Approach to Business-Policy based Job-Scheduling in HPC

Eugen Volk
High Performance Computing Center Stuttgart (HLRS)
Stuttgart, Germany
volk @ hlrs.de

***Abstract*—Job-Scheduling behavior of a High Performance Computing (HPC) provider is typically defined in the way that implicitly corresponds to its business policies. Represented mainly by a set of business rules or objectives, business policies form means to guide and control the business of HPC service provisioning. Because in HPC domain business policies exist mostly implicitly, administrators configure schedulers intuitively and subjectively. This makes it hard for business people to assess whether the actual scheduling behavior corresponds to current business policies, as there is no link defined between job-scheduling and business policies. The question, whether the scheduling behavior is configured correctly, cannot be answered without providing relationships between business-policies and job-scheduling strategies. Hence, more general question is: how much influence does business policy actually have on job scheduling? In this paper, we present an approach allowing investigating how business-policies relate to the job-scheduling in HPC domain.**

***Keywords—Business-policy, Job-Scheduling, Policy-based Management*.**

I. INTRODUCTION

Job-Scheduling behavior of a High Performance Computing (HPC) provider is typically defined in the way that implicitly corresponds to its business policies. Thereby, business-policies are a type of formal or informal behavioral guide prescribing behavior in a company, thus forming means to guide and control the business. Business-policies are usually formed by a set of business rules, business objectives, or in general, statements of control guides for delegated (to human or machine) decision making [1]. Business policies in the context of HPC affect several domains, such as security, accounting, SLAs, contracting, and others. They might have direct or indirect influences on job-scheduling. For instance, a business policy, such as “all jobs of premium customers have to be completed within 12 hours” has direct influence on scheduling, by determining the latest deadline of the job. Business-policies in HPC domain exist in most cases not explicitly, i.e., written by using domain specific language or natural language, but implicitly in the mind of the business people, whereas the configuration of job-scheduler is done by administrators.

The range of existing schedulers used for job scheduling in HPC varies from time-based scheduler like Cron [4] to advanced policy-based schedulers like Moab [3] or its open-source variant Maui, which support large array of scheduling policies. Scheduling policies define thereby behavior of the scheduler by, i.e., assigning priority to a job depending on job-size (number of CPUs or cores required), estimated job-duration, user’s priority and other factors. However, schedulers have a big amount of parameters and different scheduling policies which need to be selected and adjusted in order to meet business policies in different situations.

A problem occurs when administrators are configuring schedulers. The configuration of schedulers is done in most cases intuitively and subjectively, because of implicit business policies, system administrators unaware of them, or in general, because of missing link or mapping between business policies and selection and configuration of scheduling policies. This makes it hard for business people to assess whether the actual scheduling behavior corresponds to current business policies, as there is no link between scheduling policies and business policies and it requires understanding of scheduling configuration parameters. The question, whether the scheduling behavior is configured correctly, can be answered by providing relationship between business-policies and job-scheduling policies. In this paper, we describe an approach allowing investigating how business-policies relate to job-scheduling in HPC domain and present intermediate results.

This paper is structured as follows. Section II presents related work in the area of job scheduling in HPC, policy-based management, and SLA based scheduling. Section III provides background information on job-scheduling in HPC. In Section IV we discuss the problem related to alignment of scheduling behavior with the business policies, showing the need for business-policy-based job-scheduling in HPC. Section V presents approach allowing investigating how business-policies relate to the job-scheduling in HPC and solve the problem described in previous section. Section VI provides intermediate analysis results achieved by applying proposed approach, identifying key-factors and relating them to business policies. Finally, the last section summarizes this paper and outlines work in progress.

II. RELATED WORK

In the target-area of “business-policy based job-scheduling in HPC” currently no work is known to the author. But, there is a weak relationship between “business policy-based job-scheduling” and “SLA (Service Level

 ISBN: 978-1-61208-106-9

Agreement) based job-scheduling", which is outlined below. However, there has been much work done in related areas: Job-scheduling in HPC, policy-based management, business-policies and SLA based job-scheduling.

In the area of job-scheduling in HPC, Iqbal, Gupta, and Fang [6] offer an overview about scheduling algorithms used for job-scheduling in HPC clusters. In [7], Casavant and Kuhl provide taxonomy of scheduling strategies in general-purpose distributed computing systems. In [8], Yeo and Buyya provide taxonomy of market-based resource management systems, citing over 79 references. In [9], Abawajy describes recent advances in efficient adaptive scheduling policies.

