

# Analyzing Strategic Business Rules through Simulation Modeling

Elena Orta<sup>1</sup>, Mercedes Ruiz<sup>1</sup> and Miguel Toro<sup>2</sup>

<sup>1</sup> Department of Computer Languages and Systems  
Escuela Superior de Ingeniería  
C/ Chile, 1

11003 – Cádiz, Spain

<sup>2</sup>Department of Computer Languages and Systems  
Escuela Técnica Superior de Ingeniería Informática  
Avda. Reina Mercedes, s/n  
41012 – Sevilla, Spain

{elena.orta, mercedes.ruiz}@uca.es, migueltoro@us.es

**Abstract.** Service Oriented Architecture (SOA) holds promise for business agility since it allows business process to change to meet new customer demands or market needs without causing a cascade effect of changes in the underlying IT systems. Business rules are the instrument chosen to help business and IT to collaborate. In this paper, we propose the utilization of simulation models to model and simulate strategic business rules that are then disaggregated at different levels of an SOA architecture. Our proposal is aimed to help find a good configuration for strategic business objectives and IT parameters. The paper includes a case study where a simulation model is built to help business decision-making in a context where finding a good configuration for different business parameters and performance is too complex to analyze by trial and error.

**Keywords:** Service-Oriented Development, Business Rules, Simulation Modeling.

## 1 Introduction

Service-oriented applications are built as a set of business processes and business process flows. Business process flows are in charge of orchestrating the different services, frequently web services, which are used to give response to the business requirements [1]. A very simple definition for a web service defines it as a programming subroutine that happens to be available over the Internet, and that offers a number of advantages such as location independency, standardized access protocol, platform-independency, is highly configurable and easy to evolve to adapt new business needs [2]. In order to achieve the high level of flexibility business applications need, business rules are the instrument chosen to build and link together these dynamic and flexible services. Thus, changes in the business can be translated into the business rules resulting in a new combination of services that respond to the new business requirements without having to change much code.

Understanding and evaluating risks and rewards is now more than ever necessary to manage such flexible architectures to assure that the effect of decisions leads to improvement and benefits. Simulation techniques are known to be useful tools to help evaluate the impact of process changes or help in the design of new ones. As a consequence, the need for simulation and optimization is receiving a growing interest from the service oriented environment and vendors are offering tools to help model and simulate business processes as well as business rules.

In this paper, we propose the utilization of simulation models to model and simulate strategic business rules that are then disaggregated at different levels of a Service Oriented Architecture (SOA). Our proposal is aimed to act as a complementary tool to the available systems. It is aimed to help finding a good configuration for strategic business objectives and IT parameters that can help meet business rules and performance requirements.

This paper is structured as follows: Section 2 provides definitions for the concepts of business agility and business rules and describes a classification for business rules. Section 3, gives an overview of the related works found regarding the application of system dynamics simulation models and SOA. Section 4, describes a case of study to illustrate the application of system dynamics simulation to define strategic managerial business rules. Finally, Section 5 summarizes the paper and draws the conclusions.

## **2 SOA, Business Agility and Business Rules**

Within the SOA approach, it is necessary to distinguish between process logic and decision logic. While process logic is finally supported by means of orchestrated services, decision logic is normally represented by rules that can, or better said should, be finally supported by means of orchestrated decision services. For instance, in a given e-commerce business, a certain set of orchestrated services can support the logic of a business process called *Orders*, responsible for receiving customer orders through the business portal and dispatching the products ordered. To make these actions possible, the business process needs from the decision logic generally provided by the form of business rules that determine how to rule this business process.

Business rules are often referred to be at the border side of business engineering and software engineering. This fact is also highlighted by the Business Rules Group in the definition they provide for the concept of Business Rule, which clearly depends on the perspective one is following [3]: “From the business perspective, a business rule is guidance that there is an obligation concerning conduct, action, practice, or procedure within a particular activity or sphere. From the information system perspective, a business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business.”

There are different kinds of orthogonal classification for business rules. Some of them attend principles of soft or hard coding, or attempt to classify business rules under the information system perspective (e.g., base rules, that can be of one of the following types: derivation, constraint, invariant and script, and classifier rules).

However, the former classification does not seem to clearly provide a mechanism for business rules as described from a business perspective.

