Research project 2 (SNE)

Preceding a digital forensic investigation as a service: *analysis of job scheduling principles and application of business rules*

UNIVERSITEIT VAN AMSTERDAM

Nederlands Forensisch Instituut
Ministerie van Veiligheid en Justitie

Course:	Research project 2 (Master System and Network Engineering)
Author:	Thomas J. SchermerVoest
Supervisor:	Dr. Worring (UvA)
Mentor:	Ruud van Baar (NFI)
Date:	11-08-2012
Version:	1.2

Table of Contents

1. Introduction

Since the first personal computer was sold, there has been an ever growing demand for technological advances towards more efficient and capable digital solutions. Access to the World Wide Web and availability of mobile phones has become the standard for most consumers around the globe. This has resulted in an exponential increase in computer processing power and storage capacity.

Over the past few decades, digital forensic applications have been able to handle ever growing volume of data. However, prognoses[1] have suggested that ordinary applications used for processing, analysis and reporting digital information will no longer be able to keep up with larger and more complex data. As figure 1 demonstrates, the former way of preceding a digital investigation could be time-consuming.

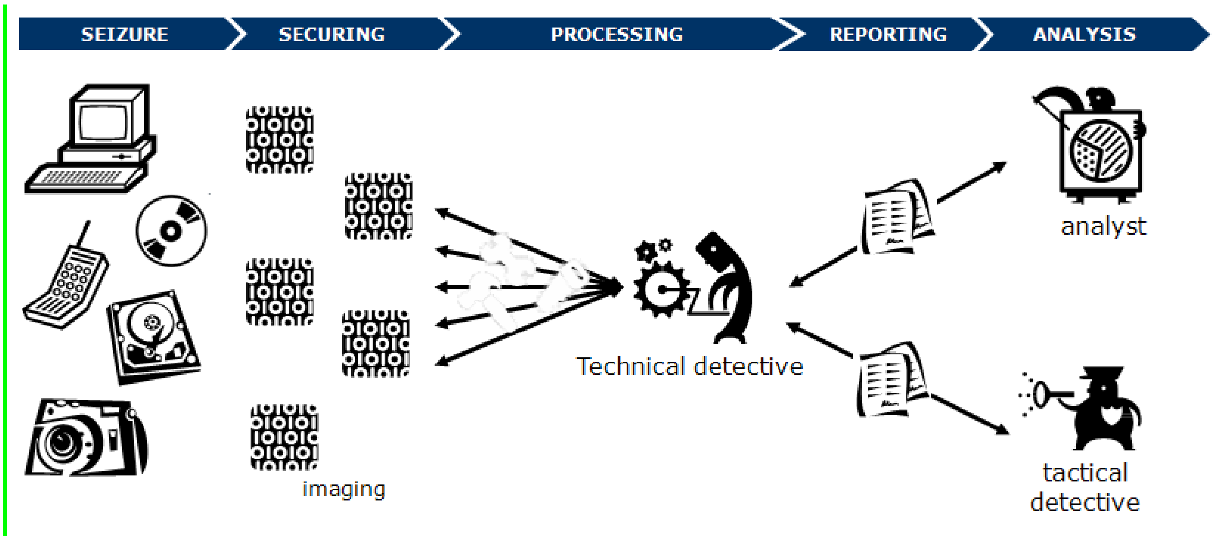


Figure 1. [2]The traditional way of preceding a digital investigation often took considerable time and effort as all communication paths ran via the technical detective. If the tactical detective or analyst wanted to be provided with new information, they sent a request to the technical staff.

In the Netherlands alone, calculations have indicated that by 2014, a daily amount of 110 TB (terabyte) will have to be processed. According to Huebner et al[3], law enforcement and investigation agencies worldwide are eager to have an efficient and future-proof application that can deliver information on demand without having to index and manipulate data at the moment they are queried.

In the push towards more efficient ways of analyzing complex digital forensic data, several products have been launched in the forensic community that may resolve the time-consuming problems. For instance, Accessdata[4] and Zylab[5] offer products that provide a distributed indexing framework for the analysis of large volumes of data.

The digital forensic community, including the Netherlands Forensic Institute (NFI), are in need of a system that allows the (distributed) indexing of images to build a trace index that gives investigators, analysts, detectives and technicians secure (on demand) access to case related information using a graphical user interface (e.g. a browser). Figure 2 displays how new forensic applications, widely accessible to authorized users could work as a service.

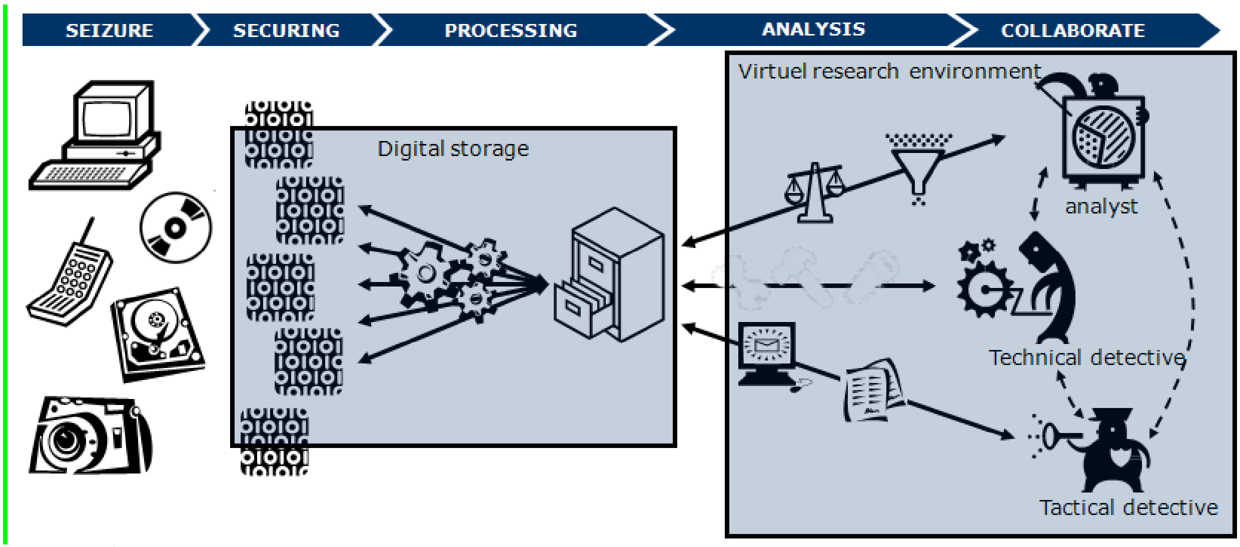


Figure 2.[1] *Compared to the traditional way of preceding a digital investigation, this method gives the analyst and tactical detective the opportunity to directly consult the trace database, without interference of the technical detective.*

A suitable solution to preceding a digital forensic investigation as a service has partially been given by the NFI with a product called XIRAF. This system is fed with captured forensic data before a researcher commences with a case. It subsequently extracts all relevant data, converting it into useful information. After the system has completed the indexing process, using a wide range of tools, each specialized for a targeted purpose, the researcher is able to query a constructed Oracle database using a web browser, which will present data on demand without the interference of intermediaries (see Figure 3).

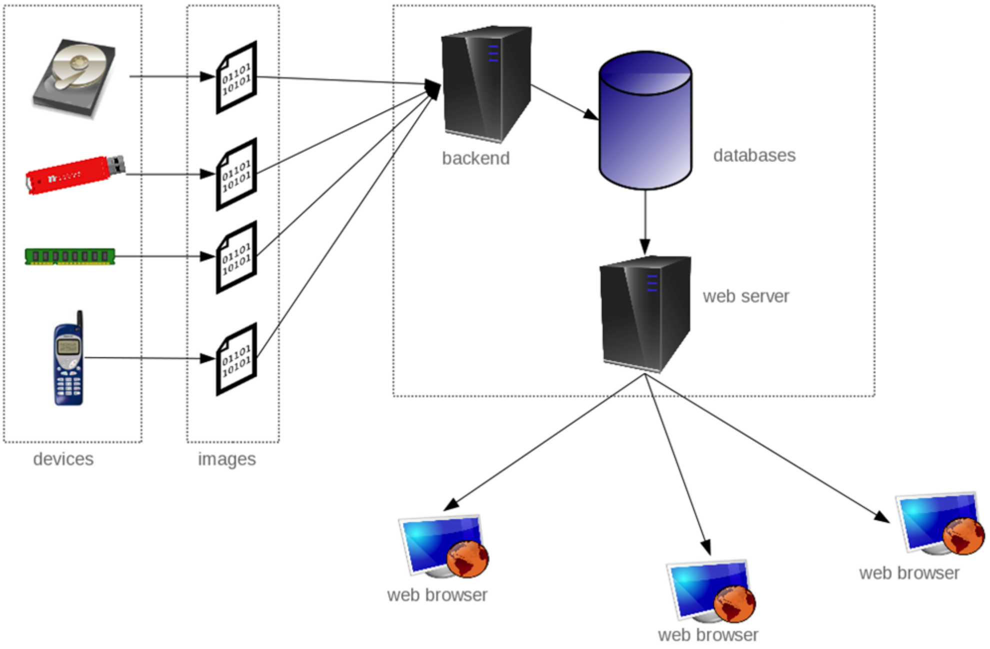


Figure 3. [6]*Global functioning of the XIRAF system.*

Due to the recent success of having a digital forensic framework as XIRAF, the NFI has expressed the need for a successor with even greater capabilities and scalability, namely Hansken. This system is meant to be implemented nationwide in all police agencies enabling them to upload and query digital case data. It could provide a solution to time-consuming and lengthy cases that were hard to analyze using traditional and commercial software that are less capable of handling large and complex data sets.