Many solutions in the area of policy-based management have been proposed. In [10], Boutaba and Aib provide history of policy-based management, referencing over 118 papers. In IBM's autonomic computing reference architecture [11], the authors drafted the principle on how policies on high level might be used to express business needs/objectives that govern IT infrastructure operations.

In the area of SLA-based job-scheduling many papers have been published. SLA is part of a service contract where the level of services or quality of services (QoS) is formally defined and agreed between service providers and customers. SLA contains usually rewards, for successful fulfillment of SLA, and penalties in case of SLA violations. SLAs are contracted in accordance with business-policies. In contrast to SLAs, business-policies prescribe, among others, kind of services and spectrum of QoS which can be offered principally to customers. Hence, SLAs can be considered as service level objectives contracted in accordance with the business-policies. On the other hand, business-policies are more prescriptive than SLAs, as SLA might be violated due to various reasons, but the behavior in a company must follow provider's business-policy. In [12][13][14][15], QoS and SLAs are used to find and allocate desired resources in quantity and quality, and determine priority and order of jobs for scheduling, among others, based on rewards and penalties declared in SLAs. In [13], authors describe how to derive IT management policies from SLAs, which in general follows autonomic computing approach (management by objectives).

In IBM's Whitepaper [16] authors provide most recent definitions of policies and rules in business area, relating them to IT. According to that definition, a "business policy" is a type of business directive that expresses the course of action that the business wants to have happen within a set of business conditions [16].

In conclusion, related work presented in this section outlined achievements needed to accomplish "business policy-based job-management in HPC" approach. As there is no work currently exist in the area of "business policy-based job-scheduling in HPC", we identified work in related areas. Approach in the area of policy-based management outlines hierarchical policy refinement process that transforms high-level policies into low-level policies. Similar methodology is used to achieve business-policy based job-management in HPC. Thereby, business policies represent high level policies that need to be transformed to low-level job-scheduling policies.

The bottom of the "business-policy based job-management in HPC" approach is formed by a work achieved in job-scheduling in HPC. This work presented scheduling algorithms and identified performance indicators needed to assess scheduling algorithms and policies. The top of the approach is formed by business policies, described and defined in IBMs' Whitepaper [16]. SLA-based job-management methodology applied policy refinement approach to SLAs and resource allocation policies in HPC domain. SLA-based job management methodology demonstrated how to allocate desired resources in quantity and quality for particular jobs, and determined, i.e., by sorting jobs according to rewards, penalties or deadlines stated in SLAs, the order of jobs. However, in contrast to SLAs, which are mostly related to fulfillment of the single job or set of jobs, business policies are type of business rules and directives used to control the whole process of HPC provisioning, involving other domains, such as: contracting, security, customer management, accounting, and resource management; they influence directly or indirectly job-scheduling behavior.

III. BACKGROUND

The cluster infrastructure of computing centers can be divided in two classes: high-throughput computing cluster and high performance computing cluster [6]. Nodes in high throughput computing clusters are usually connected by low-end interconnections. In contrast, more powerful nodes in high performance computing (HPC) cluster are interconnected by faster interconnection with higher bandwidth and lower latency. The application profile of high-throughput computing clusters includes loosely coupled parallel, distributed or embarrassingly parallel applications, requiring less communication and synchronization between nodes during the calculation. In contrast, the application profile of HPC clusters consists mainly of tightly coupled parallel applications, with high communication and synchronization requirements.