For the purpose of this study, we part from the classification proposed by Weiden and colleagues [4]. According to their proposal, business rules should be classified attending their semantic properties, that is, the role they play in the business process. Three categories are proposed for the business rules: a) structural, to describe static aspects of a business, b) behavioral, to describe the conditions of execution of tasks, and c) managerial, to define higher-level constraints on the business. In our view, this classification integrates the perspective of business and IT into one comprehensive schema of classification, being the structural and behavioral rules directly related to the IT and the managerial ones to the business perspective. However, these categories are not isolated but interdependent ones. That is, a managerial business rule is often translated into a set of structural and, mostly, behavioral rules.

### **3 Related works**

This section overviews some of the current contributions that apply simulation modeling techniques in the SOA context.

Jeng and An [5] propose the use of dynamic simulation models in SOA project management. They present a framework for managing SOA projects and how system dynamics simulation can enhance the effectiveness and agility of SOA project management. In [6], the authors present dynamic simulation modeling as a complementary technique for business requirement identification. [7] proposes a collection of heuristics and guidelines for the development of dynamic simulation models based on given business process models. [8] uses the simulation modeling techniques to present a framework for web service management. Finally, [9] presents a business-driven analysis method for business service development in the context of SOA by using the System Dynamics method to model services and simulate their behaviors.

To our knowledge, the originality of our proposal resides in the fact that it is aimed to help join business strategy decision-making with the technical issues of IT implementation. It is not used to test the behavior of decision services as in [9] but to help business management evaluate the fulfillment of strategic decisions and their impact in application performance.

### **4 Case Study**

This section includes a case study that helps to illustrate our proposal. We part from a problem description and a concrete business rule for a company. The simulation model built is aimed to help business decision-making when finding a good configuration for different business and performance parameters is too complex to analyze by trial and error.

#### 4.1 Problem Description

For the purpose of this study let us assume a hypothetical e-commerce company that sells products on the Internet. The company plays a distribution role by buying the products to their manufacturers and selling them to their customers who place orders through the company website.

As it was mentioned before, in SOA projects the main focus is on business needs. These business needs can be expressed in the form of managerial business rules that aggregate different business rules that affect the structure and behavior of the system. One of the most common managerial objectives of service-oriented business is customer satisfaction. Customer satisfaction can be described as a function of many interrelated factors. For our e-commerce company, customer satisfaction can be a function of factors such as the website usability, its functionality, availability and interactivity, the time needed to deliver the products, the system response during customer interaction, and the system capacity for processing orders, among others. It results clear that even though all these factors contribute to the fulfillment of our general business rule, not all of them can be dealt with at the same level of the service-oriented architecture. For the purpose of this paper, we will focus on one of these business rules that lead to customer satisfaction: the ability of the system to process the orders placed by the company's customers. Again, this ability depends of several factors including our in-home database services, the performance and availability of the company's servers and the response of external services, among others.

One of the main features of service-oriented development is that invoked services are platform and location independent. Issues such as performance measurement, priorities, responsibilities and problem resolution, availability, operation or billing model figure in the contract the service provider and a client company subscribe called Service Level Agreement (SLA).

In our case, one of the outsourced services will be the service to validate if the customer's credit card has enough credit to cover the purchase. Estimating the desired performance of this outsourced service is not an easy task since it is highly influenced by the market, the selling policy of the company and its effects on the tendency of customer's orders.

Among the different parameters that help define the service capacity in an SLA, the following are among the most frequent [1]:

- Abandon Rate (ABA): Percentage of calls abandoned while waiting to be answered.
- Average Speed to Answer (ASA): Average time (usually in seconds) it takes for a call to be answered.
- Time Service Factor (TSF): Percentage of calls answered within a definite timeframe, e.g. 80% in 20 seconds.
- First Call Resolution (FCR): Percentage of incoming calls that can be resolved without the use of a callback, or without having the caller call back.

Depending on the values estimates and specified for the former parameters, the billing model and the quantity the company has to pay to the service provider will vary. If these parameters have been under or over-estimated they will have a direct effect not

only on the bill but on the application performance, the customer satisfaction and, eventually, on the market position of the company.

With the aim of helping managers in this decision-making process, a simulation model is built to allow *playing* with different service capacities and customer-order tendencies in order to analyze their effects on the fulfillment of our business objective.

