Managing Knowledge in Small and Medium Sized Companies

- Lecture Notes -

by Vasile AVRAM, Prof. Dr.

Bucharest 2012

Contents

Chapter 6. Managing Knowledge in Small and Medium Sized Companies....................... 3
6.1. Introduction.. 3
The Business Environment 3
Decision Making Cycle.. 4
6.2. Factual and Procedural Knowledge 5
6.3. Business Rules and Business Rules Management Systems (BRMS) 8
6.4. Defining Controls by Rules 12
6.5. Logical Knowledge... 13
6.6. Introduction to ARulesXL 14
6.7. Usage of Business Rules Benefits... 17
References.. 19

Chapter 6. Managing Knowledge in Small and Medium Sized Companies

6.1. Introduction

The evolution of information processing paradigm during the past four decades to build intelligence and manage change in business functions has generally progressed over three phases: automation, rationalization of procedures, and reengineering. To these ones added the fourth phase, the today knowledge management (Avram and Avram, 2007):

1. Automation – increased efficiency of operations;
2. Rationalization of procedures – streamlining of procedures and eliminating obvious bottlenecks that are revealed by automation for enhanced efficiency of operations;
3. Reengineering – radical redesign of business processes that depends on information technologies – intensive radical redesign of workflows and work processes (redesign to e-business);
4. Knowledge Management - identifying, creating, representing, and distributing knowledge for reuse, awareness, and learning across the organizations.

We can postulate that the ultimate recipient of knowledge, and also the beneficiary of that, is the individual that uses these for a better "surviving" in a stuffing and suffocating informational environment. From a company point of view the knowledge is linked primary to employee or employees who carry it. According to the general theory of knowledge management this knowledge can be commonly categorized as:

- *Implicit* (or *tacit*, or *ill-defined*) that is specific to every employee and that is accumulated over years of experience and training. These knowledge is extremely valuable to the employer even is very vulnerable to loss (if the employee leaves the company in any way);
- *Explicit* when knowledge is defined, explained, or communicated to other person, even informally.

Related to these two categories we can say that the today knowledge management is focused in two directions:

- first one is to take the maximal advantage on the implicit knowledge by transforming that, as much as possible, into an explicit knowledge one;
- second one is to find unambiguous and automateable ways to represent the explicit knowledge.

To these two categories the IT creates the premises to determine, from the raw data existing in the company, the potential knowledge that raw data incorporates. The potential knowledge incorporated in raw data within the organization can be discovered or revealed and made explicit by usage of different BI techniques and tools, such as Data Mining, for example.

The Business Environment

In the post decade the business environment was altered with direct impact to the way business organized and managed by the following major worldwide changes:

a) Emergence and strengthening of the global economy with major change in the way the management and control are applied in a global marketplace, the competition take place in world market, the business activities implies cooperate global work groups and the usage of global delivery systems;

b) The transformation of industrial economies and societies into knowledge (and information) based service economy. There is a large migration from traditional business firms to new style business firm. Traditional business firms have hierarchical arrangement of specialists, standard operating procedures for delivering mass-produced products or services, and traditional managers relies on formal plans, rigid division of labour, and formal rules. New style business firm are flattened (less hierarchical), decentralized and uses flexible arrangement of generalists to deliver mass-customized products and services. The new manager relies on informal commitments and networks to establish goals (rather than planning), a flexible arrangement of teams and individuals working in task forces, and customer orientation to achieve coordination among employees. The new manager appeals to the knowledge, learning and decision making of individual employees to ensure the proper operation of the firm and having in mind all the time that "without knowledge and its associated assets of information and data, nothing can be done, nothing made, nothing achieved";

c) The transformation of the business enterprise into knowledge (and information) based economies offering new products and services and where knowledge considered as a central productive and strategic asset. The business pressed by a time based competition with a shorter product life, realized in a turbulent environment and having a limited employee knowledge base;

d) The emergence of digital firm – for convenience defined as "Organization where nearly all significant business processes and relationships with customers, suppliers, and employees are digitally enabled, and key corporate assets are managed through digital means (Laudon and laudon, 2007)". This becomes more emergent and possible from the advent and usage at a large scale of Cloud Computing.

Decision Making Cycle

The business decisions are critical source of value. But making the best decisions is well beyond the capacity of most business systems today, when decisions must be made faster, across more channels and product lines, leveraging more data, under greater regulatory demands and competitive pressures, and with more complicated constraints and trade-offs.

The ability to capture and exploit corporate knowledge has become critical for firms as they seek to adapt to changes in the business environment. Whether it been learning from past success or failures, identifying opportunities to improve custom profitability, or simply enabling teams to become more productive, knowledge management lies at the heart of any well-managed company. Again, in the new economy, the innovation, time to market, knowledge, and ability to execute differentiate the leaders from the followers. Having digitally enabled relationships with customers, suppliers and employees, core business processes accomplished via digital network, and rapid sensing and responding to environmental changes, the digital firm must offer strong support to decision process. If we consider that every decision-making cycle depends on finding answers to the following three core questions (for which the indicated capabilities used to find their answer):

- How are we doing? - the usage of scorecards and dashboards help monitor the business with metrics and find answers;

- Why? – the reporting and analysis provides the ability to look at historical data and understand trends (and even patterns, it means new knowledge) and to look at anomalies and understand why;
- What should we be doing? – a reliable view of the future and ways to fallow is obtained by planning and forecasting (and what if? analysis);

Then the integration of all these capabilities allows respond to changes happening in business and better adapt to business environment changes. IT can be to the company a catalyst for change and an engine driving rapid growth and insuring even business agility.

6.2. Factual and Procedural Knowledge

During the past decades (covered by the points 1 to 3, in §6.1) the traditional software tools, provides two ways for encoding and incorporating knowledge in application programs:

(1) as facts, represented generally as structured data stored in memory;
(2) as sequences of instructions in procedural database queries, web scripts or programming language.

For that reason, this type of representation of knowledge is denoted by the term of factual and procedural knowledge (figure 6.1). These two types are natural representations for the today's common computer architecture and represented in the way the computer engineered: a powerful central processing unit (CPU) that deals with arithmetic and logic operations (it executes instructions given as procedures) connected to a live memory (main memory) and to mass storage memory (facts recorded).