The computing nodes in cluster are managed by a resource management system (RMS), which is responsible for resource management, job queuing, job scheduling and job execution. Firstly, users who are willing to submit their applications or programs to resource management system need to express their applications as computational jobs, specifying requirements using, i.e., Job Submission Description Language (JSDL). Job specification contains usually number of nodes/CPUs/cores required, estimated maximum job-runtime, target architecture type (i.e., vector or scalar), specific I/O requirements (i.e., tools and files required for job execution) and other application or platform specific parameters. After expressing application as a job, user submits the job in batch to queue of the resource management system, where it waits in the queue with the jobs of other users, until it is scheduled and executed. Typically, a resource management system is comprised of a resource manager and a job scheduler [6]. Most resource managers have an internal, built-in job scheduler, which can be substantiated by external scheduler with enhanced capabilities [6], i.e., with support for various scheduling policies like Maui [2]. Resource manager provides scheduler with information about job-queues, loads on compute nodes, and resource availability. Based on that information, scheduler decides on how and when to allocate resources for job execution. The decision of the scheduler follows scheduling policy that determines the order in which the competing users' jobs are executed. The order of jobs typically depends on job-size (amount of resources i.e., processors/cores required), estimated maximum job-runtime (indicated by user), resource access permission (established by administrator), resources available, and might depend additionally on QoS parameters (i.e., response time) expressed in contracts or SLAs. The assessment of scheduling behavior is typically done according to the following performance indicators [6][8][18]:

- **Wait time**: the time a job has to wait before the execution of the job starts
- **Response time**: how fast the user receives a first response from the system after the job is submitted
- **Turnaround time**: total time between when the job is submitted and when the job is completed. It includes wait time and execution time of the job.
- **Resource Utilization**: reflects the usage level of the cluster system
- **System Throughput**: number of jobs completed in a period of time

Typical performance criteria for users who expect minimal response time is the mean response time [6]. In contrast, administrators are typically trying to achieve maximum overall resource utilization, as that maximizes return on investment (ROI). Improving overall resource utilization and at the same time decreasing response time are two conflicting goals, as it requires that shortly submitted jobs with higher priority are executed as soon as possible, thus reducing the optimization space for efficient resource utilization.

IV. NEED FOR BUSINESS-POLICY-BASED JOB-SCHEDULING

Business policies are control statements that guide behavior in a company and control the business. Business policies are defined usually at an overall strategic level and can be related to specific areas. In HPC domain, these areas are: security, contracts and SLAs, resource management, accounting, and others. Business policies, which relate to security, contain statements governing the access to HPC resources, i.e., prescribing the process of obtaining permission to HPC resources, granting, restricting or refusing the access. Contract and SLA business policies contain statements, which i.e., describe the spectrum of performance and capacity capabilities of HPC provisioning offered principally. Resource management business policies contain statements influencing resource allocation and scheduling behavior on high level, i.e., by prescribing the preferences between users-groups.

As already mentioned, scheduling behavior is typically defined in the way that it implicitly adheres to business policies of HPC providers, while taking users' job requirements, available resources, existing SLAs, long term contracts and other factors into account. Advanced policy-based schedulers like Maui [2] have a big amount of parameters and different scheduling policies which need to be selected and adjusted in order to meet all business policies in different situations. As business policies exist mostly implicitly in the mind of people, or because administrators are not really aware of all of them, they configure schedulers intuitively and possibly subjectively. This makes it hard for business people to assess whether the actual scheduling behavior is correct and corresponds to current business policies, as there is no link between business policies and scheduling policies defined.

Additionally, there might be a fast switch required between different business policies. For instance, in profit oriented organizations, managers try to achieve maximum return on investment which often means that they only deliver various qualities of services to various users and groups [2] to increase system utilization. In contrast, nonprofit organizations, like national computing centers, have their focus on delivering various qualities of services to various (or certain) users and groups, even if this will cause a decreasing utilization. For instance, there could be situations where the cluster resources need to be exclusively reserved to a certain user, although no jobs on reserved resources might be executed during the reserved time-period. Some of the national computing centers have joint collaboration with scientific and industrial partners through common joint cooperation company. That means the scheduling behavior in clusters of such computing centers needs to be flexible enough to be adapted to various business needs, even at the same time.

Furthermore, there are cases where the usual job-scheduling behavior must be adapted to changing situations and requires evaluation of several business policies. For example, in case of fall-out of the cluster on which jobs of industrial users are executed, these could be shifted to another cluster, if allowed. The answer on the question whether the jobs of industrial users might be shifted, i.e., to research cluster, on which jobs of students or researchers are executed, depends thereby on evaluation of several business policies and facts. Research and educational clusters are typically financed by federal authority, whereas clusters used for industrial calculations are financed through common joint cooperation company. In case of the business policies, which prescribe that (1) industrial partners have higher importance than students or researchers, (2) only the owner (who has financed it) of the cluster may decide on its usage, and (3) current contract between federal land and HPC provider that allow usage of maximum 50 % of the cluster per month for industrial jobs, then the shifting of industrial jobs to the research cluster is allowed only if the 50 % limit is not exceeded.