## **4.2. Simulation Model Building**

Following Kellner's proposal for describing simulation models [10], this section describes the simulation model built to analyze this problem.

### **4.2.1. Model proposal and scope**

The simulation model is built to help analyze on a qualitative manner the fulfillment of one of the business rules that leads to the achievement of customer's satisfaction. This business rule is: *Allow no more than 15% of rejected orders*. The scope of the model is a portion of the life cycle. Its organization breadth is multiple projects and its time span is short, since we are dealing with seconds as a unit. The simulation timeframe is ten minutes.

### **4.2.2. Result variables**

The main result variable that provides information regarding the simulation model objective is called *Degree of Business Objective Fulfillment*. It expresses the difference between the orders that have been rejected by the system due to an underestimated credit validation capacity and the maximum rejection rate allowed by the business rule (15% in this case study).

Other result variables can also be helpful to understand what is happening in the system during the simulation timeframe. They are the following:

- *Orders received*: The number of customer's orders received in the website. This number will depend on the tendency of customer's orders.
- *Validated orders*: The number of received orders that have been successfully credit-validated.
- *Rejected orders*: The number of received orders that have been rejected because they have exceeded the waiting time established by the company without being able of getting an answer from the credit validation service. Orders rejected for this reason are due to an underestimation for the credit validation capacity that has been contracted with the service provider.

### **4.2.3. Input parameters**

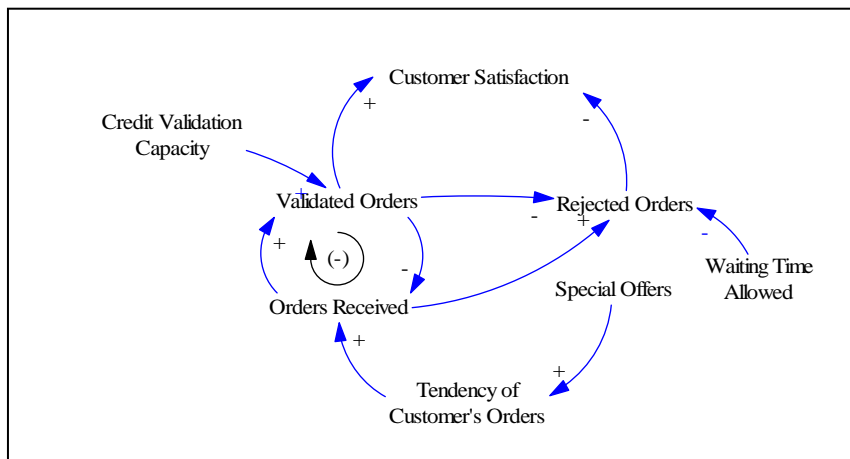
Input parameters allow us to configure different scenarios to test the effects of contracting different credit validation capacities in different tendencies of customer's

orders. The following input parameters are used to configure the different scenarios for the simulation runs included in this study:

- *Credit Validation Capacity*: It represents the maximum capacity the company has contracted with the credit validation service provider.
- *Waiting Time Allowed*: It holds the value for the maximum time a received order is allowed to wait for the credit validation service response. Once this time is exceeded, the order is rejected.
- *Tendency of customer's orders*: It represents the effect of customer's reaction to the launching of a special order on the number of orders received in the company website.

#### 4.2.4. Process abstraction

Fig. 1 illustrates the main variables of the model and their interrelationships. It shows that *Customer Satisfaction* depends on the number of *Validated Orders* (the higher, the better), and the number of *Rejected Orders* (the lower, the better). *Validated Orders* depends on the number of *Orders Received* and the *Credit Validation Capacity*. The number of *Orders Received* depends on the current *Tendency of Customer's Orders* which reflects customer's reaction to the presence of, for instance, special offers in the website. Finally, *Rejected Orders* depends on the number of *Orders Received* that has been waiting for more than the maximum *Waiting Time Allowed* as well as the number of *Validated Orders*.



**Fig. 1.** Causal Diagram with main interrelationships.

In the running model, *Orders Received*, *Validated Orders* and *Rejected Orders* are modeled as level variables whose behavior is controlled by the flow variables *Order Rate*, *Validation Rate* and *Order Rejection Rate*. The input parameters of the model are the acting elements upon which the fulfillment of the business rule can be assured.

