-th key path of enterprise business processes where activity A_{ki} is included in the $Path_k$, n is the number of activities in the $Path_k$, and k is the number of key paths to be selected, which is a waiting decision-making variable.

Definition 3. $ActIFCost(A_{ki},IF_j,t_1,t_2)$ is the activity cost that Activity A_{ki} uses infrastructure IF_j in $[t_1,t_2]$.

Definition 4. $IFCap_j$ presents the capability of infrastructure IF_j, where $IFCap_j \geq \max\{ActIFCost(A_{ki},IF_j,t_1,t_2)\}$.

Definition 5. $ResCost(A_{ki},r_j,t_1,t_2)$ is the consumption that A_{ki} consumes *Material* j in time $[t_1,t_2]$, where j is the serial number, $m_j = unit * r_j$; r_j is the quantity of resource j.

Definition 6. $Time$ is the process duration utility which can be generated from the activity flow. $OTime$ is the deadline of the order form.

Definition 7. $Speed$ is used to describe the capability of enterprise processes in which all activities are characterized with a product heap utility or the time to input products waiting for handling by subsequent activities defined in a Stream-Like process. The detailed definition can be found in [17].

B. Proposed Solution

To resolve a multi-objective optimization problem, we can follow three steps as shown in Figure 1. Firstly, according to Dynamic Programming [18], we divide all the activities into different phases to get some process parts under some single limit. Secondly, using the improved AHP [16] to choose the result of the first step, we can design an algorithm to get a weight set of candidate critical paths. The third step is to dynamically choose the right critical path in a dynamic environment in order to solve the problem according to the different requests by adjusting the weights.

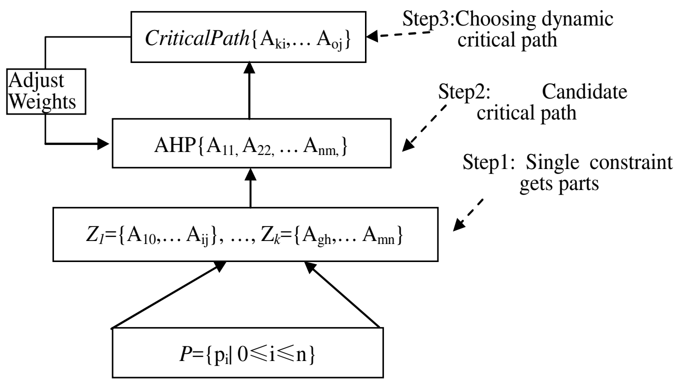

Figure 1. Schematic representation of the proposed algorithm

The proposed algorithm is described below in detail.

- **Step 1. Dynamic programming pretreatment**

An optimum critical path must meet the constraints of infrastructure, materials, and time simultaneously.

However, in a complex process activity network, it is too difficult to find a path that satisfies all the constraints. So we use Dynamic Programming to get every sub-target path which is a path satisfying only one constraint.

The Dynamic Programming goal formula is $f(s_k) = opt\{v_k(s_k, a_k) + f_{k+1}(s_{k+1})\}$, where v_k is an index function which is the measurement to decide from one state to another. s_k is the state k, and $f(s_k)$ is the most effective value of the strategy starting from the state s_k. In arithmetic, there are three decision variables to get three different optimization decisions. Z is a matrix of the candidate paths under the three constraints.

The least infrastructure used paths can be calculated as follows:

$$Z_1 = \min\left\{\sum_{i=0}^{n}\sum_{j=0}^{m} ActIFCost(A_{ki}, IF_j, t_1, t_2)\right\}. \tag{1}$$

The least material consumed paths can be obtained as follows:

$$Z_2 = \min\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{Re}\,sCost(A_{ki}, r_j, t_1, t_2)\right). \tag{2}$$

The time constraint of the business orders can be calculated using following formula:

$$Z_3 = \min\{Time\}. \tag{3}$$

- **Step 2. Dynamic critical path selection algorithm based on AHP**

From Step 1, we can get the sub-goal domains which include Z_1, Z_2, and Z_3. Those sub-goals divide P into different hierarchies. In this step, we dispose of each sub-goal with the following algorithm to determine some candidate critical paths.

The proposed of algorithm model is described as follows:

1) Construct Hierarchical Model: The top-level layer of the model is the goal layer; the middle is the criteria layer (according to the complexity of the problem, each rule can be divided into several sub-rules); the bottom layer is the candidate-path layer. The model is shown in Figure 2.