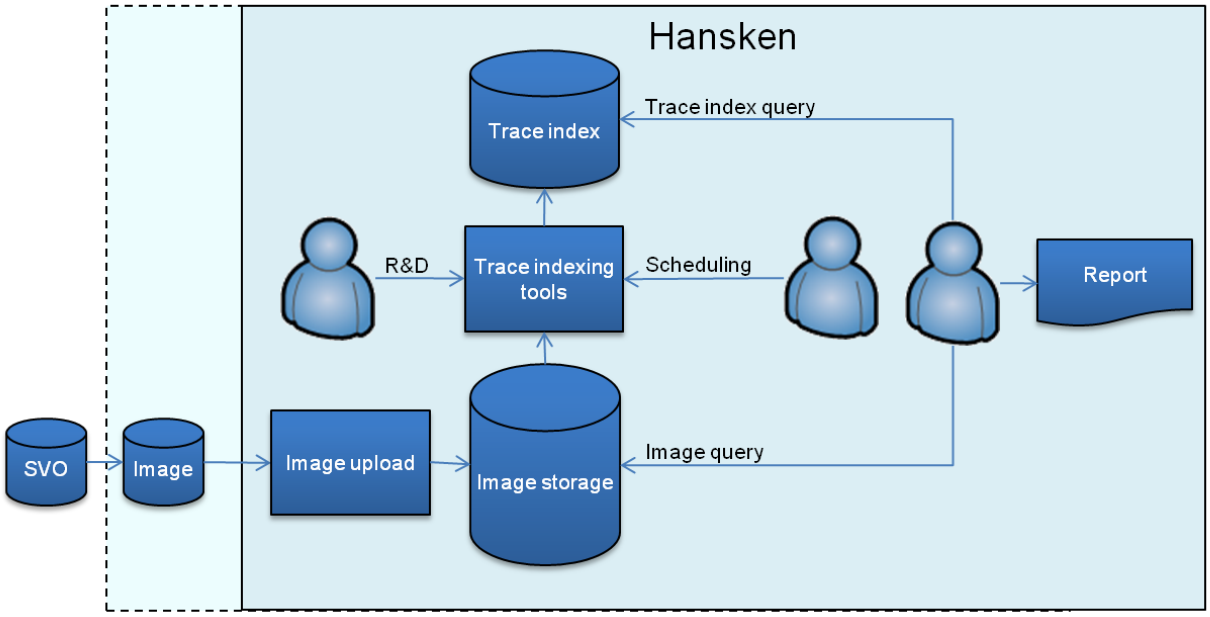


Figure 4.[7]The Hansken system

The workflow involves the acquisition of seized materials (e.g. discs, phones, USB sticks), which are referred to as SVO (piece of conviction). An exact copy of the SVO is made, which is called the image. The image is uploaded into the Hansken environment and stored for further processing. The type of image and the type of investigation determine the set of trace indexing tools and the order in which the tools are scheduled to process the image and create a trace index. Due to constant changes in image content, a researcher is required to conduct additional research & development to create the required trace index. Both the trace index and the image can be queried to collect the evidence required and create the report that is needed by the requesting party.[3]

It can be said that the XIRAF and yet-to-be constructed Hansken system provide an efficient and unique approach towards creating a digital forensic application as a service. The construction of Hansken will constitute a leap forward for the digital forensic world in addressing problems encountered in the process of preceding an investigation. However, a distributed system that enables users to upload cases in the form of images requires scheduling principles, e.g. for prioritizing images. Solving the need for **job scheduling principles** can be found in the application of **business rules**. The literature provides no evidence that any attempts have been made to couple a scalable forensic application to business rules.

Job scheduling needs to be automated using a business rule engine. A job scheduler could manage overall insight into the job processing progress and take action when events occur, such as prioritizing certain jobs according to business rules. Business rules are dynamic rules independent of an IT system that can also be modified by non-IT personnel.

It is still uncertain what (types of) business rules should be defined for a scalable forensic application and what type of business rule management system (BRMS) is adequate. Job scheduling principles have to be investigated. This research will try to define the most important business rules for a forensic job scheduling system as well as define a set of requirements that must be met by the specifications of the BRMS in order to work towards a suitable forensic indexing framework.

It will include an elaboration on business rules and methodologies to acquire, define, express, model and manage them. In order to properly capture business rules, extracted from business processes, it is necessary to use a preferably standardized method suitable to the project. Finally, the research should provide answers to the following questions:

- **What are the business rule requirements for a digital forensic application as a service?**
    - According to the 'as a service' design of a forensic application, NFI user feedback, input and literature research, several requirements in the form of business rule statements have been defined.
- **What business rules should be implemented for a digital forensic application as a service?**
    - This section will provide an answer to which business rules are essential to a distributed and scalable forensic job scheduling system. By defining, expressing and modeling rules, proper guidance is provided regarding functions that need implementing in a forensically suitable job scheduler.
- **What requirements principles for the BRMS should be met?**
    - A variety of several different requirement principles is motivated for the application of a BRMS for a forensic application as a service, such as Hansken.

Section 2 will consist of a detailed literature study on the application of business rules for a forensic job scheduling system. Several methodologies will be given concerning the structuring and management of business rules (**section 2.1 -- 2.4**). After that a selection of methods will be chosen to work with (**section 2.5**) and results produced in the form of business rules (**section 3**). In addition, requirements will be specified that will enable the NFI to choose an adequate business rule management system for a job scheduling system that can be used in Hansken (**section 4**). In the evaluation (**section 5**), the results as well as the methodologies used will be discussed. Finally, **section 6** will provide a conclusion and several recommendations for future work.

2. Literature study; business rule methodologies

This section will elaborate on several scientific methodologies found in the literature which are essential to this research. These include business rule **definition** (2.1)**, expression** (2.2)**, modeling** (2.3) **and management** (2.4). Initially, rules can be put into different categories and therefore need definition standards for proper structuring. Secondly, rules can be expressed in various forms from natural language to vendor specific implementation standards, which will be explained. Rules can also be expressed using models that can graphically represent them for better understanding. Finally, a single business rule management method can encompass the definition, expression and modeling of business rules. Therefore, several comprehensive business rule management methods are analyzed that allow the entire process of definition, expression and modeling to be properly guided.

Chapter 2.5 will present a motivation for selecting the most suitable (set of) methods concerning business rule definition, expression, modeling and management. All methods are judged and validated on the criteria that can be categorized in general criteria and project specific criteria:

- ✓ **General criteria: must** be a uniform, standardized and preferably widely supported method
  - Methodologies have to be reliable and proven to work in this research; proven to have worked in similar projects
  - Should be well-documented and evaluated by the scientific literature
  - Understandable for laymen as well as technical skilled readers (e.g. developers as well as business analysts)
- ✓ **Project specific criteria: must** be suitable for this specific project
  - Provide a quick start-up phase due to limited project duration. Focus should mainly be on the efficient gathering and structuring of business rules
  - Should be suitable for working with business rules (e.g. modeling methodologies should have extensions to work with business rules)
  - Provide an evolutionary approach, e.g. step by step method beginning with plain text rules to finally reach architectural rules definition
  - Should be usable without the need for specific software architecture related information, which is unavailable at the moment of this research

2.1. Business rule definition

In knowledge-based systems, human reasoning and knowledge can be captured and used in a set of rules. These rules are often defined as declarative languages and stored in a rules database. They are controlled and processed by a special component, called an interference engine. This engine checks the conditions of the rules at any given time and determines if one is to be fired or not.

In the literature, multiple definitions for business rules have been used by different methodologists. Rosca et al define business rules as "*statements of goals, policies, or constraints on an enterprise way of doing business*".[8] Herbst describes business rules as "*statements about how the business is done, i.e. about guidelines and restrictions with respect to states and processes in an organization*".[9] Some scientists share a different point of view; Krammer has a more information-system based definition that states that business rules are characterized as "Projections of external constraints on an organization's way of working, and on its supporting information systems 'functionality'". [10]

The GUIDE Project method

The GUIDE (an IBM-oriented industry group) model in figure 5 clearly describes the business rule model, which provides a deeper understanding of business rule concepts.

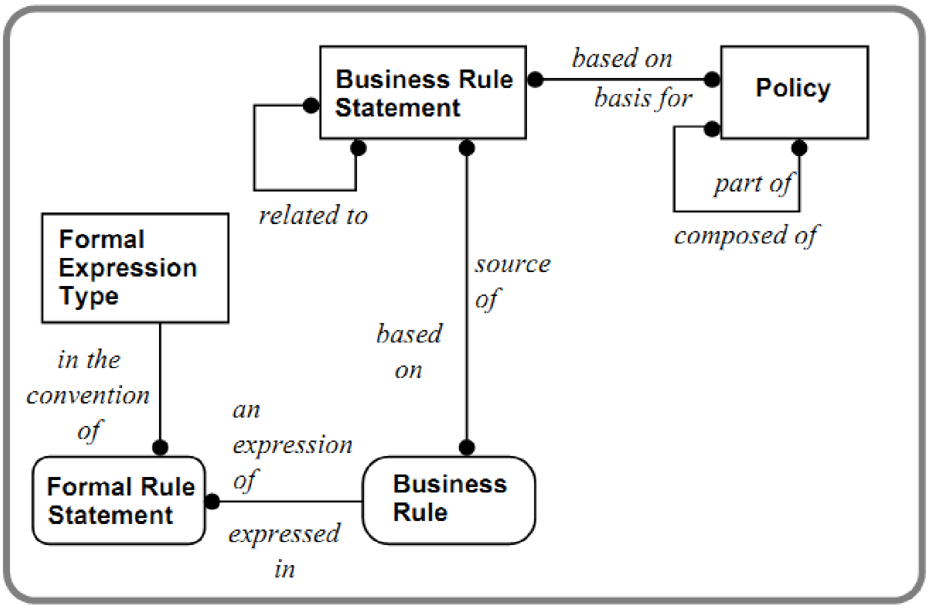


Figure 5. The business rule model

The basis for business rule statements is the business policy. Rules can be typed as the general statement for the direction of an enterprise. The business rule statement can be seen as a declarative statement or structure or constraint which is placed upon a business. GUIDE states business rule as *"a statement that defines or constrains some aspect of the business"*.[11] It can be applied to people, processes, corporate behavior and IT systems with as final goal aiding the organization in achieving its goals.

For example, a **business** rule might be:

> "A car with accumulated mileage greater than 5000 since its last service must be scheduled for service."