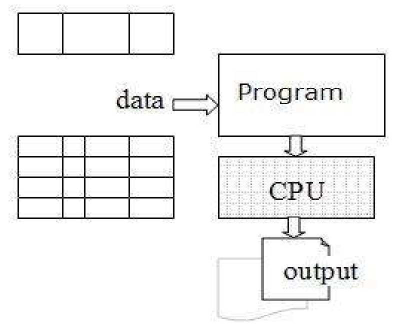

Figure 6.1 The representation of processing for Factual and procedural knowledge

The essence of the matter is that computers are devices that simulate the way that we handle things (Avram and Dodescu, 2003) and in that case they simulate the way we manipulate knowledge. The program in figure 6.1 can be, for a simple hourly salary computation (the payroll problem modeled in the web page content at http://www.avrams.ro/payroll-problem.html), the code listed in figure 6.2, where we have

```
<script type="text/javascript" language="javascript">
<!—
/* General Parameters */
  var startFreeTimeFriday=15;
  var salRaiseOverTime=1.5;
  var hourlySalary=18;
/* The rules to compute the HourlySalary */
  function Salary(day, workedHours, plannedHours)
  {
   if(day="Saturday" || day="Sunday" || (day="Friday" & workedHours > startFreeTimeFriday) ||
workedHours>plannedHours)
   { Salary=workedHours*hourlySalary+salRaiseOverTime*(
hourlySalary*(workedHours-plannedHours));
   }
   else
   {
   Salary=plannedHours*hourlySalary
   }
  }
  -->
  </script>
```

Figure 6.2 A simple JavaScript to compute the salary

some facts represented such as the number of times the salary raise when somebody works overtime (we refer by the name *salRaiseOverTime* in the algorithm) or the amount of salary for one hour (referenced by the *hourlySalary* name) and computational formulas to calculate the salary depending on the circumstances (normal salary when no overtime worked and a raised salary when works overtime or in the free time). Encoded in that way, as facts and/or as procedures, the knowledge representation is difficult to be identified and even to be re-comprehended even by its author, at a later time. This knowledge representation is too formal and too difficult for maintain and deduce in a natural way, and requires medium to high levels of knowledge about programming and the used programming language itself. In that case the incorporated knowledge is processed by the physical machine: the CPU executes the procedures having as input the recorded facts. When designed and realized applications incorporating factual and procedural knowledge this one is provided by policy experts from the business domain and modeled and formalized by IT experts. A solution to solve that problem for existing application software is given in (Avram, 2010/1) and (Avram, 2010/2).
The following paragraphs are excerpts from the source (Avram, 2010/2):

> “The increasing complexity of business relationships and global competition induces an increasingly complexity and dependence of organizations on Information Technology (IT). This complexity makes more and more important for organizations to use both business and IT best practices. Brand software, sold and delivered by leader IT companies, for business support such as Supply Chain Management (SCM), Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), Human Resources Information System (HRIS), Business Intelligence (BI) etc. incorporates good practices inspired from those of leader companies in various domains or, in other words, it incorporates the knowledge allowing the application and the operation of those practices. An organization adopting and using such software in its current operation, will start its activities and competitiveness at least from that point of good practices and associated knowledge.
>
> In this context, regarding the used software, the knowledge management will have two aspects:
>
> - Knowledge management, in the classic sense of knowledge about software product and its operation. This knowledge, acquired by training and practice, in time becomes also implicit for employees that uses the product;
> - Knowledge management, in the sense of knowledge incorporated in the software products that is, generally, knowledge about the way software processes the business itself (as its business processes). This knowledge is generally documented externally in the product manuals and is a key factor in deciding a software adoption.
>
> The knowledge incorporated in the software products and inherited by a company at software product adoption time is very difficult to be revealed to business people since, generally, this is included in the product documentation and revealed to the users when they are trained to use the software product. This knowledge becomes generally explicit/ implicit for business people. More, almost all of the time, even if undesirable, there are a lot of differences between the documentation content, how this is understood by business people and how it is really implemented into the product.
>
> The knowledge embedded in software is mobile, similarly to the tacit knowledge of employees: the withdrawal of a software product or non-usage of all his incorporated knowledge in current activities (as unused branches or options)

always leads to losing it, similarly to the way a company loses the tacit knowledge of a person who leaves it. The difference is that the unused options can be reconsidered and reintegrated in the company operation.

The incorporation in the company's knowledge base of the knowledge contained by new adopted software products creates a real base for company's model to migrate from a product-centered model to a knowledge-centered model.
Considering the way in which it incorporates and manipulates the modeled domain knowledge, the software will be classified in three broad categories of products:

- Business rules management systems (BRMS) that manipulate and execute business rules described, generally, as production rules. Production rules represented graphically and/or as declarative sentences are easy for humans to understand. Since each rule represents a small independent granule of knowledge it can be easily added or subtracted from a knowledge base. Because the rules are independent one from each other, they support the declarative style of programming which considerably reduces maintenance problems but makes algorithms extremely difficult to be represented and flow of control very hard to be supervised by a system designer. The rules formalism makes no allowance for uncertain knowledge. BRMS products can act any business rule described in terms of its rule description language and compliant with the constraints of the domain deserved by that BRMS.
- Externalized knowledge products that externalize the business rules and knowledge and that allow changing some aspects of the behavior without requiring rewrite the application. Only externalized business rules and knowledge, that represents the source of problem, will be rewritten. This architecture gives a high flexibility and agility (but not the highest, as given by using BRMS) to the application, allowing it to adapt its behavior to the organization's policies changes. This knowledge is described by a specific description language and understood and acted by an application that accepts a limited knowledge area.
- Monolithic products that incorporate business rules and knowledge together with the procedural part and the processing logic. They require rewriting the application if aspects of their behavior must change (triggered generally by business rules and knowledge changes). This architecture, even if it is very used in practice, is rigid to changes and requires rewriting the involved modules. More, they hide the knowledge in code and make it difficult to be used and even extracted by using different automated tools. The term monolithic is used here to outline that a software product (an entire application, a task or process, or even a service) incorporates all domain knowledge it models. (Avram, 2010/2)"

"Considering the previous definitions and the benefits and requirements for extensive knowledge management towards the organizations, the software architecture must change, doesn't matter which software category it fits. The change refers to adding an architectural component, in the form of one or multiple knowledge repositories (Figure 6.3 a), in order to systematically acquire, structure, store and maintain knowledge, formalized as business rules for all domain business rules that are incorporated in the software product itself."

For demonstration, of viability and applicability of that theory, the business rules are formalized by using the Web standard RDF, defined to model knowledge in Cyberspace.

"The formalization of the business rules is realized through RDF/XML serialized triples. The tool used to describe the RDF triples can be an RDF Parser that is able to convert the triples in the serialized format or an RDF Editor able to process directly the serialized format of the triples. This knowledge refers to the domain knowledge of the business domain modeled by the application software and together with the knowledge about the operation of the current application, it represents the knowledge of interest to the end users of the application.