### 4.3. Simulation Runs

This section includes different simulation runs resulting from different patterns of behavior of the input variable *OrderRate*.

- CASE 1 (CONST): Assumes that after the launching of a special offer, the mentioned rate experiments a rapid grow and then remains constant at its peak value.
- CASE 2 (RAMP): Assumes a similar behavior for the order rate, but the increment is not step-shaped but gradual, hence it has been modeled as a ramp.
- CASE 3 (PULSE): Assumes that after the launching of a special offer the order rate grows rapidly, stays at its peak value for a while and then descends gradually.

Fig. 2 shows graphically the patterns of behavior considered for this input rate.



**Fig. 2.** Patterns of behavior for input variable *OrderRate*.

The simulation runs shown in this section represent the deviation of the output variable *OrderRejectionRate* respect the initial business objective. It is important to notice that we are focusing only on the rejections due to orders that have been waiting for a credit validation for more than the maximum time allowed by the company. Other reasons for rejections such as server down or incorrect credit card data are not included since they are not affected by the input parameters of this simulation model. The results obtained in the different scenarios follow.

#### CASE 1: *OrderRate* grows rapidly and then stays constant (CONST)

##### SCENARIO CONFIGURATION

- ***OrderRate (OR)***: Table 1 shows the features of the order rate assumed in CASE 1 for the tendency of customer's order.

**Table 1.** Order Rate behavior in CASE 1.

Initial OR	Increment for OR	Pulse Begins at	Pulse Width
500 orders/minute	1000 orders/minute	1 minute	10 minutes

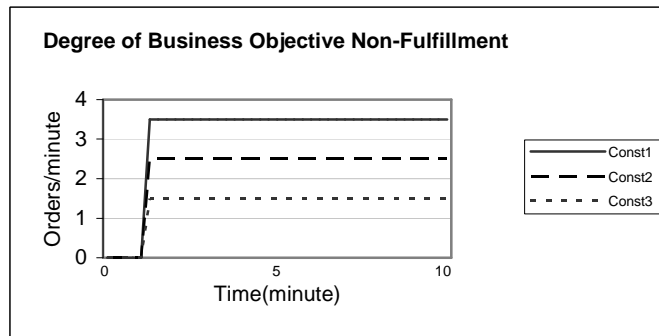
- ***CreditValidationCapacity (CVC)***: Table 2 shows the different values for this parameter used to configure four different scenarios (named, Const1-Const4).

**Table 2.** Credit Validation Capacity values for CASE 1.

Scenario	Const1	Const2	Const3	Const4
CVC (orders/minute)	500	600	700	900

### SIMULATION OUTPUT

Fig. 3 shows the different results obtained for the degree of non-fulfillment for the business objective for each of the scenarios previously described. The variable graphically displayed shows the difference between the *OrderRejectionRate* and the business objective. This figure does not show the results for the scenario *Const4* since in this case the deviation respect the objective is always zero, that is, the business objective is always met during the simulation time frame.



**Fig. 3.** Simulation output for CASE 1.

Looking at the results for the rest of scenarios, the business objective is met in all of them before the increment of the order rate takes place, since  $OR \leq CVC$ . However, once this increment is achieved, the business objective is only met in scenario *Const4* and for that, a  $CVC=900$  orders/minute is needed. *Const1* is the scenario with the maximum deviation from the objective (350 orders/minute), while *Const2* has a deviation of 250 orders/minute and *Const3* presents a deviation of 150 orders/minute.

Therefore, the simulation runs offer the expected results for the given inputs and constraints: the higher the increment of the order rate during the special offer, the higher the capacity needed to validate the credit for incoming orders.

### CASE 2. *OrderRate* grows gradually and then stays constant (RAMP)

#### SCENARIO CONFIGURATION

- ***OrderRate (OR)*:** In this case, the gradual grow of the order rate is modeled by a ramp with a different slope and length in each scenario. Three main groups of simulations have been run (Case21 –Case23) as shown in Table 3.



**Table 3.** Groups of simulations for CASE 2.

	<b>Initial OR (orders/minute)</b>	<b>Ramp Slope</b>	<b>Ramp Length (minutes)</b>
<b>Case21</b>	500	1	2
<b>Case22</b>	500	2	2
<b>Case23</b>	500	1	3

***CreditValidationCapacity (CVC):***

Table 4 shows the different values for CVC and the former group of simulation leading to the definition of twelve different scenarios.