Business rules express policies within an organization using a formalized vocabulary of 'if-then' statements. A business rule is therefore converted into a **formal** rule statement:

> *If Car.miles-current-period > 5000 then invoke Schedule-service (Car.id) End if*

According to IBM's GUIDE method, business rules are classified as followed:

1. **Structural assertion**

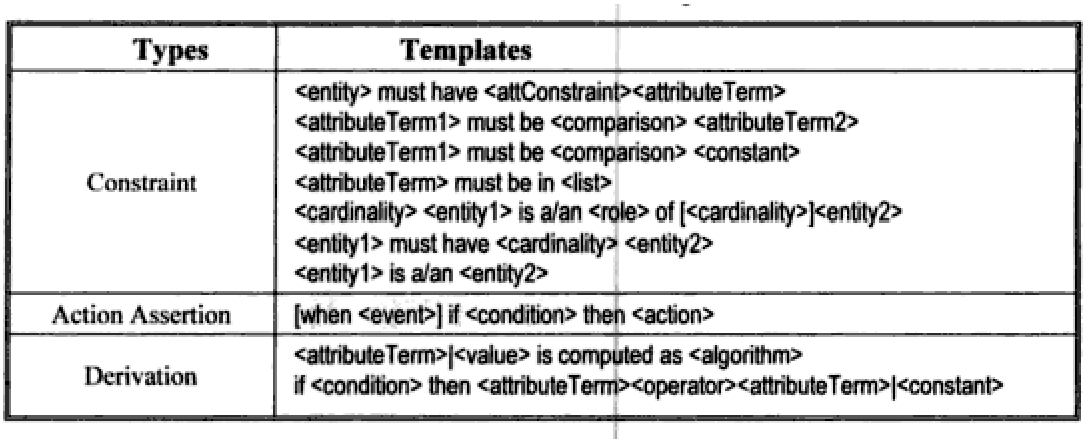
   The manner of business operation can be described in terms of the facts that relate terms to each other. For example, a user can schedule a case for indexing is a business rule. These facts can be documented in natural language, relationships, attributes or graphical models.

2. **Action assertion**

   In order to constrain enterprise behavior, constraints are defined in order to prevent certain actions. For example, 'only level 5 users are allowed to schedule jobs' or 'every project must have a minimum of one project manager'.

3. **Derivations**

   Derivations can express conditions that result in conclusions. Such rules define the validity of facts and can be used to infer new facts based on known facts. For example, owners of a gold membership receive a 10% discount. Peter is a gold-member. Therefore he receives a 10 percent discount. [12]



Types	Templates
Constraint	<entity> must have <attConstraint><attributeTerm> <attributeTerm1> must be <comparison> <attributeTerm2> <attributeTerm1> must be <comparison> <constant> <attributeTerm> must be in <list> <cardinality> <entity1> is a/an <role> of [<cardinality>]<entity2> <entity1> must have <cardinality> <entity2> <entity1> is a/an <entity2>
Action Assertion	[when <event>] if <condition> then <action>
Derivation	<attributeTerm>|<value> is computed as <algorithm> if <condition> then <attributeTerm><operator><attributeTerm>|<constant>

Figure 6. The GUIDE template for different types of business rules.

The business rule is created from various components inside a formal business vocabulary. This vocabulary defines business terms, operators and values that are of importance within a business application. They are composed of:

1. Business terms, objects that are part of, or affected by business processes. For instance, customer, producer or administrator.
2. Operators, to compare the properties or characteristics of different business terms. For instance, arithmetic operators.
3. Values, such as numbers, plain text or other defined business terms.

The Ross method

Next to the traditional GUIDE method, the Ross method provides a more extensive framework to define and classify business rules. The method was initially designed by Ronald Ross and proposes three business rules types: terms, facts and rules. [13] It is based on the method offered by the GUIDE structure, but includes several additions.

Ross stated that rules can either be atomic or derivative. Every derivative rules consists of two or more atomic or derivative rules. Due to the fact that it consists of 32 atomic and 58 derivative rule types, the Ross method is clearly the most detailed rule classification scheme that has been developed.

According to Ross, the five methods of rule modeling are as follows: [11]

1. Determining whether a business rule is a condition or an integrity constraint
2. Establishing the anchor of the rule
3. Establishing the correspondent (-s) of the rule
4. Determining the type of the rule
5. Establishing an association between rule anchor and correspondent (-s). Rule anchor and correspondent (-s) can also be constants, other rules or yield values of other rules.

2.2. Business rule expressions

In order to accurately express business rules, a uniform approach is needed. Most literature advises that business rules be formulated in natural language. However, this would prove less functional for the application of formal methods and tools that use automatic reasoning. Still, it is generally the most efficient and appropriate representation for business rules that are related to organization policies. For the simple reason that business rules are initially formulated and analyzed by business-oriented people that generally lack computer science expertise.

Once natural language (intentional) rules are transformed into operational rules, there is a wide variety of language formats to choose from. Rules may vary between research prototypes (N3), **vendor-specific formats** (Drools (**DRL**), Fair Isaac's **SRL** or ILOG's **IRL**) and the ones used for the XML-based exchange of business rules (SRML, PRR and SBVR). Each type of rule language has its own philosophy and therefore its own field of expertise. For example, rules used for reasoning applications require a different representation from rules used for productions goals. Another difference is the domain of use. Rules used in semantic web practices often focus on generating new rules from a set of already known facts, while rules optimized for business operations and human behavior often distinguish between permissions, constraints and desires.[14]

However, rules need to address business concepts as well. These can be clients, providers or any other typical user. (This is where UML use cases, flow models and possibly (for IS architecture rules) class diagrams are valuable assets.) By doing so, the relationship between business rules, users and final implementation become clear.[15] In this document, rules are expressed in a unified form, similar to the Drools Rule Language. Furthermore, unified modeling language (UML) is used to present use cases and flow diagrams. The rule language to be used for the job scheduler will likely be vendor-specific, such as DRL or IRL.[16]

Rule Interchange Format (RIF)

Since it is uncertain what type of BRMS is to be used for rule management, the language to express rules is therefore still uncertain. In a wide variety of (vendor specific) rule languages, the business rule community made several efforts to define standard rule languages. Some implementations are: Rule Interchange Format (RIF), RuleML and SWRL (Semantic Web Rule Language). These were specifically developed in order to become a standard for exchanging rules among disparate systems and achieving interoperability. However, none of these languages have been implemented in commercial BRMS products so far.

K-Sites Rules Tool

In recent years, a tool named K-Site Rules has emerged that supports the harvesting, definition and implementation of business rules. More importantly this tool helps the standardization of business rules by providing automatic translation among SWRL and languages interpreted by commercial products. [17] Especially its translation capabilities could prove beneficial, should a uniform language be preferred above a vendor-specific language.

2.3. Business rule models

Business rules have proven their importance for organizations as they describe how business is done. They have the ability to make applications flexible and open to change.[18] Business rules originated from the artificial intelligence community, where they functioned as a way of representing knowledge. While working with knowledge-based systems, the knowledge and reasoning of human experts could be captured and stored in a repository of a complex network of rules.[19] This repository was often referred to as an interference engine. The system continuously evaluated rule conditions and at any point in time decided if a rule was allowed to interfere.

Once the business and IT communities were convinced of its potential, business rules in database systems started to emerge. From here on, valuable research was conducted to find a way to represent business rules in data models.[20] Dynamic business rules could not be expressed in Entity-Relationship-Models (ERM), because they do not allow for explicit representations of events, conditions and actions.[21] Therefore multiple extensions to ERM have been proposed, that would allow modeling of business rules, such as ELH and SSADM.[2223] Business rules have also been expressed in terms of business concepts and corporate knowledge that are captured in a more general conceptual modeling architecture, such as BRADES (see section 2.4).

As the popularity of integrating business rules in information architectures grew, more effort was put into having means to accurately model them.[24] The object-oriented community recognizes the fact that business rules deserve attention, but to date no agreement has been reached on where to put them in object oriented models.[25] Part of the community believes that objects are responsible for their own behavior and that business rules therefore should be modeled in object and class models as class properties. The Unified Modeling Language offers a broad perspective in modeling business rules, but does not always provide the guidance needed to model rules properly.[26] Another method for defining business rules can be sought in the OCL, which stands for Object Constraint Language, and is part of the UML standard. Although this language presents detailed methods for setting constraints on rules, it does not prove that well in system requirements when working with business people. [27]

2.4. Business rule management

BRADES

In comparison to other methodologies, BRADES covers the entire lifecycle of business rules. Furthermore, it describes a proper way for the acquisition, deployment and evolution of rules.

The **acquisition** phase describes the organization's objectives, goal-oriented rules and constraints. The raising of issues in the initial phase helps to uncover solutions to progress towards the next phase: generating operational business rules. Furthermore, this phase tries to capture the entire process towards initial rules, starting by defining high level enterprise rules.

The **deployment** phase accurately separates the deterministic rules from the non-deterministic rules. By doing so deterministic rules are capable of characterizing situations in which they can be applied without further need for decision making. The non-deterministic rules require human interaction to assess whether they are conflicting or ambiguous. The BRADES methodology helps to solve issues with these types of rules by providing definitional rules that aid in solidifying the terms with values available at the operation time.

Based on monitored data provided from the deployment phase, one is able to extract relevant information for the start-up of the next phase: the **evolutionary** phase. In this phase, the discovery is often made that some rules are flawed or incomplete and need correction. A decision model is used to assess what specific part of the rule is flawed, such as rule criteria, arguments, alternatives or assumptions. Furthermore, changes in the organization's internal or external sources might require changes in one or more business rules.

In essence, the BRADES methodology is equipped to assist from the very beginning of the rule discovery approach towards the systematic collection of rules and final implementation. [13]

PROTEUS

PROTEUS is yet another methodology that provides a chronological approach to defining, capturing, expressing and organizing business rules. Its main advantage is that it provides a guide to facilitate the rule requirement analysis. In its turn, this helps to build a user involved business model that allows the harvesting of business rules from the products delivered with the business model.

As in most common business approach methodologies, this method defines rules as a (formal) business expression about a specific theme. According to PROTEUS, rules are typically classified in **business** categories (the main function of the rule in business execution), **functional** categories (its operation of effect), **abstraction** categories (defines the strictness of the rule) and finally the system category (handling of the resulting actions, after a rule is initiated).[15]

Manchester Business Rule Management

One approach that is particularly suitable and interesting to this research is the Manchester Business Rule Management (MBRM). It comprises numerous techniques for the elicitation, organization and management of business rules. The approach has its origin in Enterprise Knowledge Development (EKD), a development framework that is also used in large-scale industrial applications such as banking, electricity deregulation and e-business.[28][29]

The MBRM approach consists of several key information system development stages, all centered on a business rules paradigm. An example of the scientific framework is displayed in figure7.