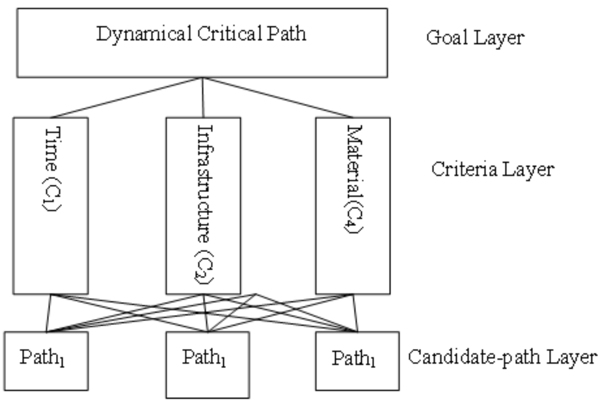

Figure 2. AHP model of the selection algrithm

2) Compute the weights of the rules: The objective is to compare the c_i （i=1, … , mof the $path_j$ with c_i of the $path_k$ the importance being to get a comparison matrix $(Path_{ij})n \times n$. This last step is very important. According to the scheduler situation, we give a proposal to define the weights.

3) *Compute the characteristics vec*tor $(w1,\ldots,w_n)^T$ of the *determination matrix:* Firstly, we compute the $\overline{A_i} = \left(\prod_{j=1}^{n} path_{ij}\right)$, which is normalized to get $w_i = \overline{\overline{w_i}} / \sum_{i=1}^{n} \overline{\overline{w_i}}$, $W = (w_1, \ldots, w_n)^T$.

4) Compute the maximum eigenvalue: It is used to judge the consistency of the determination matrix. When the determination matrix is consistent, $path_{ij} = w_i / w_j$. Thus, the determination matrix can be presented as follows.

$$Path = (path_{ij}) n \times n = \begin{pmatrix} w_1/w_2 & \ldots & w_1/w_n \\ \vdots & \ddots & \vdots \\ w_n/w_1 & \cdots & w_n/w_n \end{pmatrix}. \tag{4}$$

$$PathW = Path \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = n \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = nW \tag{5}$$

where n is eigenvalue.

When the determination matrix is completely consistent, $\lambda_{\max} = n$. Otherwise, $\lambda_{\max} > n$.

- **Step 3. Critical path selection**

In Step 2, we should get $Path$ which includes the candidate critical path of the processes. In this step, we should optimize the paths according to the six process flows to evaluate the simulation and to evaluate the business processes we defined [17]. In this system, the optimum is dynamic following the change of environment, which is shown as follows:

$$CriticalPath = Optimize(Path \mid Time, Service, Quality, Speed, Efficiency, Cost) \tag{6}$$

where, $CriticalPaht()$ is the final goal; $Optimize(A \mid B)$ is to choose a optimum result A under the situation B; The definitions of *Time, Service, Quality, Speed, Efficiency* and *Cost* can be found in [17].

Then we can choose one as the critical path. In the simulation system, the arithmetic supports the dynamic critical path of the complex processes. In a real business environment, it is very important that the process of reasonable planning, and diagnosis of the selection process supports multiple goals.

The number of feasible paths for the entire workflow can be calculated as fellows:

$$K = \max\left(\begin{array}{l}\left[IFCap / \sum_{j}^{m}\sum_{i}^{n} ActIFCost(A_{ki}, IF_j, t_1, t_2)\right], \\ \left[\mathrm{Re}\,sCost / \sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{Re}\,sCost(A_{ki}, r_j, t_1, t_2)\right]\end{array}\right) \tag{7}$$

According to the real situation changing and new requirements, the weights change back to Step 2 as a feedback, and is re-calculated.

IV. PROTOTYPE IMPLEMENTATION AND CASE STUDY

In this case study, we use the process simulation and flow-analysis technology to get different kinds of data, such as activities the AON network, resource efficiency, process cost, and cost flow information scheduled by three-level combined scheduling strategies. Before the process simulation, we need to define the coordination rules. In the AON network, the nod A_i is an activity i including the beginning and the end of the activity, as shown in Figure 3. The activity diagram is described in [17]. A network of tasks is set up to show the dependent sequence of activities within a project. The critical path method can be applied to a network to answer the most common question asked by project managers. In Figure 3, (A_{12}, A_{23})(3, 1, 5) indicates that from Activity A_{12} to Activity A_{23}, there are three constraints, the values of the three constraints, are respectively 3, 1, and 5.