For every company adopting software having this architecture, the companion repository act as a vector for transport and communication of the knowledge inherited from software, for that reason the software companion knowledge repository was named 'software inherited knowledge (SIK)' (Avram, 2009: 77). SIK must be accessible to all persons and/or applications having the rights to access and manipulate it. The knowledge involved here will be captured and maintained in the context of analysis, design, coding and testing phases (and of every reiteration of these required in implementation or current operation), and will be expressed in the form of business rules statements, as defined and understood by the software developer, since this incorporates the rules in the application body. (Avram, 2010/2)"

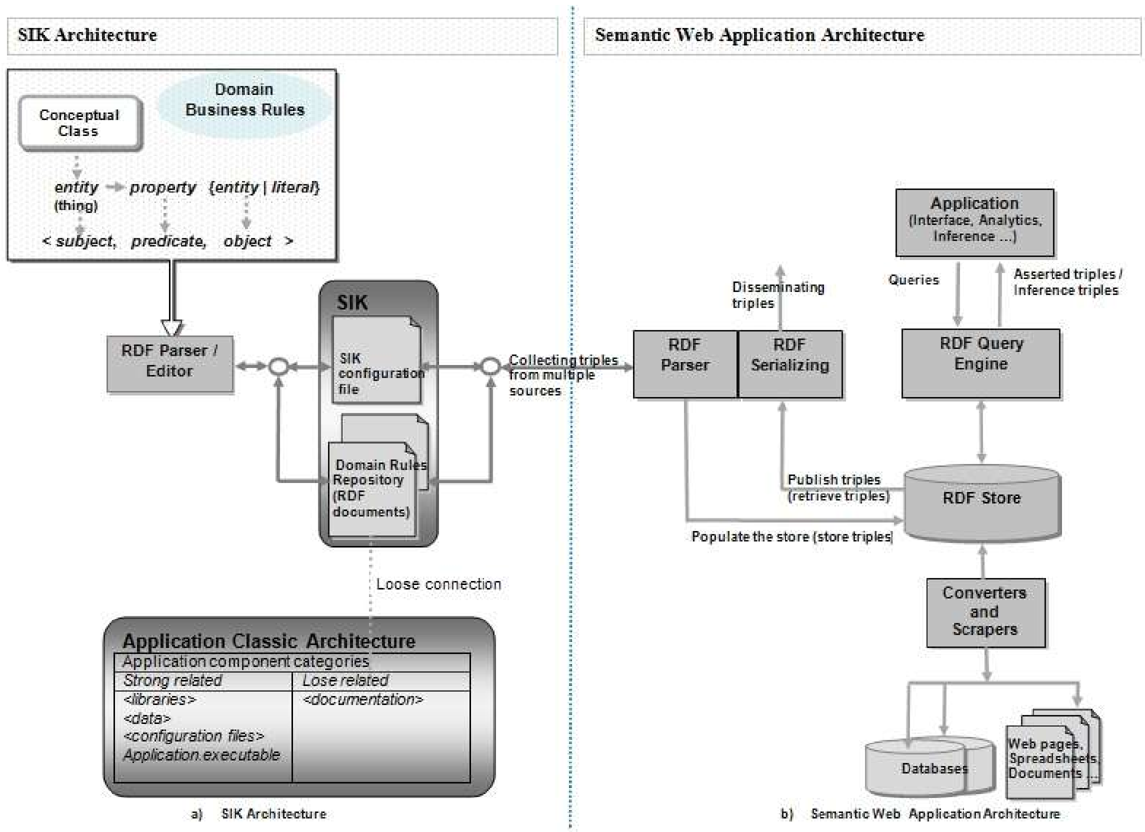

Figure 6.3 Making the software incorporated knowledge explicit and process-able (Avram, 2010/2)

6.3. Business Rules and Business Rules Management Systems (BRMS)

The Business Rules can be defined as the day-to-day politics that determines the interaction way with customers, suppliers, employees, and partners. They describe the

operations, definitions, and constraints that apply to a company in realizing and attaining its roles and goals.
The Business rules are represented by:

- **facts**: data (remember the data definition, "a set of unconnected facts") about the business environment (such as orders, invoices, information about consumers, information about employees etc);
- **simple instructions**: of the de forma <*if (condition) then actions*> in which *condition* is a Boolean expression, built using comparison operators and logical operators, evaluated by comparison with a set of facts and *actions* executed if the condition is satisfied, as in the following samples:
    - *If* the client is favorite (as Gold or Platinum to Vodafone, for example) *then* rise the given discount with 3%;
    - *If* the employee works during the current month overtime *then* pay him with 1 and half (1.5 times) the normal hourly salary for hours worked overtime.
- **actions**: simple returns or calls to direct actions (for example: If QuantityPurchased >= 50 Then apply a discount of 30% to the value). An action can be the source of generating a new fact and this is called inference (remember the processes realized in your mind to transform data into information: induction, deduction and inference processes as introduced in the parallel between the Human Mind and Computer that simulates the way we manipulate things).

The appliance domains of Business Rules include:

- Marketing strategies – the BI support added to CRM solutions for a better relationship with the customers. The marketing campaigns realized by these products on digital communication channels can be personalized and adapted to every individual customer as in a person-to-person strategy;
- Price politics, which can be directly applied when a customer places his orders (without a delegation given to the operator that prepare the invoice);
- Management practices used in the client relationship - it can manage, collect, and update information about customer on each online touch-point this one interact with the company's applications (via company's website/ portal, socialization networks accounts, e-mail messages, SMS's etc.);
- Human resources related activities (as the payroll problem given as example down here, of course the real one);
- Control forces, either to insure the conformity with the laws/ regulations, either to make decisions in accordance with laws/ regulations (and a rapid adaptation of the ways the application interact by the simple change of business rules accordingly);
- Products and services offers (again they can be personalized according to the customer preferences and behavior so that the customer will receive "his own message" (a message as written especially for him and only him);
- Insurance and healthcare – the insurance operator can define directly, for a personal insurance, the medical facts of the client and then obtain a healthcare risk measurement which can select/ dictate the conclusions from the insurance rules;
- Finance and risk analysis (it can help implement the checks of correlations between the debits and credits of accounts in the trial balance, can deduce the financial reports and even can analyze them and conclude, etc.);
- Operational management, can help in automating both, the decision making and actions (is an equivalent to Use-Case selected according to the combination of facts and rules);

- Call centers (the voice recognition of the customer and directing of interaction to his “own operator” if possible).

The business rules approach is applicable to both humans and IT systems. Business Rules Engine allows define the rules in a simple language so that the users are not constrained to understand or know a programming language.