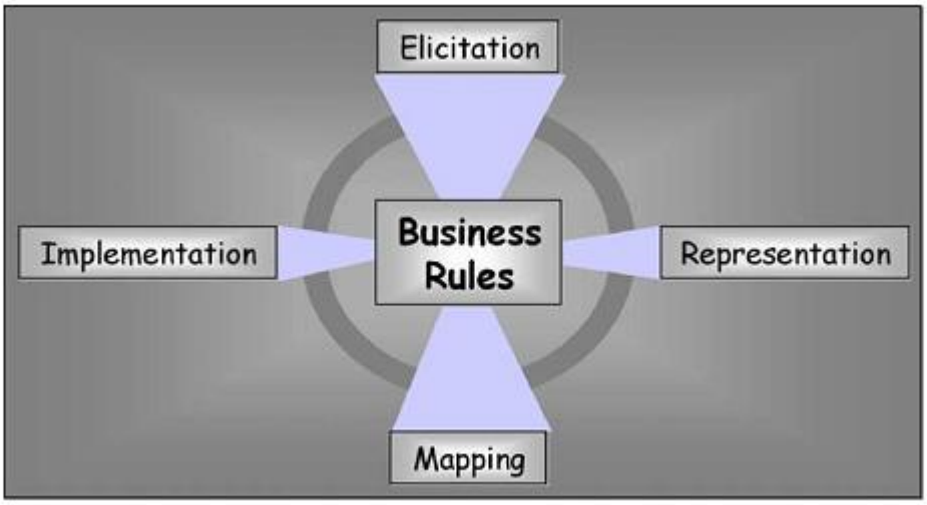


Figure 7. The MBRM framework[30]

All activities displayed in figure 7 are vital to the correct functioning of the entire framework. First of all, the **elicitation** process is concerned with the identification of stakeholders, the domain ontology and the rules that govern the behavior of the business application. [31]

In its turn, the **representation** handles the way that business rules are specified according to their typology. Thirdly, the **mapping** process is concerned with linking business rule specifications to equivalent software design structures. Finally, the **implementation** deals with the way that software designs are realized in software code and database structures.

The MBRM methodological framework constitutes three views for approaching information systems analysis, which are: [19]

- **The intentional view**
  - Holds information concerning preparatory activities that aim at initial understanding of the organization and crystallization of the project scope and objectives.
- **The operational view**
  - Incorporates the deployment of enterprise knowledge context, thereby referencing to business process concepts, such as actors, activities and information objects used or produced by enterprise activities.

- **The information systems view**
  - This step correlates between the initial analysis of the system and the proposed design. It transforms the implementation free requirements to the implementation specific requirements and specifications. A method for this transition could be the creation of class diagrams on the basis of earlier contextual work.

In each subsequent step of this methodology, business rules are treated differently. Therefore a distinction can be made between intentional, operational and architecture rules.

- **Intentional rules** are typically seen from a business view, and are often expressed in forms of natural language. They express principles, practices and specify the way an organization does business. They relate to current and future goals set for the organization, information about rule collection and details about their enforcement. In order to classify intentional rules, the following matters have to be identified:
  - Identify key users / actors within the application
  - Specify project goals and boundaries
  - Determine stakeholder viewpoints
    - In order to separate different stakeholder concerns

- **Operational rules** are rules set from a business process perspective. Their function is to describe the actions that should be taken once a rule is triggered based on the occurrence of a business event. They are derived from the translation of intentional rules to formal rule statements, often described by a suitable rule language (see business rule expression section). It is common for operation rules to make reference to other knowledge concepts, such as actors, activities, information objects and their attributes.

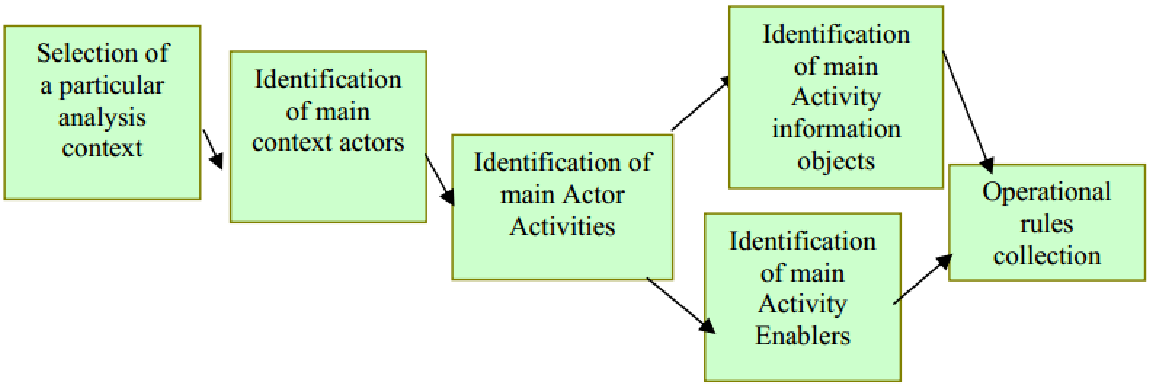

Selection of a particular analysis context
Identification of main context actors
Identification of main Actor Activities
Identification of main Activity information objects
Identification of main Activity Enablers
Operational rules collection

Figure8.Operational rule analysis steps. [17]

- **The IS architecture rules** are business rules set from an implementation perspective that is in accordance with the system architecture that has been set for implementation. Although these rules are outside the scope of this project, they are suitable for further developing the system.

2.5. Method reflection

A wide variety of methodologies concerning business rule **definition**, **modeling**, **expression**, and **management** are given in the literature.

Compared to the **GUIDE** Project, the Ross method is equipped with a very detailed and defined rule model. Furthermore, it allows flexible rule classification and even has a unique modeling technique that can be used to create a visual representation of rule statements. However, the GUIDE Project is equipped with a very solid structure to explain and define several rule types. The definition of job scheduling rules does not require an extended classification scheme, but a simple and straight-forward methodology that is understandable for all user types. Therefore, after comparison of the two most-used business rule definition methodologies, the GUIDE project was chosen. Furthermore, GUIDE has been used quite frequently for business rule definition and is far more standardized and globally accepted than the Ross method (mainly because the Ross method is quite new).

For business rule modeling, the widely supported **UML** was chosen to model use cases and process flow diagrams. The Object Constraint Language (OCL) would prove to be ideal for defining business rules, due to its extended constraint parameters which were added to the UML standard. However, in order to use this language for modeling rules a great deal of (developmental) implementation knowledge would have to be available. Due to the fact that Hansken is in its early (definition) stage, rule parts that include object-specific knowledge are unavailable. Therefore user-related diagrams were used to express business rules, because they are more suitable for the early stages of business rule definition. Concerning rule expression, it is clear that rules should be expressed uniformly, since no current BRMS system has been elected or chosen yet. Although no architectural rules will be defined in this project, but merely intentional and operational rules, they should still be expressed in a widely supported format. Therefore, usage is made of the Rule Interchange Format **(RIF)** that allows rules to be converted into vendor specific languages.

Following analysis of business management methodologies, it was seen that PROTEUS, BRADES and MBRM have much in common. Each method shares three different consecutive rule phases that allow for rule discovery up to rule implementation. The initial phase is often used for the crystallization of the project, by allowing the user to gain knowledge of the organization and its objectives. Furthermore, time is given to identify project stakeholders, domain ontology and finally define project-specific rules and constraints. After a solid basis for the identification of rules is found during the initial phase, the secondary phase, often called the operational or functional or deployment phase, is entered. In this phase, reference is made to business processes, actors, activities and information objects. Finally, rules are set from an implementation perspective that is in accordance with the defined system architecture.

The PROTEUS method was considered; however, in order to properly utilize this methodology, an extensive list of requirements would have to be obtained to allow software architects the design and development of the rule based system. Although this methodology might be useful in the construction of a suitable rule-based system, it would need to be provided with a great amount of details which are currently unavailable and outside the scope of this research.

Even though BRADES and **MBRM** share many similarities, the MBRM method is most suitable for the implementation of the Hansken job scheduler because it allows for business rule traceability to system components. The need for traceability between business rules and system components has been acknowledged by Kilov & Simmons. [32]Business rules are captured and stored in a structured form and templates that allow for linking them directly to software components. The necessity for business rule storage to provide for clarity, consistency and completeness has been proven by Feurlicht & Blair.[33] In turn, each rule is associated with important management information, such as business process and process owner. This provides for system transparency, which will be elaborated upon in section 4 (requirement principles).

Furthermore the MBRM method has proven itself in similar in large-scale industrial applications such as banking, electricity deregulation and e-business. Secondly, it can be used without the need for architecture-related information that is needed for the implementation of software.

The following table presents all methods chosen for this research, divided over four subcategories:

Element	Selected method
Business rule definition:	IBM Guide Project
Business rule modeling:	Unified Modeling Language (UML)
Business rule expression:	Rule Interchange Format (RIF)
Business rule management:	Manchester Business Rule Management (MBRM)

3. Results

3.1. MBRM Phase 1: Intentional view

- **What type of business rules should be implemented for a digital forensic application as a service?**

In accordance with the defined research management method, the initial step is to define business rule statements, seen from a business perspective, expressed in natural language. This section will describe the goals set for the Hansken job scheduler, information concerning rules and the processes involved. Furthermore it will identify the key actors and activities within the application. In the next section rules will be extracted, based on the information provided in this chapter.

The business rules were defined in consultation with all actors involved: the Hansken developers, system controllers and front-end users. They describe various features of the scheduling system including scheduling rules, priority management, tool (priority) management and resource management. The most important aspects were selected from all stakeholders and actors involved by conducting interviews.