As stated, there are many different business policies from different areas, which need to be considered when configuring schedulers. Furthermore, there might be a fast switch between different business policies required, and a fast adaptation of the scheduling behavior dependent on evaluation of several business policies from different domains. Because of implicit existence of business policies and missing link between business policies and scheduling policies, there is a risk of resulting incorrect scheduling behavior. In order to reduce the risk of miss-configuration we present an approach enabling to identify the link between business policies and scheduling policies, in next section.

V. Approach

An approach to handle problems described in previous section, induced by changing business objectives or altering situations, might follow IBM's autonomic computing reference architecture [11]. Autonomic computing is thereby defined "as a computing environment with the ability to manage itself and dynamically adapt to changes in accordance with business policies and objectives" [11]. Following this approach, there must be (1) business policies defined, capable to express business requirements influencing scheduling behavior on high level. Once, there are business policies defined, the next step (2) consist then of transforming these business policies with other sources (as SLA, Contracts, Accounting, etc.) influencing scheduling behavior into scheduling policies to configure advanced policy-based schedulers like Maui or Moab. In order to define business-policies explicitly, there must be HPC business policy specification language elaborated, capable to express business needs for various situations.

In order to address this problem, we will follow a bottom-up process:

The first step (1) consists of the analysis of existing scheduling policies in HPC in order to identify performance indicators (as described in section III) and key-factors like priority of user/customer (i.e., dependent on SLAs), accounting data, available resources, fairness etc. which influence scheduling behavior. In order to assess whether analyzed scheduling-policies make sense, there should be business-policies identified, which describe analyzed scheduling behavior on high level.

The next step (2) involves the analysis of existing business policies of particular high performance computing provider, in order to identify relationship to performance indicators and key-factors identified in the first step (1). The outcome of the second step will be a model, which explains relationships between specific business policies of particular HPC provider, performance-indicators, key-factors and scheduling policies. Especially the relationship between existing business policies and scheduling policies will provide an overview on how policy refinement process of transforming business policies to scheduling policies will principally looks like.

The third step (3) comprises the identification of HPC business policy schema, derived from the model developed in second step (2), capable to express HPC business policies influencing job-scheduling. Elements of the identified schema are used in the domain specific language, such as TEMPORA [19], to capture and model business policy specifications.

Finally, in order to evaluate results achieved in previously steps, the last step consists of the reference implementation, enabling mapping of reference business policies together with other key factors to scheduling policy configuration for advanced schedulers such as Moab [3] or Maui [2]. The implementation of the transformation rules, needed to translate business policies into scheduling policies, may be implemented using prolog engine, such as XSB [20].

The approach described in this section outlined steps of the transformation between business policies and scheduling policies. In order to illustrate this approach, we will present examples for steps (1) and (2) in the next section of the paper.

VI. From Job-Scheduling-Policies to Business-Policies

In this section, we present intermediate results achieved by applying the first two steps of the described methodology. Firstly, we analyze briefly job-scheduling in HPC, presenting common scheduling algorithms and policies in subsections A and B. In Subsection C, we identify key-factors, influencing scheduling behavior from different point of views. In Subsection D, we investigate relationships between key-factors and business policies, illustrating mapping between business policies and job-scheduling policies in few examples.

A. *Analysis on Job-scheduling in HPC*

Job-Scheduling algorithms or policies can be divided in two classes: time-sharing and space-sharing [6]. Time sharing algorithms divide time on a processor into several slots, each time-slot is assigned to unique job then. In contrast, space-sharing algorithms assign requested resources to unique job, until job is completed. In all HPC clusters is space-sharing approach used mostly, as time-sharing approach increases synchronization overhead between nodes of the same job.

The simple space-sharing algorithms are [6] first in first out (FIFO), first come first serve (FCFS), shortest time job first (STJF), longest time job first (LTJF), largest job first (LJF) etc. FIFO and FCFS execute jobs in the order in which they enter the queue. In case, there are not sufficient resources available to start a job, FCFS waits, until required resources are available. STJF periodically sorts the incoming jobs in the queue assigning jobs with the shortest estimated running time for the execution. LTJF sorts periodically the incoming jobs and assigns jobs with the longest estimated running time for the execution. LJF sorts incoming jobs periodically assigning jobs with the highest number of nodes/cores required for the execution. Additionally, there might be a priority to each job assigned, with the aim to reduce response time, as job with higher priority are executed prior lower priority jobs.