The rules architecture continues the evolution of programming (figure 6.4):

- Initially programming languages such as C/C++ or Java;
- Internet adds HTML for logical display;
- Business Rules adds rules for policies description, appliance, and actions.

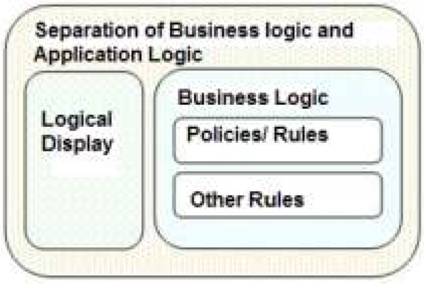

Figure 6.4 Rules Architecture
(Source: Oracle Business Rules)

The Rules Engines use the rules to analyze the facts (figure 6.5). After the facts analysis the Rules Engine can return results (represented by simple values or by complex data structures) or can invoke actions (here illustrated the evaluation of terms and finding final result of the computation formula).

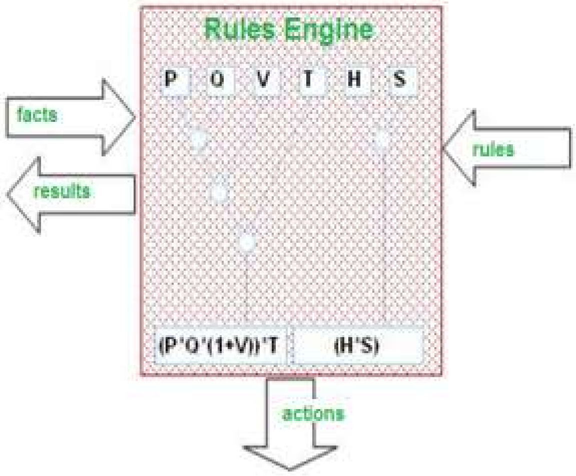

Figure 6.5 Rules Engine
(Source: Rule Enabling Applications with Oracle Business Rules, Oracle Whitepaper, September 2005)

The main offered advantages by Rules include:

Increased Agility: the business analysts can quickly develop and deploy the business rules which reflects the new business policies which applied and acted immediately.

High Transparency: the business analysts can review the rules and can easy deduce the policies they implement.

Low Costs: the business policies are implemented using authorized facilities that are extremely efficient and that are offered via an efficient interface for describing policies.

Reducing the dependence on IT: the business analysts can directly realize the implementation (of the new policies or those affected by the change of working rules) of the rules and that don’t necessitate (or necessitate a little) assistance from the part of IT.

Over past years the developments centres to two solutions designated to easy and faster adapt IT systems to changes: one based on business process management and another one based on business rules engines.

The business process management (BPM) solution is process-centric focusing on the workflow around a given process. The business process can be defined as “the unique ways in which organizations coordinate and organize work activities, information, and knowledge to produce product or service” (Laudon, Laudon, 2007). The business process management guides an application through a number of steps that must be followed to accomplish a specific task.

The business rules engines (BRE) solution guides application through rules that determines the BPM steps. Business rules are translations of the policies, the statement of guidelines governing business decisions, into detailed conditions and actions that unambiguously enforce the policy. In IT systems the business rules engines read business rules and then they do automatically what is necessary to do so that they still in concordance with them. The today dynamic business environment has a major impact to the business rule lifecycle. The business rule lifecycle is represented by the process by which a company manages changes to policies and their enforcement. Enterprises are using business rules engines as a means to reduce costs of managing changes to policies, and consequently, changes to business rules.

The two solutions, BPM and BRE, are integrated into a more complex one solution: business rules management systems (BRMS).

BRMS technology is a “horizontal” technology – it can be used in any business that has a specific problem: “How can I reduce the maintenance costs of IT applications that need change rapidly?”(Srinivas et al., 2007). With a BRMS (figure 6.6) the business logic is separated out and can be changed without impacting the remainder of the application. It adds decision capabilities to application and workflow-type processes.

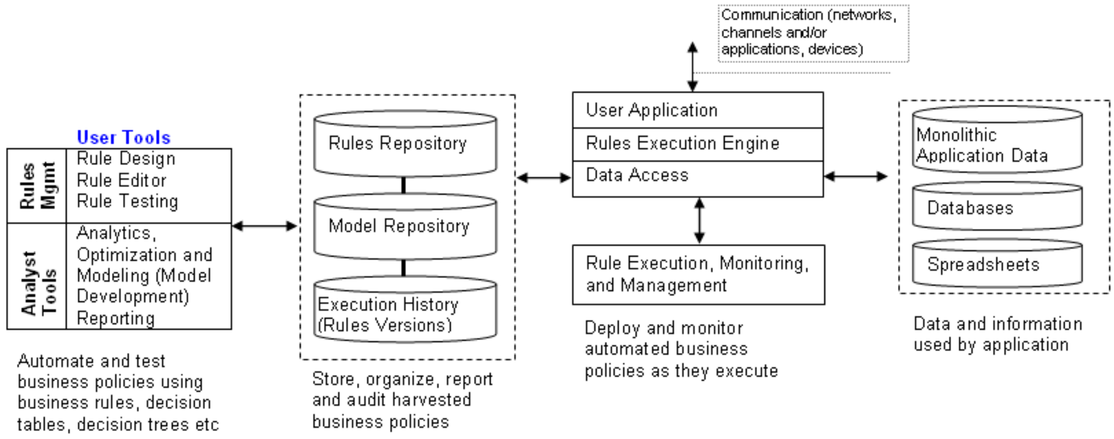

Figure 6.6 BRMS components

Within BRMS:

- Business policy experts can manage and evolve business policies using the methods and vocabulary with which they are most familiar;
- Technology experts can manage and evolve technology using methods and vocabulary most suitable to their tasks.

The information management solution for small and midsized companies is to fully integrate, share and deliver information so that they able to improve accuracy and speed of decision making by providing consistent, rapid and secure access to all relevant information. This type of information management implies the usage of business rules management systems. As a small to midsized business, budgets don’t allow for rip-and-replace approach and the software licensing can be prohibitive and restrictive, leaving few options for complete information management (Mooney, 2007).

Since both, the adoption of a standard language to define business rules and the adoption of a standard for Knowledge Interchange Format, appears to be to far in time the proposal for description of the SIK is to use RDF (Resource Description Framework) as shown in (Berners-Lee, Connolly and Swick, 1999) and (Daconta, Obrst and Smith, 2003). RDF has a simple data model, a formal semantic and provable inference. Because it uses an XML-based syntax it allows to anyone to make statements about any resource. This format allows for a better portability being adapted to Web and usable in all intranet, Internet, and extranet. In all these environments the application can be also delivered as a service and his SIK will allow for better communication of his capabilities.