**Table 4.** Scenarios simulated for CASE 2.

<b>CVC</b>	<b>Case21</b>	<b>Case22</b>	<b>Case23</b>
500 orders/minute	Ramp11	Ramp21	Ramp31
600 orders/minute	Ramp12	Ramp22	Ramp32
700 orders/minute	Ramp13	Ramp23	Ramp33
900 orders/minute	Ramp14	Ramp24	Ramp34

**SIMULATION OUTPUT**

Fig. 4 shows the different results obtained for the degree of non-fulfillment for the business objective. The outputs for the scenarios Ramp12, Ramp13, Ramp14, Ramp24, Ramp33 and Ramp34 do not appear in this figure since in all of them, the deviation is zero and hence the business objective is always met.

Analyzing the scenarios where a non-fulfillment is found, it can be seen that for a CVC = 500 orders/minute, the business objective is not met at any of the scenarios in which CVC takes that value (Ramp11, Ramp21, and Ramp 31), having Ramp 21 both the largest deviation and the soonest apparition of the deviation. When CVC = 600 orders/minute, the objective is not met at Ramp22 and Ramp32 scenarios, having in this case Ramp22 both the largest deviation and the soonest it appears. When CVC = 700 orders/minute, the only scenario in which the objective is not met is Ramp23. Finally, when CVC=900 orders/minute, the objective is met in every scenario. Therefore, it can be concluded that for a certain CVC, the larger the slope for the order rate and the sooner that growing begins, the larger the deviation respect the business objective is. Besides, the deviation respect the objective grows with the length of the slope for the order rate.

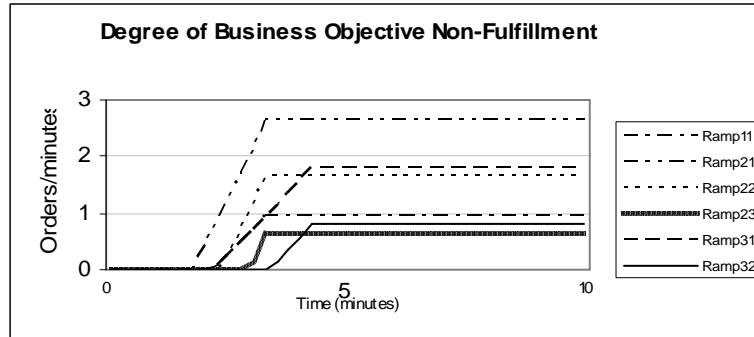


Fig. 4. Simulation output for CASE 2.

**CASE3: OrderRate grows gradually, stays constant, and then decreases gradually (PULSE)**

#### SCENARIO CONFIGURATION

- **OrderRate (OR):** In this case, the gradual grow of the order rate is modeled by a pulse with different heights and widths in each scenario. The three main groups of simulations that have been run are shown in Table 5.

Table 5. Groups of simulations for CASE 3.

	Initial OR (orders/minute)	Increment (orders/minute)	Step Begins at	Pulse Width (minutes)
Case31	500	1000	1 minute	6
Case32	500	1000	1 minute	4
Case33	500	700	1 minute	6

- **CreditValidationCapacity (CVC):** Table 6 shows the different values for CVC and the former groups of simulation leading to the definition of nine different scenarios.

Table 6. Scenarios simulated for CASE 3.

CVC	Case31	Case32	Case33
500 orders/minute	Pulse11	Pulse21	Pulse31
700 orders/minute	Pulse12	Pulse22	Pulse32
900 orders/minute	Pulse13	Pulse23	Pulse33

#### SIMULATION OUTPUT

Fig. 5 shows the different results obtained for the degree of non-fulfillment for the business objective. The outputs for the scenarios Pulse13, Pulse23, Pulse32 and Pulse33 do not appear in this figure since in all of them, the deviation is zero and, hence, the business objective is always met.