During these interviews and brainstorm sessions, several ideas as well as issues arose. The following questions were asked concerning job scheduling in a digital forensic application presented as a service:

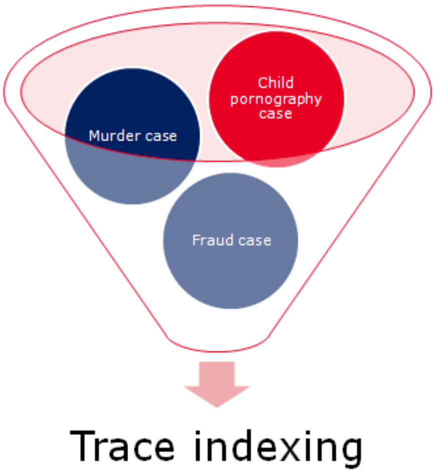
- What rules should be applied concerning:
  - Distribution of cases, events and processes (workload) using specific system resources or server nodes
    - Indexing and query processing distribution
  - Prioritizing and scheduling cases, events and processes
    - Job postponing or cancelling
    - Processes: Indexing and asynchronous query processing prioritizing
  - Event logging
    - Monitoring purposes
  - Notification parameters
    - Alert generation
  - Customizing toolsets per case type
    - Tool dependencies
  - Job validation

The results provided more than answers to the initial questions, as new ideas emerged and were processed. Often, users indicated similar requirements that the job scheduling system would have to adhere to. The final results of all interviews were anonymized and generalized to a set of rule requirements that will be presented in this chapter.

The following features for the job scheduling system in a digital forensic application as a service were defined from a business rule perspective by conducting interviews and will require operational rule expression:

Case priority

Before, during or after indexing digital images, case supervisors (OM users) must be able to *dynamically set case priority* for individual cases as well as *for individual tools*. The reason for prioritization could be given by the investigator, after which the Prosecution Office decides whether the given prioritization is valid. Next to a valid reason, the trial date, as well as the case type could be supplied by the investigator. In that way, it is absolutely clear at what date the case must finish indexing.



Case scheduling

Some cases do not have a higher priority, but still have to be finished before a certain time. Therefore, the implementation of a scheduler would be beneficial. The user should be able to set end times for indexing. The system could, according to the data size, data complexity and tools to be run, provide the user with an indication of the time needed for the indexing process to be completed.

Quick indexing (scan) option

In cases where time is of the essence, a quick assessment of the digital data could provide timely information regarding criminal activity. It might only need several tools with a few hits to discover useful evidence. The amount of tool-hits could also be defined by an authorized user, after which the user could be notified accordingly.

The quick indexing option could be implemented using a default set of tools, optimized for case specific goals. For example, if a child pornography case needed a quick scan, the digital media would only be scanned for image (meta) data or chat logs. If necessary, full option scans can be performed after quick indexing scan.

Indexing process

Do all police agencies have equal priority rights? It could be advantageous to set business rules regarding the amount of data that each corps can input: **a quota**. Perhaps a service level agreement (SLA) could be made between all parties that suggests the maximum amount of TB per day to be uploaded. Should a corps exceed the quota, additional measures would have to be taken, such as additional financing. On the other hand it would provide the NFI with valuable information regarding the amount of workload and capacity to be installed. Business rules could be set according to this information, such as the number of indexing hours.

Business rules can also be configured regarding the uptime of the indexing system. It is up to the NFI to set the systems working hours, which can be a variety of sets. For example, the system should perform indexing jobs on workdays from 8:00-17:00.

Within the business rules set, a specification for the case queue should also be set. Regardless of priority, should minor cases, with small amounts of data be given priority automatically? For example, should a fraud-case with only 40GB of image data be given priority over a fraud-case with 4TB of data if they have equal priority? Regardless of data complexity, business rules could be set for such matters.

A-synchronous query processing

When querying the system for data, a distinction can be made between synchronous and asynchronous queries. The first one is characterized by the fact that once the query is passed to the database server, the application waits until it receives an answer from the database server.

An asynchronous connection to the same database server means that multiple sets of commands can be passed and processed at once (multithreading). In this case, there is no need to wait for the query to return before sending another command to the database server.

Performing asynchronous queries is a task fit for the job scheduler to do. However, handling these queries can hamper indexing speed if they run at the same time. Therefore it would be preferable if they ran separately from each other and not simultaneously. In accordance with the business rules for job indexing, rules should be defined to establish when the scheduler should handle asynchronous queries.

Resource allocation & load distribution

Cases should automatically and efficiently be distributed over server nodes, according to the type of servers and resources that are available. Cases that require priority handling also require sufficient resources for them to finish faster than under normal circumstances. Therefore the system should automatically allocate resources for priority jobs, in order for them to finish faster. This could be done by either postponing or cancelling other current jobs. This process could be automated or set within business rules.

Enable or disable tools & tool dependency

Some tools require more indexing time than others. Therefore it should be possible to disable specific tools that do not seem necessary for a specific case. However, certain tools have to wait for the output of other tools because some tools reveal new data for other tools to work with. Therefore, users should not be able to disable all tools, because some are essential.

The scheduler has to automatically asses tool dependencies and reports these to the user. In order to do so, the system will have to know which tools are dependent on one another. Perhaps a dependency tree can be constructed.

Job validation

To provide for a valid chain of custody, all jobs have to maintain integrity and preferably be validated to do so. The scheduler should be equipped with a function that assesses all events, and report the suspicious ones.

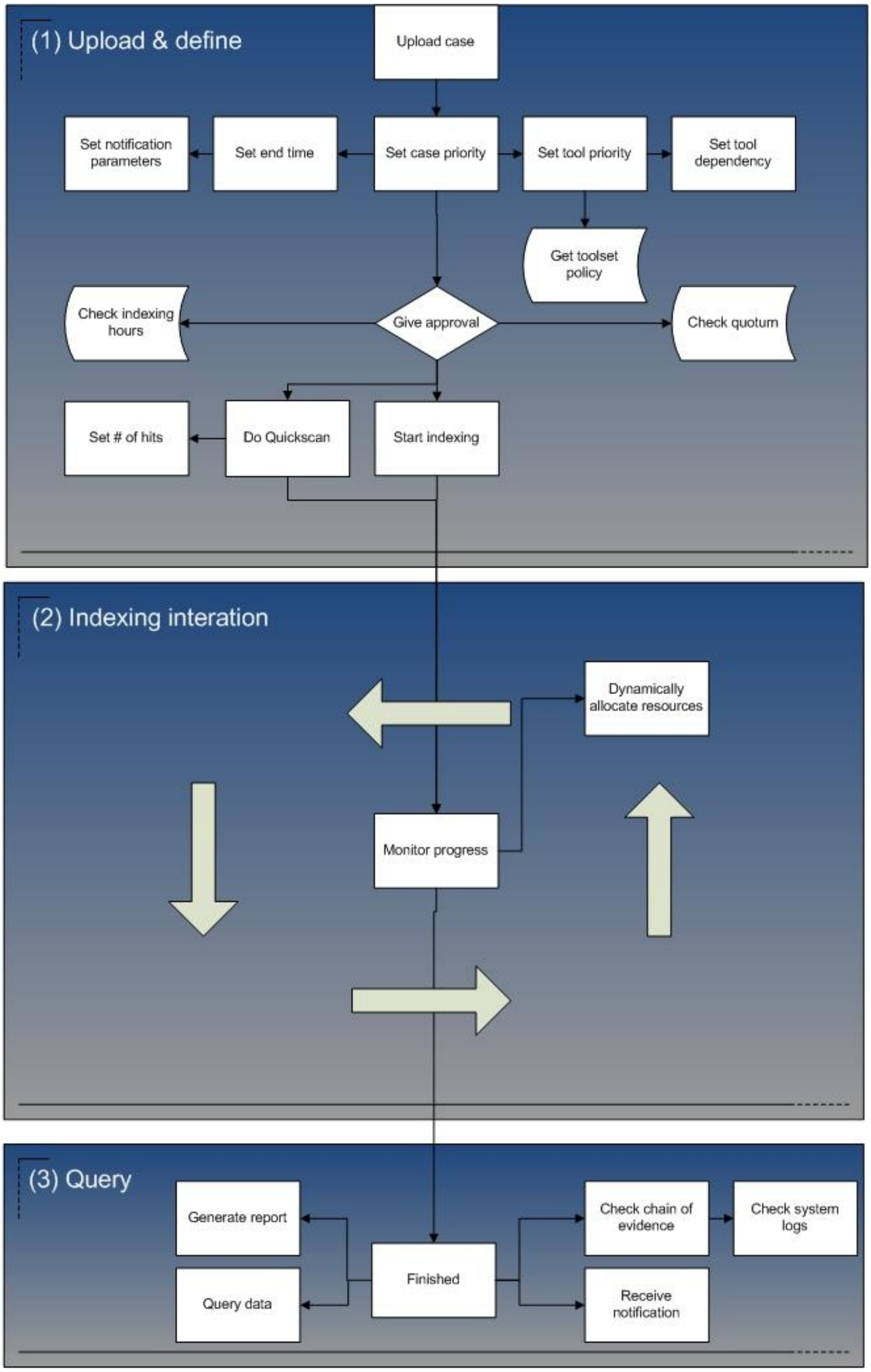
Alert generation

Business rules can be written that generate alerts when certain system events occur. The implementation of rules that describe ‘out of tolerance ‘events is advisable. As the job scheduler processes these events, it is easy to have it raise a notification as well. A typical business rule that raises a notification-event could be when systems are operating at 100% while jobs are being queued.

Event logging

The logging of events must be variable and should therefore be defined as a business rule. The rules should define which events are to be logged. It would be advisable to determine beforehand which events are vital to the chain of evidence. Some event logs could be needed for the generation of the end-report. A typical business rule concerned with event-logging could be to log all events that include any tool failure (such as the inability to index a certain part of an image).

Finally, the following process flow model provides an **overall perspective** on the three key processes that embed a variety of business rules that are handled in this chapter as business rule statements:



Hansken Scheduling – Process Flow Diagram
T.J. Schermer Voest 6/25/2012

3.2. MBRM Phase 2: Operational view

- **What specific business rules should be implemented for a digital forensic application as a service?**

The following operational rules have been defined according to the MBRM method. This chapter will address the business rule statements discussed in the previous chapter, and translate these business rule statements, composed of natural language, into formal operational rules. In this document, rules are expressed in a unified form, similar to the model presented in Figure 3. Finally, UML is used to present a complete use case.