6.4. Defining Controls by Rules

Decision making criteria are often written as rules. Rules described for management with business rules management systems (BRMS) and interpreted by rules engine generally do not imply any particular method for use: they express logical relationships between condition and conclusions and are intended to react dynamically to various inputs rather than executed in a predetermined manner. A rule engine offers to users the possibility to express the rules in a declarative manner. Rules are defined as logical assertions without procedural implication following an iterative process that determine and refines the outputs, inputs and rules. Domain experts can readily verify the rules as correct. A rule base is a collection of rules stating relationships between facts, values and the conditions. A rule engine dynamically chooses which rules to use on the values that are derived from the rules. In that way we can combine the usage of knowledge together with knowledge management. The term “knowledge” for the purpose of knowledge management is completely different from data or information. If we accept that data (what?) is a collection of non-redundant facts and that information (what?, who?, when?, way?, and where?) is data that have been processed so that they are meaningful then the knowledge is information associated with strategy and process (how?). Depending on who is the owner and how formalized is that knowledge we can categorize (CORTICON, 2007) that as:

1. *Implicit* (ill-defined or tacit) knowledge: as the personal knowledge of every employee. This is extremely valuable and most vulnerable (can be lost when employee “disappear” from the company);
2. *Explicit* knowledge: the implicit knowledge that is defined, explained or communicated to another person (does no matter if formal or informal).

The first two categories still owned by individuals so that they are vulnerable and difficult to maintain, share, deploy etc.

3. *Documented* knowledge: that is the knowledge included in the defined policies and procedures. This category allows knowledge to be organized, published, revised, and managed;
4. *Actionable* knowledge: this defined as a disciplined refinement and structuring of documented knowledge. This do not implies the application of judgment in its execution: the actions are taken confident to the statement alone.

This previous two types of knowledge are expensive and error-prone if enforced to be applied “manually”.

5. *Automated* knowledge: the actionable knowledge translated in code as processing.

If the knowledge or information can be easily viewed as a corporate asset in their own rights for data (or raw data) that is not possible even those data are the sources for information. For example, the generally accepted accounting principles do not recognize data as an asset in an organization’s financial statements unless it has been purchased. The value of data produced inside of an organization comes from how that is used. A high level of data make available can be represented by directly integration of databases (as data and data relationships and constraints repositories) together with reasoning engines as direct source for facts or for their rule base facts, as shown in figure 6.7. The benefits of that architecture are represented by the creation of a foundation for naturally describe and manage the rules together with the dynamically adaptation of automated processing to the changes in business and decision making described by that rules. This architecture allows after system’s users discover and document the knowledge to formalize, automate, deploy and share the knowledge.

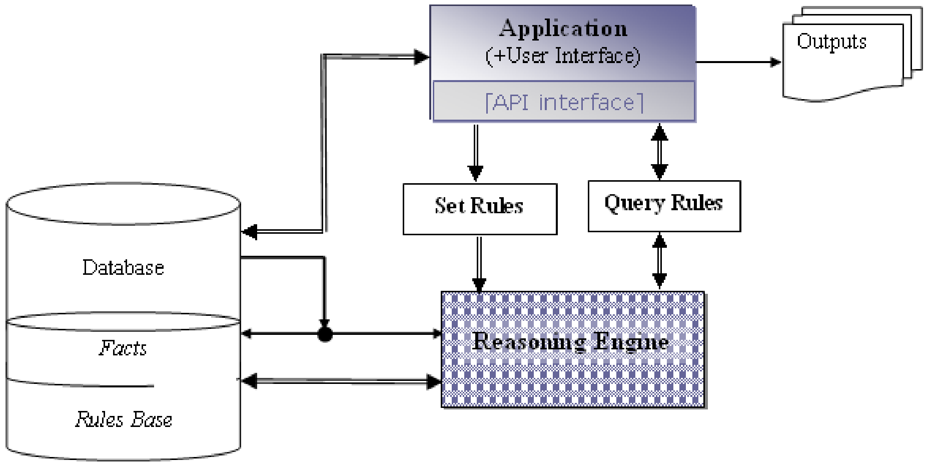

Figure 6.7 Integrating databases together with rules bases via reasoning engines

The rule-set (rule base) in the figure 6.8 uses three basic building blocks:
- Outputs: the result of querying the rules;
- Inputs: values that comes from the user, data tables and other sources;
- Rules: deduces values for outputs based on inputs.

6.5. Logical Knowledge

Starting with the point 4 from §6.1 (it also happens now) the software tools provides the representation of the knowledge as logical knowledge that means as relationships that mimics the human decision making. In a natural language, and more precisely on paper, the knowledge can be represented as rules, decision tables that supplies answers, graphs showing connections, formats specific to a given application area etc.

The relationships defining the knowledge are more complex that ones that can be represented in databases. In that case the knowledge is processed by a virtual machine running in a physical one. This is indicated in figure 6.8 by facts and rules (they form the knowledge base) and by the reasoning engine that uses the CPU to find an answer. This virtual machine can be identified in all of the tools available for business rule systems, expert systems, knowledge base systems, case-base systems and others (ARulesXL). This logical knowledge creates the premises to be modelled, handled and defined/used directly by the domain experts (and policy experts) without or little intervention in the processes of IT experts. The definition of knowledge is realized more and more closed to the natural language and/or with requiring little informatics knowledge from the part of domain experts (for example, general office support tools, like applications in Microsoft Office package).

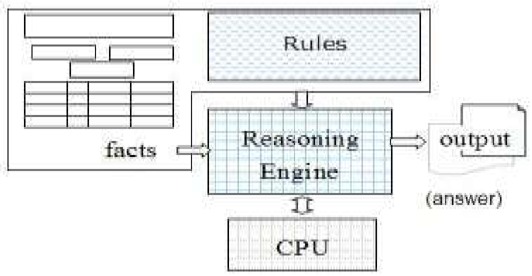

Figure 6.8 The representation of processing for Logical knowledge (ARulesXL)

The facts and rules outlined in figure 6.8 can be represented, for the same simple problem of salary computation, as in figure 6.9.

The rules here can be expressed directly in a natural language as follows:
- the salary is obtained by multiplying the worked hours with the tariff when is not weekend and no overtime worked;
- the salary is obtained by multiplying the allowed time with the tariff and by raising that with one and half time the salary computed for the difference between the worked time and the allowed time when is overtime or weekend;
- the weekend is when the day is “Saturday” or “Sunday” or is “Friday” after the beginning of weekend.

The facts and rules are interpreted together by the reasoning engine (figure 6.8) which determines for every cell in column Salary in the table the right salary computation rule depending on the facts recorded (day, allowed time, worked time) into the table and other general parameters (such as *salRaiseOverTime*).