The simple scheduling algorithms might be enhanced by combining them with the use of advanced reservation and backfill techniques. Advanced reservation algorithms use

 ISBN: 978-1-61208-106-9

estimated job-runtime to make reservation on resources for particular jobs and create time-schedule for certain time period. The problem thereby is that schedule is based on estimated job-runtime, which is in most cases much longer than the real one. That means the schedule needs to be adapted as soon as jobs are completed earlier than expected. The backfill strategy improves basic strategies by combining them with additional iteration to fill out the gaps. Given schedule on high priority jobs i.e., by applying LTJF strategy, the scheduler use in second iteration lower priority jobs to fill out the gaps (free time slots on unused resources) between higher priority jobs.

B. Policy-based Job-Scheduling

In order to enable administrator to control and adapt scheduling behavior (when, where and how resources are allocated to jobs) more fine granular and more quickly to different situations and needs, there exist policy based job-schedulers like Maui [2] and Moab [3]. Policies include in particular for Maui [2]: job prioritization, allocation policies, fairness policies, fairshare configuration policies, and scheduling policies. These are explained in detail in [2].

C. Identifying Key-Factors

Analyzing the scheduling algorithms and policies leads to identification of key-factors, characterizing (and determining) scheduling behavior from different point of view: customer, provider, and administrator of the cluster.

Customer-centric key-factors are: turnaround-time, response-time, meeting deadlines, and exclusive resource reservation.

Provider centric key-factors need to differentiate between maximum return on investment (ROI) and customer satisfaction. Hence provider centric key-factors are: resource-utilization (in case of max ROI) or high job-throughput, and customer satisfaction (trying to satisfy customer centric key-factors).

Administrator centric key-factors are: achieving provider's goal by configuring job-queue and identifying right scheduling policies based on provider's preferences.

In addition, there are independent key-factors, which form the scheduling situation: available resources (quantity and quality of resources), job complexity (quantity and quality of resources required for job-execution and job-execution-time).

These key-factors are in the next step related to business policies, explained in next section.

D. Key-factors and Business-Policies

As outlined in the previous section, there is a need for business policy specification capable to express business needs in order to adapt scheduling behaviour to new situations, without need to understand scheduling configuration parameters in detail. Considering all those key-factors identified in previous section from the business point of view, they might be divided in two categories *decision making* and *optimization of scheduling* behavior. Decision making affects principal question on how the customers (including what kind of customers/users) are supplied with services and what kind (QoS) of services can be offered/delivered to customers. Scheduling optimization criteria determine the focus of scheduling optimization. A possible taxonomy of business policies is outlined below.

Decision making includes prioritization (between users or their jobs), reservation of resources (to certain user), meeting deadlines, fairness, preemption.

Optimization of scheduling behavior comprises: optimization criteria (ROI / resource utilization, customer satisfaction, energy efficiency, etc.), prioritization between criteria and expression to what degree criteria might be not fulfilled.

Typical Business Policies might looks like and mapped to scheduling policies as follow:

Industrial users are preferred against the scientific users or students. This will be mapped to scheduling policies as follow: jobs submitted by industrial user-group have higher priority than the jobs of scientific or student user-group.

Jobs of GOLD customers must have response time of X hours. Jobs for all users of user-group GOLD must be started latest after X hours after the job submission. In order to fulfill such kind of business policy, there might be a dedicated job-queue for GOLD customers created. The amount of resources granted to GOLD job-queue depends thereby typically on mean job-size and amount of customers/users of type GOLD. In critical case there might additional resources allocated from other job-queues.

VII. SUMARY AND FUTURE WORK

In this paper, we outlined why business policy-based jobs-scheduling is needed, and presented an approach allowing to investigate how business policies relate to job-scheduling in HPC domain. The proposed bottom-up process explains identification of relationships between scheduling policies and business policies in several steps, including scheduling-performance-indicators, and key-factors. The process includes also elaboration of business policy language, capable to express business policies in HPC. The general aim of the proposed approach is to realize hierarchical policy refinement, allowing transformation of business policies together with other constrains into selection and configuration of parameters and policies needed to configure policy based schedulers. Intermediate results outlined in Section VI showed how identified key-factors, characterizing and determining scheduling behavior, might relate to business policies.