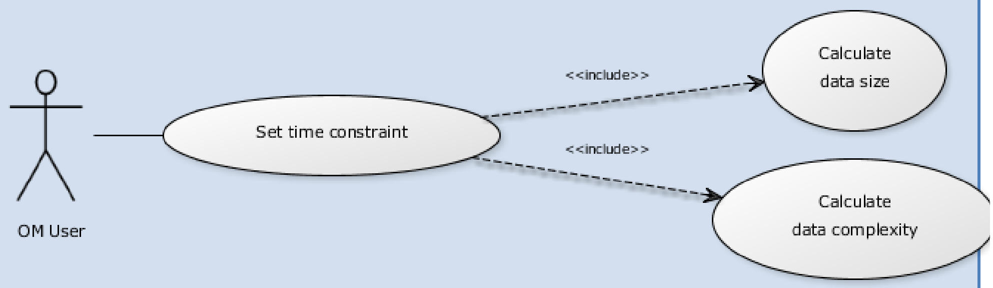
All rules have been specified for authorized users, which in most cases will be a member of the Prosecution Office (OM-User). Other authorized users are administrators, operators or controllers. It would be advisable to have an operator or administrator decide on various case-related parameters, such as case prioritization. This would prevent bias and tunnel vision amongst investigators. Moreover, a member of the prosecution office would have insight into the entire system (load), as well as its selectable options, as opposed to an investigator, giving him a better overview of the system and enabling him to make better choices regarding case parameters such as case prioritization.

Name	Assign case priority
Identifier	BR01
Description	Authorized users are granted the ability to assess and decide case priority. It must be possible to dynamically prioritize the trace indexing of an image, before and during processing.
Example	Child pornography cases are granted superior priority, if this is consistent with current policy or applicable to the case itself.
Intentional rule	• Authorized OM users can prioritize entire cases and individual images (structural and action assertion)

Operational rule	if (user.Authorized)
	{
	setCasePriority(image_Id || case_Id);
	}

Name	Set quickscan
Identifier	BR02
Description	Should a user want to perform a quick indexing scan (because time is of the essence), and not want to configure all tools individually, it might be wise to have a few tools to rapidly uncover relevant information. This could enhance indexing speed and still provide important traces in a shorter period of time, than if all tools were to run.
Example	
Intentional rule	• Authorized OM users can set quickscan parameters (structural and action assertion)

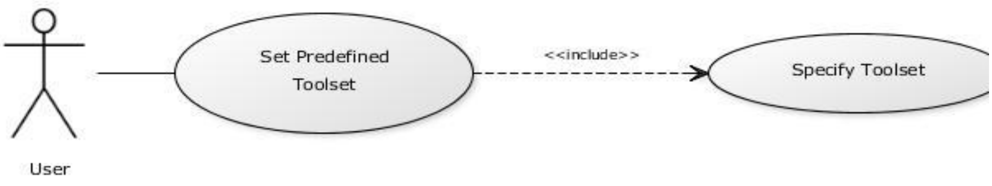
Operational rule	if (user.Authorized)
	{
	doQuickscan (KP);
	}

Name	Schedule case
Identifier	BR03
Description	Authorized users are granted the ability to schedule an image for indexing at a certain moment in time. This constraint could be in compliance with equal to the pre-calculated time needed for indexing. Only an end time would be supplied.
Example	• A fraud case with low priority needs to be done within a month • A murder case with high priority needs to be done within 4 hours.
Related rules	This rule is highly related to a setting case priority.
Diagram	 OM User Set time constraint <<include>> Calculate data size <<include>> Calculate data complexity
Intentional rule	• Authorized OM users can schedule cases and individual images (structural and action assertion)

Operational rule	if (user.Authorized && resource.Available && startTime.Set)
	{
	schedule(image_Id || case_Id);
	}

Name	Assign tool priority
Identifier	BR04
Description	Authorized users are granted the ability to assess and decide to give certain tools priority. A certain tool could also be given priority should it conflict with another tool.
Example	Some cases might benefit by assigning a higher tool priority, e.g. due to time constraints. For instance, a user could have a certain tool handle most data while other tools are dismissed.
Intentional rule	• Authorized OM users can assign tool priority (structural and action assertion)

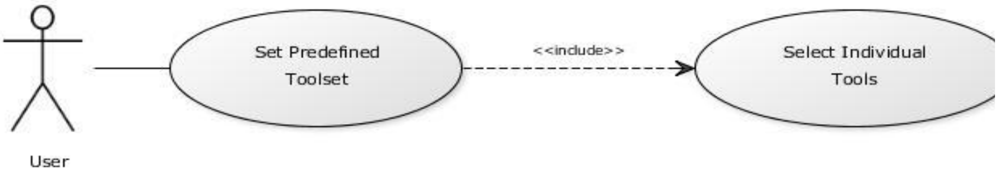
Operational rule	if (user.Authorized)
	{
	setToolPriority(case_Id, tool_Id);
	}

Name	Set predefined toolsets
Identifier	BR05
Description	This business rule defines what types of default toolsets are in compliance with certain case types. It could be advisable to have a child-pornography case treated differently from a fraud case.
Example	Child pornography cases could focus on tools such as image (meta) data, chat-&event logs, mail-tool and hash-compare tools. Therefore, these tools might have greater relevance for these types of cases than for a typical case of fraud.
Diagram	
Intentional rule	• Authorized OM users can set a predefined toolset for specific case types (structural and action assertion)

Operational rule	if (user.Authorized && case.Id == ‘KP’)
	{
	run_KP(case.Id);
	}
	else
	{
	run_Standard(case.Id);
	}

Name	Set number of hits
Identifier	BR06
Description	Within the quickscan ability or set predefined toolset business rule, it might be convenient to set a certain amount of hits a tool should have before a notification is made. The notification can be one of many forms.
Example	• If the hash-compare tool has found 1 match with the child porn image database, send a notification / alert.
Intentional rule	• Authorized OM users set the number of tool hits before notification (structural and action assertion)

Operational rule	if (toolHits.equals(value))
	{
	sendNotification(user.ID);
	}

Name	Enable or disable a certain tool
Identifier	BR07
Description	Authorized users should be given the ability to select merely a few tools, and if needed, disable others.
Example	The case in question only needs to be indexed and scanned for chat logs
Diagram	 User Set Predefined Toolset <<include>> Select Individual Tools
Intentional rule	• Authorized OM users can enable or disable tools (structural and action assertion)

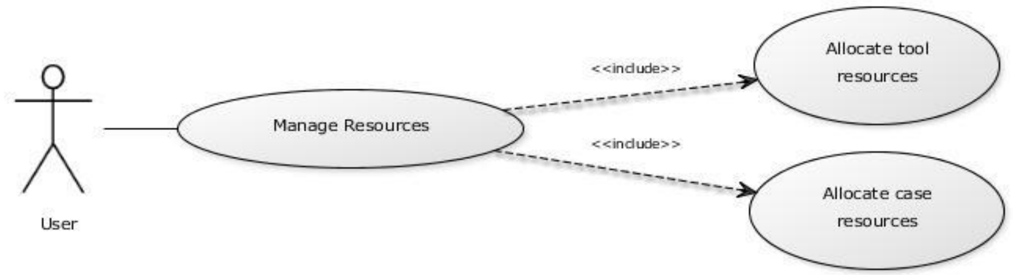
Operational rule	if (user.Authorized)
	{
	select(tool_Id);
	}

Name	Set indexing hours
Identifier	BR08
Description	When should the system be online for indexing of image-data? When should the system handle asynchronous query processing? Authorized users should be able to set and modify system indexing hours for different processes.
Example	• The system should perform indexing jobs/query processing from 8:00-17:00, only on workdays. • The system should perform indexing jobs/query processing from 8:00-17:00, every day. • The system should perform indexing jobs/query processing 24/7. • The system should perform indexing jobs/query processing only at night.
Intentional rule	• System administrators can set process priorities and uptime hours(structural and action assertion)

Operational rule	if (user.Authorized)
	{
	setIndexingHours();
	}

Name	Set corps quota
Identifier	BR09
Description	It could be wise to set business rules regarding the amount of data every corps can input on a daily or weekly basis: a quota.
Example	• The Amstel/Amsterdam corps can input a daily amount of case data no more than or equal to 10 TB for indexing.
Intentional rule	• System administrators and OM users can set the quota for each police agency

Operational rule	if (user.Authorized)
	{
	setAgencyQuota(datasize);
	}

Name	Manage resource allocation
Identifier	BR10
Description	Authorized users must be able to dynamically allocate resources before and during the indexing process, for the entire case as well as for specific tools. They are granted the ability to assess and decide to allocate a certain amount of resources to certain jobs events. • Dispatch tools to certain servers, close to source data • Dispatch a tool to a certain server location, close to source data Suppose the system's capacity hits 100% and a high priority job is stacked within the queue. Should certain events simply be postponed, cancelled, halted or be finished?
Example	A case is given additional CPU power, which will benefit most essential tools
Diagram	 User Manage Resources <<include>> <<include>> Allocate tool resources Allocate case resources
Intentional rule	- System administrators can dynamically set parameters regarding resource allocation: • Resource tool-related parameters (CPU - GPU - Memory - Hard disk I/O) • Resource location parameters • - Manual postponing and canceling of jobs at certain resources

Operational rule	if (user.Authorized)
	{
	allocateResource(job_ID || tool_ID || image_ID || case_ID);
	postpone_low_priority_jobs(job_ID || tool_ID || image_ID || case_ID);
	}

***Remark**	Allocation of resources can also be **automated** (according to a priority assessment) but should be configured according to business rules.