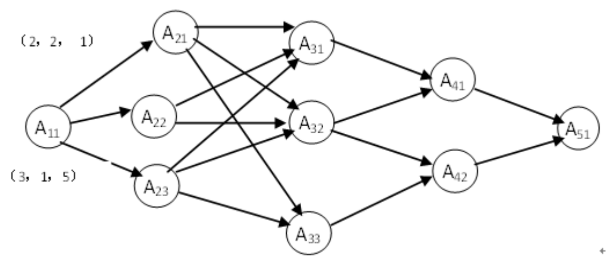

Figure 3. A use case of activities of a process simulation system

From Step 1, we can get the three candidate paths: Z_1=(A_{11}, A_{23}, A_{32}, A_{42} ,A_{51}), Z_2=(A_{11}, A_{23} ,A_{33} ,A_{42} ,A_{51}), Z_2=(A_{11}, A_{23} ,A_{32} ,A_{42} ,A_{51}). Based on Saaty's Nine Class Degree Table [19], we define the criteria weights.

Taking the weight of time as an example, a weight w_j of the corresponding elements of Z_1 and Z_2 on the time is computed as: $t = \dfrac{1}{t_{1j} - t_{2j}}$, where t_{1j} and t_{2j} are the time of the corresponding elements of Z_1 and Z_2. Then weight w_j is the result of the value of the mapping t into Saaty's Nine Class Degree Table [19]. In this case, we compare the time criteria (C_1) of Z_1, Z_2 and Z_3, relative to the A_{23}, which is shown as in Table Ⅰ.

TABLE I. WEIGHTS OF THE C_1

C_1	Z_1	Z_2	Z_3
Z_1	1	1/3	1/2
Z_2	3	1	1/4
Z_3	2	4	1
Total	6	16/3	7/4

The Eigenvector of the determination matrix is computed by the approximation algorithm [19]. The result is showed in Table Ⅱ.

TABLE II. EIGENVECTOR OF THE DETERMINATION MATRIX ON C_1

C_1	Z_1	Z_2	Z_3	w_i
Z_1	1/6	1/16	2/7	0.172
Z_2	3/6	3/16	1/7	0.277
Z_3	2/6	12/16	4/7	0.552

w_1= (0.172, 0.277, 0.552), then

$$AW = \begin{pmatrix} 1 & 1/3 & 1/2 \\ 3 & 1 & 1/4 \\ 2 & 4 & 1 \end{pmatrix}\begin{pmatrix} 0.172 \\ 0.277 \\ 0.552 \end{pmatrix} = \begin{pmatrix} 0.540 \\ 0.931 \\ 2.004 \end{pmatrix} \tag{8}$$,

$$\lambda_{max} = \frac{1}{3}\left[\frac{0.540}{0.172} + \frac{0.931}{0.277} + \frac{2.004}{0.552}\right] = 3.504 \tag{9}$$

Through the testing using the rule of the consistency method [20], the weight vector w_1 can be used to compare to the three candidate paths, relative to the time constraint.

Using the same method, we can get the weighting vector of three candidate paths, being relative to the other two factors, as shown in Table Ⅲ.

TABLE III. EIGENVECTOR OF THE DETERMINATION MATRIX ON C_1, C_2 AND C_3

	C_1	C_2	C_3
Z_1	0.172	0.055	0.263
Z_2	0.277	0.198	0.547
Z_3	0.552	0.731	0.065

Then, we can calculate the weight vector of the three-criteria set in relation to the final goal, and as follows:

$$W = \begin{bmatrix} 0.444 \\ 0.111 \\ 0.222 \end{bmatrix} \tag{10}$$

Lastly, sorting the candidates is used to select an activity as a point of the critical path.

$$\begin{bmatrix} A_{ij} \\ A_{kj} \\ A_{nj} \end{bmatrix} = \begin{pmatrix} 0.172 & 0.055 & 0.263 \\ 0.277 & 0.198 & 0.547 \\ 0.552 & 0.731 & 0.065 \end{pmatrix}\begin{bmatrix} 0.444 \\ 0.111 \\ 0.222 \end{bmatrix} = \begin{bmatrix} 0.141 \\ 0.266 \\ 0.340 \end{bmatrix} \tag{11}$$

According to the sorting results, A_{nj} is the optimal choice in terms of the overall objective. Using the same procedures, we can get the critical path. In the above case, the optimum critical path is shown as follows.

$$CriticalPath = (A_{11}, A_{22}, A_{31}, A_{42}, A_{51})$$

If the simulation environment changes at the last step, weight W is conveniently changed to satisfy the requirements of the actual application. For example, if the importance of the time criteria declines in some application environment, the only thing that needs to be done is to adjust the value of the relevant time of ithe overall-goal weight matrix. Thereby, the corresponding W value of time is changed. This will satisfy the system requirements in the new environment. There is no need to change the other structures of the algorithm. The results show the proposed algorithm has good dynamic adaptability.