The approach and results presented in this paper are part of currently ongoing PhD work. The scope of ongoing and future work comprises all steps stated in Section V.

ACKNOWLEDGMENT

The results presented in this paper are partially funded by Federal Ministry of Education and Research (BMBF) through the TIMaCS project [17]. TIMaCS deals with the challenges in the administrative domain upcoming due to the increasing complexity of computing systems especially of

resources with performance of several petaflops [17]. The business policy-based management approach presented in this paper is part of the extended TIMaCS vision to use business policies for differentiated and goal oriented system management, while ensuring correct system management (in accordance with the business policies of the provider).

REFERENCES

[1] W. T. Greenwood, "Business Policy-Case Method Forum: A Rejoinder", in *The Academy of Management Journal*, Vol. 10, No. 2 (Jun., 1967), pp. 199-204

[2] Maui Scheduler Administrator's Guide, version 3.2 from http://www.clusterresources.com/products/maui/docs, access on 08.09.2010

[3] Moab Workload Manager user-guide, http://www.clusterresources.com/products/mwm/docs/moabusers.shtml, access on 03.09.2010

[4] Cron Wikipedia description, from http://en.wikipedia.org/wiki/Cron, access on 08.09.2010

[5] IBM, "Policies and Rules improving business agility", IBM website, http://www.ibm.com/developerworks/webservices/library/ws-policyandrules/index.html, access on 16.06.2010

[6] S. Iqbal, R. Gupta, and Y. Fang, *Planning Considerations for Job Scheduling in HPC Clusters*. Dell PowerSolutions, Feb 2005

[7] T. L. Casavant, G. J. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems". *IEEE Transactions on Software Engineering* 1988; 14(2):141–154.

[8] C. S. Yeo, R. Buyya, "A taxonomy of market-based resource management systems for utility-driven cluster computing." in *Software-practice and experiences* 2006; 36:1381–1419, Published online 8 June 2006 in Wiley InterScience

[9] J. H. Abawajy, "An efficient adaptive scheduling policy for high-performance computing", in *Future Generation Computer Systems*, Volume 25, Issue 3, March 2009, pp. 364-370.

[10] R. Boutaba, I. Aib, "Policy-based Management: A Historical Perspective", *Journal of Network and Systems Management*, pp. 447-480, Springer, 2007

[11] IBM, "An architectural blueprint for autonomic computing.", *IBM Whitepaper*, June 2006, http://www-01.ibm.com/software/tivoli/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf

[12] R. Sakellarioiu, V. Yarmolenko, "Job Scheduling on the Grid: Towards SLA-Based Scheduling." in *High Performance Computing and Grids in Action*, pp. 207–222. IOS, 2008.

[13] V. Yarmolenko, R. Sakellariou, "An Evaluation of Heuristics for SLA Based Parallel Job Scheduling." *3rd High Performance Grid Computing Workshop* (in conjunction with IPDPS 2006), 2006.

[14] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, R. Buyya, "Libra: Economy-Driven Job Scheduling System for Clusters.", in *Software: Practice and Experience* 2004; 34(6):573–590.

[15] L. Tang, Z. Yang, Z. Yu, Y. Wang, "A Quality-Driven Algorithm for Resource Scheduling Based on Market Model on Grid.", 2007 International Conference on Parallel Processing Workshops (ICPPW 2007)

[16] M. Hondo, J. Boyer, A. Ritchie, "Policies and Rules – Improving business agility: Part 1: Support for business agility", IBM Whitepaper, 16. March 2010

[17] TIMaCS - Tools for Intelligent Management for Very Large Computing Systems, web site: www.timacs.de

[18] Scheduling Wikipedia description, from http://en.wikipedia.org/wiki/Scheduling_(computing), access on 08.09.2010.

[19] P. M. Brien, M. Niezette, D. Pantazis, A. H. Seltveit, U. Sundin, B. Theodoulidis, G. Tziallas, and R. Wohed, „A Rule Language to Capture and Model Business Policy Specificatins., in *Proceedings of the third international conference on Advanced information systems engineering,* 1991, pp. 307 – 318.

[20] XSB sourceforge project web site, http://xsb.sourceforge.net/, access on 08.09.2010