Name	Generate logs
Identifier	BR11
Description	In order to have a valid chain of evidence, job scheduling logs could be consulted to provide for index processing details.
Example	• How should event-logs be generated within the scheduler? • What events should be logged?
Intentional rule	• System administrators can set log generation parameters

Operational rule	if (user.Authorized)
	{
	setLoggingParameters(system);
	}

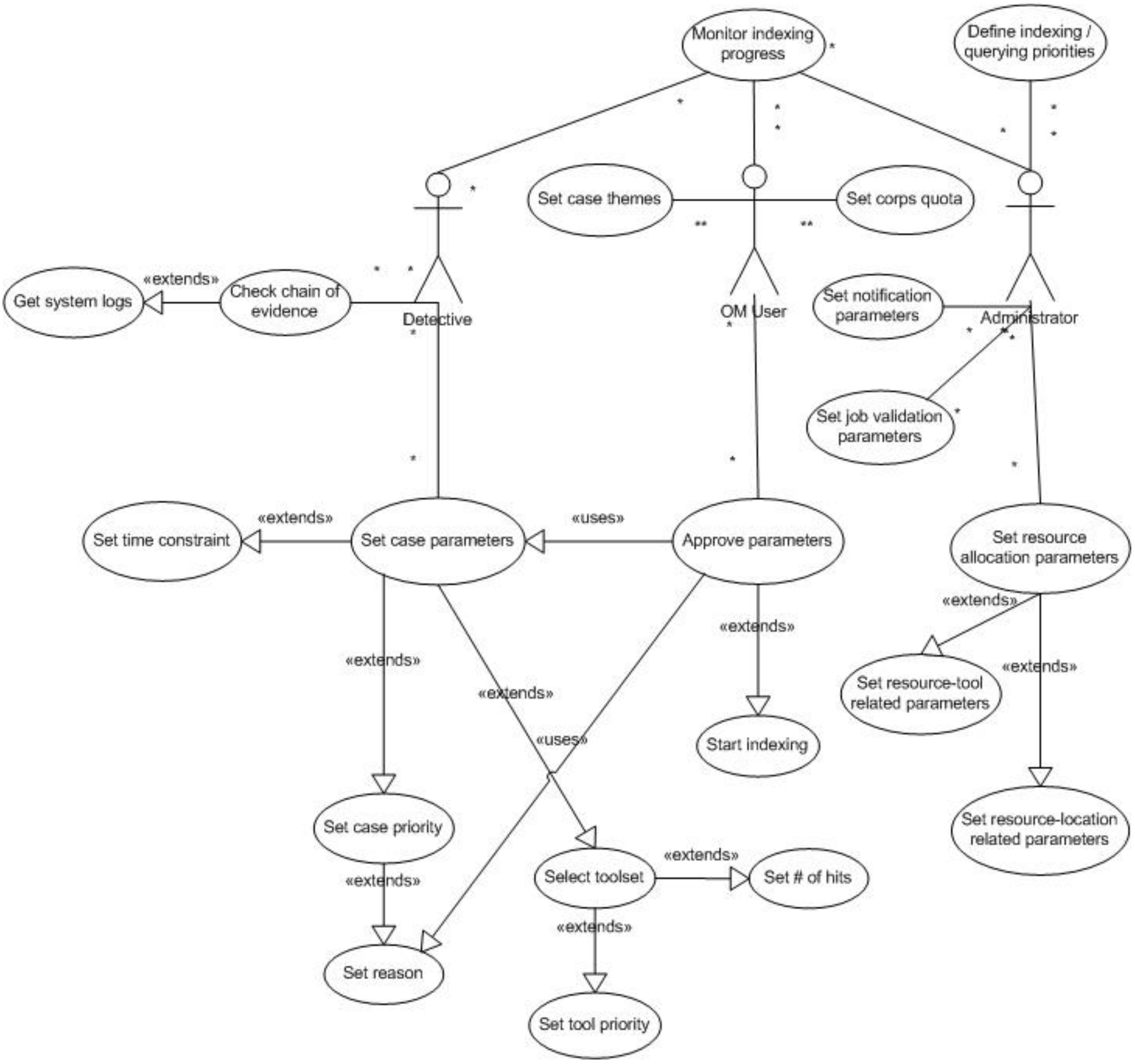
Name	Job output validation
Identifier	BR12
Description	Validation of all jobs is essential in order to guarantee the chain of evidence. Several business rules can be applied to the validation of jobs within the scheduling system. Even though the method of validation is programmed, its configuration should not be.
Example	• Configuration of validation of job events within the scheduler
Intentional rule	• System administrators can set job validation parameters

Operational rule	if (user.Authorized)
	{
	    setValidationParameters(system);
	}

Name	Generate system alert
Identifier	BR13
Description	It could be beneficial to set parameters concerning the generation of alerts. These could be constructed as rule sets.
Example	When should system alerts be given? • If the system reaches 100% load? • If cases are indexed incompletely? • When part of a trace indexing process is ready, output is made available immediately, even before other trace indexing processes have been processing the image, at what level of progress should an alert be given?
Intentional rule	• System administrators and OM users can set alert generation parameters

Operational rule	if (user.Authorized)
	{
	    setAlertParameters();
	}

The following use case describes all defined business rules for a forensic application as a service according to the operational view of MBRM method:



4. What requirement principles for the BRMS should be met?

Because a BRMS accompanied by business rules can be the solution to having a job scheduling service, its potential should be assessed by using the forensic guideline principles. In this chapter, the functioning and application of a BRMS will be explained (section 4.1) and an evaluation will be given of several requirements based on principles (section 4.2).

4.1. What is a BRMS?

A business rule management system is typically a compilation of software tools that allow for the creation, management and support of business rules in an organization. The system separates the business logic from the IT environment. By allowing rule modification to be done in a simple and accessible manner, business analysts are given back control over IT infrastructures.[34]

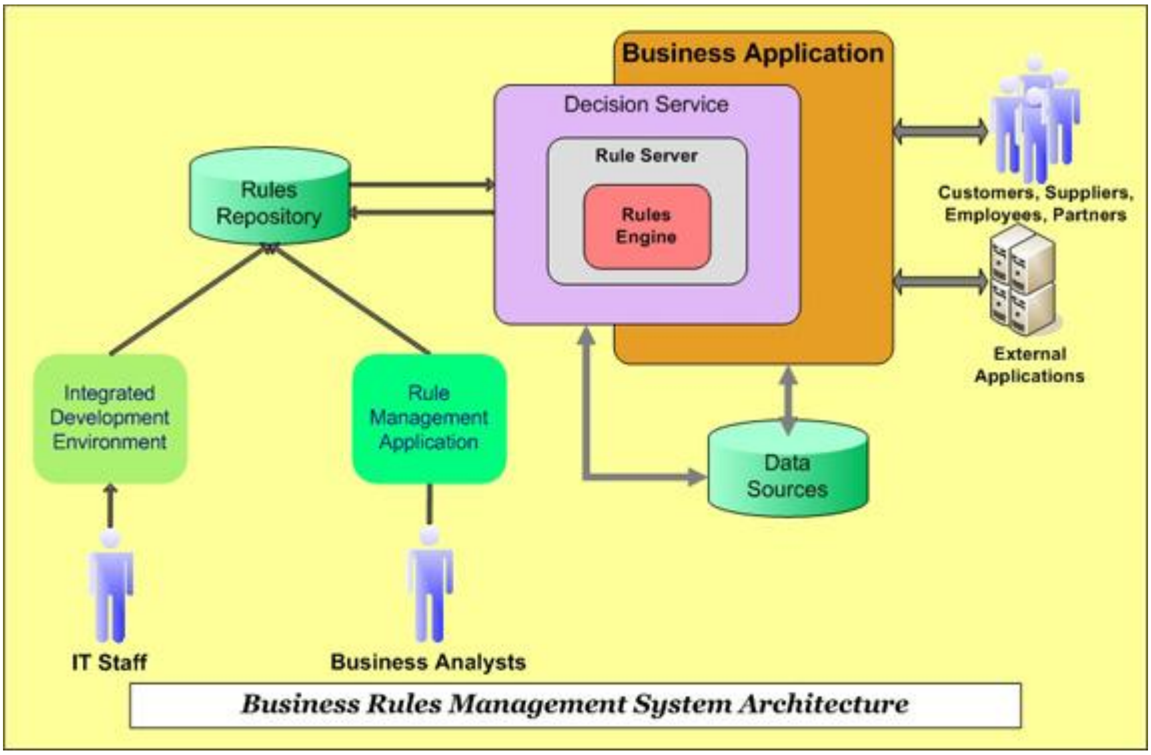
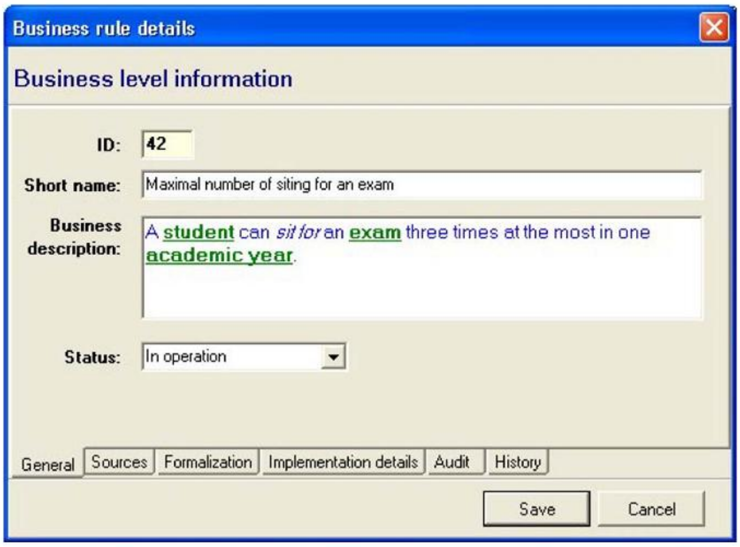

Business Application
Decision Service
Rule Server
Rules Engine
Rules Repository
Customers, Suppliers, Employees, Partners
External Applications
Integrated Development Environment
Rule Management Application
Data Sources
IT Staff
Business Analysts
Business Rules Management System Architecture

Figure 9. A BRMS architecture (source: IBM)

Regarding figure 9: On the bottom left, the IT personnel create the framework necessary for the organization to create and manage rules. Next to the IT staff are the business analysts that use a GUI in order to create and manage rules. Both the integrated development environment and the rule management application lead to the rules repository, a database that includes all rules available to the decision service. The rules repository can be typed as a user-driven system evolution environment. This service interacts with the business application and data sources. In the Hansken system, this would respectively be the toolset and image source data. The customers are represented by tactical detectives.[35]

Because the Hansken system is a decision rich environment, it requires a system that can make numerous rule-based decisions upon rules specified within the rule repository. The job scheduling system can be defined as a variable environment; therefore it needs a variable-based rule management system. A BRMS system can easily handle policy changes, expressed as rule changes. Furthermore, rules can be accessed from any given place and allow for the complete automation of job events.