The product that allows this type of definitions is called ARulesXL (http://www.arulesxl.com) realized by US Amzi Inc Company (http://www.amzi.org). The product allows combine the computational power of Excel with a rules engine and permit defining, collecting, and deploying the business rules inside the company. This task can be accomplished without requiring a high IT qualification for domain workers (is enough to know what is supposed they know: the office package, most precisely working with spreadsheets).

The facts as table values:

Day	Planned Hours	Worked Hours	Salary	
Monday	8	10		The column salary contains queries on the rule set SalaryRules with the =RQuery(...) defined into the Add In ARulesXL
Tuesday	8	10		
Wednesday	8	8		
Thursday	8	8		
Friday	5	8		
Saturday	0	8		
Sunday	0	6		
Total Salary				In these cell is a normal ormuls in Excel =Sum(...)

And the Business Rules:

SalaryRules

Salary = workedHours * hourlySalary WHEN (workedHours <= plannedHours AND NOT FreeTime)
Salary = plannedHours * hourlySalary + salRaiseOverTime * (hourlySalary * (workedHours - plannedHours)) when (workedHours > plannedHours or FreeTime)
FreeTime WHEN Day ="Saturday" OR Day = "Sunday"
FreeTime WHEN Day = "Friday" AND
workedHours > startFreeTimeFiday

Figure 6.9 The representation of processing for Logical knowledge [ARulesXL]

6.6. Introduction to ARulesXL

Each spreadsheet implementing a Business Rule solution uses three basic building blocks, as illustrated in figure 6.10:
- Outputs: the result of querying the rules;
- Inputs: values that comes from the user, data tables and other sources;
- Rules: deduces values for outputs based on inputs.

that acts under a <Inputs-Processing-Outputs> pattern.

It can contains other elements, such as any kind of elements accepted and that can be defined within a spreadsheet, that are not included in these three categories. The spreadsheet can include any legal Excel formula or user defined formula (or macros/ VBA code).

The first element that must be analysed represented by Outputs which permit to find and design, both, the Inputs and the Rules that must be applied to the Inputs to obtain that Outputs.

The Inputs for the Rule Engine are defined by using two functions provided by ARulesXL, **RArray()** and **RCell()** functions.
For **RArray()** you must define in the Cell the formula:
=RArray(*"name", range, ifRowHeader, ifColumnHeader, storeAsUnidimensionalVector*)
where:
- *" name"* is the name we give to the input array;

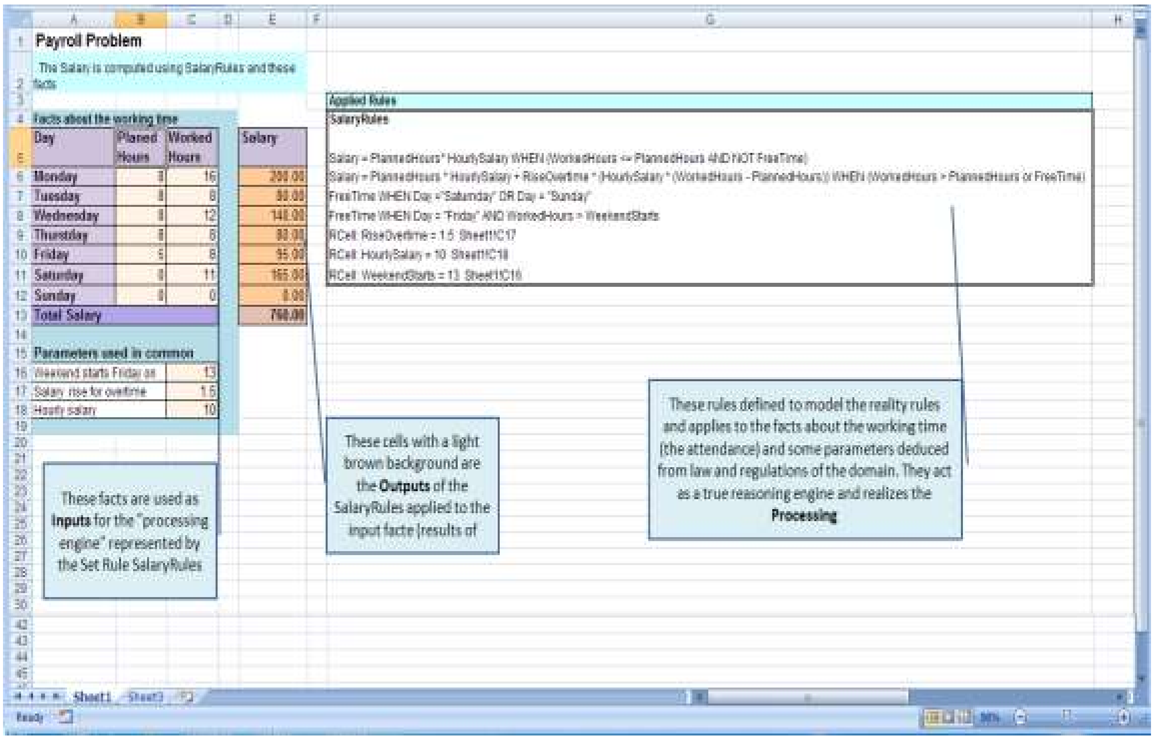

Figure 6.10 The Payroll Problem spreadsheet structure

- *range*, is the range of cells for the input. The input array must be a contiguous range of cells;
- *ifRowHeader,* is TRUE if we want include the headers for rows. The header can be referenced in the array index as input["*row header*"] for retrieval (as an index for the array). If the header not required use FALSE;
- *ifColumnHeader,* is TRUE if we want include the headers for columns. The header can be referenced in the array index as input["*column header*"] for retrieval (as an index for the array). If the header not required use FALSE;
- *storeAsUnidimensionalVector,* is TRUE if we want represent the multi-dimensional array (maybe) as a unidimensional vector. Otherwise use FALSE.

For **RCell()** you must define in the Cell the formula:
=RCell(*"variableName",cellAddress*)
where:

- “*variableName*”, is the name you want give to the value in the cell given by *cellAddress*.
- *cellAddress* is a reference (address) of a spreadsheet cell.