V. CONCLUSION AND FUTURE WORK

This paper proposes a new approach to select a critical path in a business process simulation towards process performance management, which supports enterprise business reengineering and enterprise process flow analysis and prediction [21]. The main contributions of this paper include a new algorithm for choosing a critical path and a sub-algorithm of fuzzy criteria weight designed to compare the same criteria in different paths.

The proposed algorithm guarantees a satisfactory scheduling result of a coordination system that can be suited to the conditions of the process of multi-constrained dynamic simulation.

Our future work is to address the effectiveness of the close loop activities of the process and the different weights of the algorithm used in different domains. As the algorithm does not take into account the division of the granularity of the activities, we will extend the implementation effectiveness of the granularity for accuracy in a large system, based on our algorithm.

ACKNOWLEDGMENT

This paper was supported in part by the National Natural Science Foundation of China under Grant No. 60874120, the Aeronautical Science Foundation of China under Grant No. 20081952014, and the Science Foundation of Baiyulan technological academic under Grant No.2009B049.

REFERENCES

[1] C. Moller, “Process innovation laboratory: a new approach to business process innovation based on enterprise information systems”, Enterprise Information Systems, 1(1) 2007, pp.113-128.

[2] L. Wang, L. Zhang, and B. S. Zhou, “Enterprise Process Modeling System Research and Implementation”, Computer Engineering and Application, 23(3) (2001), pp.74-87.

[3] R. Bate, D. Kuhn, C. Wells, et al. “A Systems Engineering Capability Maturity Model”, Version1.1, CMU/SEI-95-MM-003, Carnegie Mellon University, Pittsburgh, PA, 1995, pp. 58-60.

[4] V. R. Basili, and H. D Rombach. “The TAME Project: Toward Improvement-Oriented Software Environment“, IEEE Trans. Soft. Eng., Xi’an, China, 14 (6) (1988), pp.58-73.

[5] K. Pulford, A. Kuntzmann-Cambelles, and S. Shrilow, “A Quantitative Approach to Software Management”. The AMI Handbook, Addison-Wesley: Wokingham, 1996, pp. 211-215.

[6] [EB/OL] .http://sharkengine.com/.

[7] Process Modeling [EB/OL]. http://doc.javanb.com/jboss-jbmp-3-0-4-userguide/processmodelling.html.

[8] WMP van der Aalst, AHM ter Hofstede YAWL. Yet, “An Other Workflow Language” (Revised version), [EB/OL]. http://www.yawlfoundation.org/documents/yawlrevtech.pdf.

[9] W. Tan, C. Jiang, L. Li, et al., “Role-oriented process-driven enterprise cooperative work using the combined rule”, Inf Syst Front, 10 (2008), 519–529.

[10] H. Wang, et al. “PCAR: Finding Critical Paths Based on Principal Component Analysis”, Computer Engineering and Science, 30(6) (2008), pp. 1-4.

[11] K. Suh, Y. Guo, J. Kurose, et al. “Locating Network Monitors: Complexity, Heuristics, and Coverage”, Proc of IEEE INFOCOM’05, 2005.

[12] F. Chudak, F. Shmoys. “Improved Approximation Algorithms for the Un-capacitated Facility Location Problem“, ACM SIAM Journal on Computing, 33(1) (2003), pp. 1-25.

[13] S. Khuller, A. Moss, and J. Naor. “The Budgeted Maximum Cover-age Problem”, Information Processing Letters, 7(1) (1999), pp. 39-45.

[14] D.H. Chang, J.Y. Son, M.H. Kim. “Critical path identification in the context of a workflow”, Information and Software Technology, 44(7) (2002), pp. 405-417.

[15] J. Pan, et al. “Computing Critical path of Workflows AON”, Computer Engineering, 30(2) (2002), pp. 3-26.

[16] T.L. Saaty. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw-Hill, 1980.

[17] W. Tan, W. Shen, J. Zhao, “A methodology for dynamic enterprise process performance evaluation”, Computers in Industry 58 (2007), pp. 474–485

[18] R.E. Bellman. Numerical Studies of a Two-point Nonlinear Boundary Value Problem Using Dynamic Programming, Invariant Imbedding, and Quasi linearization (2005) http://en.scientificcommons.org/19813206.

[19] T.L. Saaty, L.G. Vargas, Models, Methods, Concepts & Applications of the Analytic Hierarchy Process, Kluwer Academic, 2001.

[20] T.L. Saaty, B. Cillo, The Encyclicon, Volume 2: A Dictionary of Complex Decisions using the Analytic Network Process. RWS Publications, 2009.

[21] S.J. Mellor, K. Scott, A. Uhl, D. Weise, MDA Distilled: Principles of Model Driven Architecture. Addison Wesley Professional, 2004.