Analyzing the scenarios where non-fulfillment is found, it can be seen that for a CVC = 500 orders/minute, the objective is not met in scenarios Pulse11, Pulse21 and Pulse31. Pulse11 presents the largest and longest deviation. When CVC=700 orders/minute, the business objective is not met in Pulse12 and Pulse22. Even though the deviation respect the business objective is the same in both scenarios, the deviation is Pulse 12 lasts longer than in Pulse 12. Finally, when CVC=900 orders/minute, the objective can always be met. Therefore, it can be concluded that for a given CVC, the larger the pulse width, the less time the business objective can be fulfilled. Besides, the larger the pulse height, the larger the deviation respect the business objective.

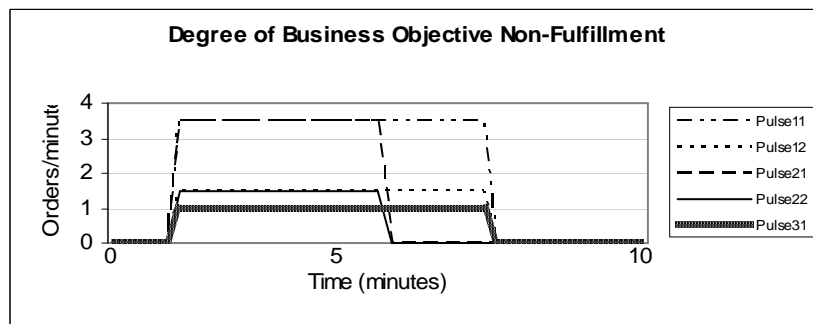


Fig. 5. Simulation output for CASE 3.

## 5 Conclusions and Further Work

In this work we have presented the first results of a research effort aimed at applying simulation modeling in service-oriented architectures. In this paper, a simulation model was developed to help analyze the consequences of under- or overestimation of the capacity of an outsourced service reflected in its SLA on the application performance and hence customer satisfaction. To do that, different scenarios were configured by varying the outsourced service capacity and the tendency of customer's orders that tried to reflect the effect of launching a special offer at a company website.

The simulation model helps mainly to design the suitable combination of service capacity in a customer's demand context to satisfy a business objective (allow no more than 15% order rejection). Other studies can also be made with this simulation model such as sensitivity simulations or optimization studies in the context of determining the suitable billing model.

Our future work is mainly focused on the development of simulation models to help in decision-making in different domains of service-oriented architectures as well as the service-oriented development process. In this sense, new features will be added to the model presented in order to better resemble real life projects and applications. After developing the models, we intend to apply them in real companies to help both

calibrate and validate the simulation models and provide the benefits of its usage for their potential users.

**Acknowledgments.** This research was partly supported by the Spanish Ministry of Education and Science and the European FEDER funds under project TIN2007-67843-C06-04.

## References

1. Marks, E.A., Bell, M., *Service-Oriented Architecture: a Planning and Implementation Guide for Business and Technology*. John Wiley & Sons, Inc., New Jersey (2006)
2. SOA World Magazine, <http://soa.sys-con.com>
3. Business Rules Group, <http://www.businessrulesgroup.org/defnbrg.shtml>
4. Weiden, M., Hermans, L., Schreiber, G., van der Zee, S., *Classification and Representation of Business Rules*. <http://www.omg.org/docs/ad/02-12-18.pdf> (2002)
5. Jeng, J.J., An, L., *System Dynamics Modeling for SOA Management*. In: IEEE International Conference on Service-Oriented Computing and Application, pp. 286-294. IEEE Computer Society, Washington, DC (2007)
6. An, L., Jeng, J.J., Gerede, C.E., *On Exploiting System Dynamics Modeling to Identify Service Requirements*. In: IEEE International Conference on Services Computing, pp. 277-280. IEEE Computer Society, Washington, DC (2006)
7. An, L., Jeng, J.J., *On Developing System Dynamics Model for Business Process Simulation*. In Winter Simulation Conference, pp. 2068-2077. Winter Simulation Conference, Florida (2005)
8. An, L., Jeng, J.J., *Web Service Management Using System Dynamics*. In IEEE International Conference on Web Services, pp. 347-354. IEEE Computer Society, Washington, DC (2005)
9. An, L., Jeng, J.J., *Business-Driven SOA Solution Development*. In: IEEE International Conference on e-Business Engineering, pp. 439-444. IEEE Computer Society, Washington, DC (2007)
10. Kellner, M.I., Madachy, R.J., Raffo, D., *Software process simulation modeling: Why? What? How?* J. Syst. Software, 46(2-3): 91-105 (1999)