The figure 6.10 contains the following RCell() definitions:

```
SalaryRules

Salary = PlannedHours* HourlySalary WHEN (WorkedHours <= PlannedHours AND NOT FreeTime)
Salary = PlannedHours * HourlySalary + RiseOvertime * (HourlySalary * (WorkedHours -
PlannedHours)) WHEN (WorkedHours > PlannedHours or FreeTime)
FreeTime WHEN Day ="Saturnday" OR Day = "Sunday"
FreeTime WHEN Day = "Friday" AND WorkedHours > WeekendStarts
=RCell("RiseOvertime",  C17)
=RCell("HourlySalary", C18)
=RCell("WeekendStarts", C16)
```

Both input specification instructions must be defined inside the Rules Set, here called SalaryRules. In figure 6.10 the RCell() are evaluated and is written the name RCell fallowed by the name for the variable and the value assigned, and the source cell address.
For the Credit Approval problem the input specified by the instruction:
=RArray("Input", (A4:B16), TRUE, FALSE, TRUE)
where the range is for the facts as given in figure 6.11 and produce the Output in figure 6.12.

Credit Approval sheet for:

Applicant Name:	Vasile Avram
Variable Data for an Applicant	
Age	60
Employed	Permanent
Employer state	Active
Credit History	Pozitive
Credit Score	101
Loan Amount	70000
Mortgage Payment	1200
Property Price	140000
Monthly Income	8000
Down Payment	70000
Other Monthly Payments	700
Total Assets	100000
Total Liabilities	10000

Figure 6.11 The Input for Credit Approval

Status Report Based on the Input Data and the Rules	
Age Status:	OK
Employee Status:	OK
Credit Status:	OK
Monthly Nut Status:	OK
Down Payment Status:	OK
Status	Approved

Figure 6.12 The Output for Credit Approval

The Rule Set for the Credit Approval is defined as:

```
LoanRules
Status = "Approved" when CreditScoreStatus = "OK" and MonthlyNutStatus = "OK" and DownPaymentStatus = "OK"
Status = "Denied"
AgeStatus="OK" When input["Age"]>=18 and input["Age"]<=65
AgeStatus="To Young person!" When input["Age"]<18
AgeStatus="To Old Person!" When input["Age"]>65
EmployeeStatus="OK" When input ["Employed"]="Permanent" and input["Employer State"]="Active"
EmployeeStatus="Must Have a permanent job for such loan" When input ["Employed"]<>"Permanent"
EmployeeStatus="The Employer is to risky!" When input["Employer State"]<>"Active"
CreditScoreStatus = "OK" when input["Credit Score"] >=100
CreditScoreStatus = "Poor Credit Rating"  when input["Credit Score"] < 100
MonthlyNutStatus = "OK" when MonthlyPayment < 0.3 * input["Monthly Income"]
MonthlyPayment = input["Other Monthly Payments"] + input["Mortgage Payment"]
MonthlyNutStatus = "Monthly nut is too big"
DownPaymentStatus = "OK" when input["Down Payment"] >= 0.2 * input["Property Price"] and input["Down Payment"]
< 0.8 * NetAssets
DownPaymentStatus = "Insufficient percentage of purchase price" when input["Down Payment"] < 0.2 * input["Property
Price"]
DownPaymentStatus = "Too great a percentage of net assets" when input["Down Payment"] >= 0.8 * NetAssets
NetAssets = Input["Total Assets"] - Input["Total Liabilities"]
RArray: Input[?]  CreditAprovalSheet!A4:B16
```

Figure 6.13 The Rule Set for Credit Approval

The rule set is a named collection (the name is specified in the first cell/ line of the range for definition). The rules can be defined in any order because they are declarative. For easy management purpose is preferable that they defined grouped. The base expression for rules is

Fact = Value WHEN conditions

where:
- **Fact**, is the name of a fact for which we specify a new rule;
- **Value,** is an expression that can be a number, a text or a date to be assigned to Fact;
- **Condition,** is a Boolean expression that tell when the fact take the value specified by the rule.

In the Output part (figure 6.13) the cells for answers to the status contains queries defined as

=RQuery(ruleSetName, “find aFactName”)

where:

- *ruleSetName*, is the name of the Rule Set in which you search for output;

- “**find** *aFactName*” is a simple expression as defined here, that find in the rule set the fact according with the inputs and the evaluation of conditions attached to the rules. Generally is a search sentence of the form:

FIND Fact WHEN InputFact1=Value1 AND INputFact2=Value2 ...

The Output from foigure 6.11 contain the following qiueries:

Age Status:	OK	=RQuery(LoanRules, "find agestatus")
Employee Status:	OK	=RQuery(LoanRules, "find employeestatus")
Credit Status:	OK	=RQuery(LoanRules, "find creditstatus")
Monthly Nut Status:	OK	=RQuery(LoanRules, "find monthlynutstatus")
Down Payment Status:	OK	=RQuery(LoanRules, "find downpaymentstatus")
Status	Approved	=RQuery(LoanRules, "find status")

6.7. Usage of Business Rules Benefits

Benefits and advantages from using business rules based automated systems can be grouped in two benefits areas: business and technology benefits.

a) Business Benefits:

- Business Control – Business use control of the decision logic and decision process. Business rules can be modified by business users, in a controlled, auditable manner that is cohesive and consistent across applications. They offer closed-loop capabilities for designing, deploying and executing decision;
- Flexible and consistent workforce – Flexibility for rule-based decision process. All the company’s analysts can use the same methodology to build models for different application area;
- Learning environment – Business users and analysts can rapidly improve and evolve their decision logic by: a) a more quickly deploying updated models in an integrated environment, and b) learning at a faster pace through continual feedback loops and champion/challenger strategy testing;
- Increased consistency - By lowering the costs of making decisions through automation, by reducing the number of people, by streamlining the process needed to make or process a decision, by lowering costs of compliance, regulatory requirements, through centralized and easy-to-update business rules management.

By providing faster decision that operate at speed of transaction it lowers also hand-off costs between systems and between people;

- Increased Agility - Improved strategic alignment and greater competitiveness through faster response to market changes and regulatory changes. It ensure a greater return on new products and market opportunities, through faster time to implement and to change decision-based processes, and by being able to change approaches to the market more quickly. All that obtained together with lower IT development costs to change decisions, by placing control of business rules in the business user's hands.

b) Technical benefits:

- Real-time deployment – Decision intelligence placed into the transaction processing stream for real-time decision processing. Models can be deployed into production environment immediately after development;
- Development productivity – Decision logic is updated centrally and deployed automatically, regardless of underlying technology. Business rules are built and maintained once and are not replicated in fragmented systems.
- Ability to represent and simplify complex decision logic in standardized scores, decision tables or English-like business rules;
- Auditable decisions: the ability to understand why an automated decision was made that way it was.

The conclusions we want outline here refers to:

- by using rules we can eliminate the complex If instructions used in procedural programming to model decisions;
- we can define the rules by using a formal manner, with a very simple syntax, and closed to the natural language. This is very helpful to collect and define formally the business rules in a central rule base that can be used by all workers and applications in the company;
- by the simplicity of expression the automation effort is redirected from programmers to data-workers and domain-workers; it allows in that way to automate the decision at operational level;
- the application can be easy adapted to changes by reflecting that changes in the definition of business rules;
- the defined rules are "accessible" (they closed to the natural language and can be easy translated in this one).

References

[1] Albin, Stephen T. (2003), The Art of Software Architecture: Design Methods and Techniques. John Wiley & Sons. © 2003. Books24x7. <http://common.books24x7.com/book/id_6020/book.asp>

[2] Avram, Vasile (2010/2), The Acquisition and Sharing of Domain Knowledge Contained in Software with a Compliant SIK Architecture in:

a) International Journal of Engineering and Industrial Management, Number 2 (2010), ISSN 1647-578x, Lisboa, Portugal;
b) PROCEEDINGS OF THE 11TH EUROPEAN CONFERENCE ON KNOWLEDGE MANAGEMENT, VOLS 1 AND 2 Pages: 19-26, 2010, http://apps.webofknowledge.com/summary.do?SID=P2ngDG1okGoka39nnMC&product=WOS&qid=1&search_mode=GeneralSearch

[3] Avram, Vasile (2010/1), Benefits of Using Software with SIK Compliant Architecture In Cloud, Proceedings of the 5th International Conference on Knowledge Management: Projects, Systems and Technologies, Bucharest, November 12-13 2010 http://ideas.repec.org/p/rom/km2010/13.html, August 2010,

[4] Avram, Vasile (2009), Should Software Architectures Change To Adapt To The Knowledge Era*?,* Proceedings of the 4th International Conference on Knowledge Management: Projects, Systems and Technologies, Bucharest, November 6-7 2009 http://ideas.repec.org/p/rom/confkm/15.html, August 2009,

[5] Avram, Vasile, Dodescu, Gheorghe (2003), Informatics: Computer Hardware and Programming in Visual Basic, Editura Economica, pages 11-46

[6] Avram, Vasile (2008), The Informatics and Enterprise in Information Society. Virtual organization, 2008, http://www.avrams.ro/master/

[7] Avram, Vasile, Avram, Diana (2007/1), Managing knowledge within Small and Midsized Companies, rev. Informatică Economică, Nr. 44, ISSN: 1453-1305

[8] Avram, Vasile, Avram, Diana (2007/2), Utilizarea sistemelor bazate pe proceduri pentru descrierea şi automatizarea regulilor afacerii, in volume Calitate, Management, Integrare Europeana, Ed. Cartea Universitară, 2007, ISSN: 1582-2559

[9] Avram, Vasile (2007) - *Replacing Factual and Procedural Knowledge by Logical Knowledge in Application Software*, Review of Management and Economical Engineering Vol. 6, No. 5, ISSN: 1583-624X, 2007

[10] Babar, Muhammad Ali, Gorton, Ian (2007), Architecture Knowledge Management: Challenges, Approaches, and Tools, 29th International Conference on Software Engineering (ICSE'07 Companion), 0-7695-2892-9/07 IEEE Computer Society, 2007

[11] Baca, Claudia M. (2007), Project Management for Mere Mortals®: The Tools, Techniques, Teaming, and Politics of Project Management, Addison-Wesley Professional, 2007, http://acmsel.safaribooksonline.com/9780321423450

[12] Bass, Len, Clements, Paul, Kazman, Rick (2003), Software Architecture in Practice, Second Edition, Addison-Wesley Professional, 2003 <http://acmsel.safaribooksonline.com/0321154959>

[13] Berners-Lee, Tim, Connoly, Dan, Swick, Ralph R. (1999), Web Architecture: Describing and Exchanging Data, W3C Note 7 June 1999, www.w3c.org

[14] Daconta, Michael C., Obrst, Leo J., Smith, Kevin T. (2003), The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management. John Wiley & Sons. 2003. Books24x7. http://common.books24x7.com/book/id_6073/book.asp

[15] Greenman, Marla Beth (2006), The contribution of Knowledge Management Systems to Interorganizational Learning, SIGMIS-CPR'06, ACM 1-59593-349-2/06/0004, 2006

[16] Harvey, Michael (2005), Essential Business Process Modeling, O'Reilly, 2005
[17]. Laudon, Kenneth C., Laudon, Jane P. (2005), Essentials of Management Information Systems, Fifth Edition – Managing Digital Firm, Prentice Hall, page 2-69
[18] Mooney, Laura (2007), BPM-A Whole New World of Opportunity, Best Practice in Business Process Management, KMWorld January 2007, page 4-5
[19] Nonaka, I., Takeuchi, H. (1995), *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation,* Oxford: Oxford University Press.
[20] Srinivas, Reddy (2007), How to choose a BRMS for a Loan Origination System in 5 Simple Steps, YASU Technologies 2007, http://www.yasutech.com
[21] Song , Seokwoo, Nerur, Sridhar, Teng, James T.C. (2007), An Exploratory Study on the Roles of Network Structure and Knowledge Processing Orientation in Work Unit Knowledge Management - The DATABASE for Advances in Information Systems, Volume 38, Number 2, May 2007
[22] ARulesXL Manual and ARulesXL Runtime Manual, Amzi! Inc, http://www.arules.com
[23] BRG (2000), Defining Business Rules – What Are They Really ?, Business Rules Group, Final Report, revision 1.3, July 2000, from http://www.businessrulesgroup.org
[24] IBM (2007), Information Management Solutions for Midsized Business, http://www.ibm.com
[25] ILOG (2007), Business Rules Management System - ILOG JRules, http://www.ilog.com
[26] CORTICON (2007), The Knowledge Pyramid: A Framework for Understanding the Challenges and Opportunities of Managing Decisions as an Enterprise Asset - CORTICON Technologies, Inc., www.corticon.com
[27] http://www.forrester.com
[28] http://www.w3c.org – World Wide Consortium, Web Standards
[29] ORACLE (2005), Building Flexible Enterprise Processes Using Oracle Business Rules and BPEL process Manager, Oracle Corporation, Ian 2005, http://www.oracle.com
[30] http://www.objectconnections.com – Common Knowledge 3.0
[31] ORACLE 2007/1, Business Rules for Electronic Commerce, Project at IBM T.J. Watson Research, http://www.ibm.com
[32] ORACLE 2007/2, Oracle Business Rules, Bruce Lowenthal, http://www.oracle.com
[33] BRML, Business Rules Markup Language (BRML), http://xml.coverpages.org
[34] Business rules management systems, www.infoworld.com
[35] www.javarules.org
[36] www.essentialstrategies.com/publications/businessrules/