In a typical BRMS environment, a business rule would be defined on the business and IT level as follows:



Business rule details

Business level information

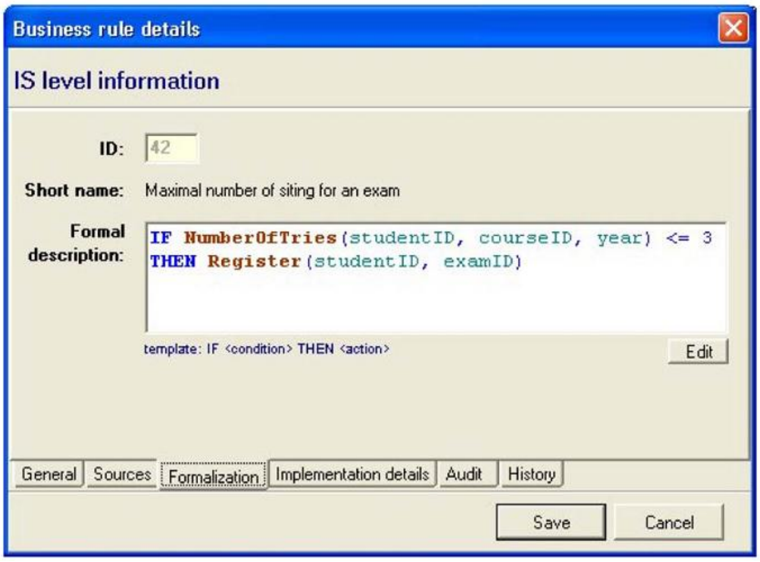
ID: 42

Short name: Maximal number of siting for an exam

Business description: A student can sit for an exam three times at the most in one academic year.

Status: In operation

Figure 10, typical configuration of a business rule in a business management environment on the ***business level***



Business rule details

IS level information

ID: 42

Short name: Maximal number of siting for an exam

Formal description:

```
IF NumberOfTries(studentID, courseID, year) <= 3
THEN Register(studentID, examID)
```

template: IF <condition> THEN <action>

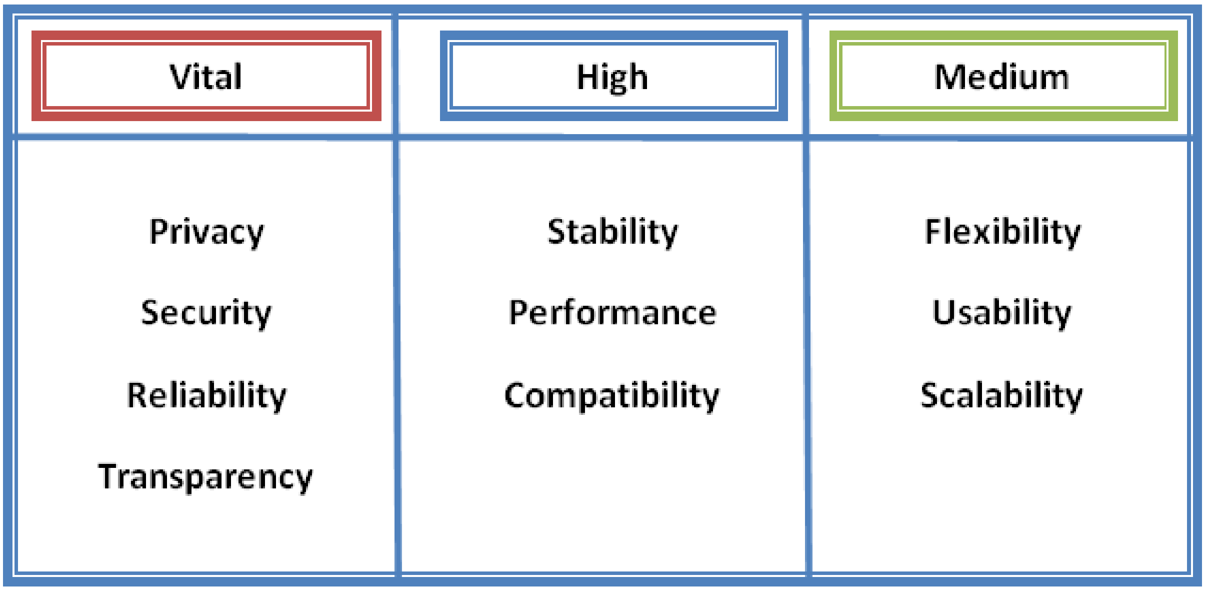
Figure 11, typical configuration of a business rule in a business management environment on the ***system level*** [36]

4.2. What are the BRMS requirements for a digital forensic application as service?

Business rules are derived from the business itself. Therefore BRMS should conform to the organization's demands for several key components, such as reliability, integrity and stability. The principles that have to be met for a digital forensic application as a service, and particularly the job scheduling system, are evaluated in this section and are based on user interviews from the Digital Technology staff, the Hansken high level document as well as requirements extracted from the needs and standards set by the digital forensic community (based on literature). These have been combined and put in perspective to form a unique set of principles for BRMS implementation.

The requirements for the BRMS were based on the evaluation of user feedback, forensic standards and the Hansken high-level design document. A total of 10 users, from 3 different disciplines, who will interact with Hansken on different levels were interviewed by means of a questionnaire. By organizing brainstorming sessions, various requirements were evaluated and ranked according to importance. For instance, security in a BRMS was found to be more important than flexibility, which is reflected in the final results.

It is clear that all requirements are essential, but some are more important than others. When selecting a proper BRMS for a digital forensic application that works as a service, this needs to be kept in mind. The BRMS selected should reflect the standards and values that the NFI and the forensic IT community set store by. These values naturally demand high standards regarding system privacy and security. As all requirement principles were initially lumped together, they needed to be pulled apart and categorized according to importance, as seen in figure 12.



Vital	High	Medium
Privacy	Stability	Flexibility
Security	Performance	Usability
Reliability	Compatibility	Scalability
Transparency		

Figure 12. Requirement principle importance diagram based on the study of all principles in this chapter

Privacy & Security (V)

First of all, data that is fed into a forensic system can be identified as information that needs to be protected. It is not only **vital** to federal law enforcement and investigation agencies, but also to the general public (as well as victims and offenders). Policing and security organizations in particular are prone to the huge impact information leaks can have. Therefore privacy and security are of **vital** importance.

Even though the BRMS might not directly handle crucial information regarding cases, event-handling, business rule modification and communication has to be secured; the chain of evidence has to be valid at all times. For that reason, the first two requirements (Privacy and Security) were chosen to be of highest importance. The BRMS has to comply with these values in order not to compromise system integrity as a whole, because systems are likely to be interconnected.

In order for the system to work with highly sensitive and classified data, it is of the utmost importance that the system functions within sufficient privacy and security parameters. Furthermore appropriate measures must be taken to prevent breaking classification legislation, loss of cases, and protection of individuals and reputation damage. Together with other key components, the BRMS has to provide the same level of security. No weak links in the system are allowed.

The BRMS should not leave unwanted traces that might include case-related data. Therefore system event logs should be stored securely and only be available to authorized sources. Furthermore, business rules should only be accessed, modified and deleted by authorized personnel; proper authentication is a must. Business rules should also be modified without leaving any traces, e.g. they should be adapted using a GUI. It would be preferable if the BRMS were connected (or modified to do so) with the authentication system of other key systems, such as an already existing LDAP.

Finally, guaranteed patching and update procedures are essential. Commercial solutions can probably provide these features, as well as long-term stability and scalability. However, some open-source alternatives also enable fast patching, such as Drools.

Reliability (V)

Reliability can be typed as the confidence that technology-powered business processes will be available, perform well, and adapt to changing business conditions. It can be measured as the probability of failure, frequency of failure or availability. Above all, the BRMS would have to be reliable, but still provide sufficient data processing power and stability. It should be clear how the system will respond to power failures and backup procedures. *Are events logged? What happens to the events when failure occurs?* It should be clear how the system will respond to all sorts of events that might upset system reliability. Another concern for reliability is the detection of faulty or suspicious events. *How should events be handled that are considered a compromise to system reliability?* The system should incorporate a clear policy regarding these questions. Some of these questions are dealt with in the stability section.

The system should also have an unambiguous policy regarding event-logging, because the **chain of evidence** must be maintained and transparent at all costs. How long should these events be stored? Can events be hashed for proper validation? A policy regarding the maintenance and validation of these logs should be defined. Because having a proper chain of evidence is crucial to the NFI as well as the entire forensic community, a reliable system is **vital**.

Transparency (V)

Due to the fact that the Hansken system works with confidential data, all actions and events handled within the BRMS should be traceable and visible. To provide for proof of the chain of evidence (e.g. evidence reports), all system and user actions, including those performed within the BRMS, should be logged. All tool functions related to processing the image file must be visible and traceable. Therefore event-logging must be accounted for.

With the business rules separated from the engine core, transparency is automatically increased. The system becomes fit for easy auditing, monitoring, debugging and analysis (upon failure). In this way, analysts can ensure that business rules reflect business policy. Transparency is categorized as a **vital** key principle because a transparent system can increase system trustworthiness.

Stability (H)

Stability is categorized as a fairly high key principle because the BRMS should be resistant to change (e.g. inflicted by changing business rules). Secondly, the software should be able to withstand stress, e.g. a decision service has to process (decide for a) large quantity of events based on a wide set of stored rules (related to performance).Furthermore, stability is co-related to reliability, which is of vital importance. Therefore, dependability on the correct functioning of a BRMS should naturally be of **high** importance.

Long-term stability is decided by several key factors. It is essential to have an automatic backup system that schedules backups, which guarantee business rule availability. Furthermore, the ability to make backups also requires that the BRMS measures have an adequate recovery management. In the event of a power failure, business rules have to be secured.

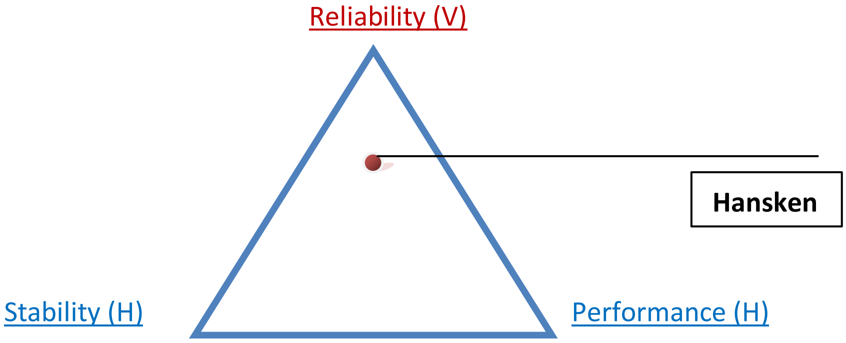
Performance (H)

As mentioned in the stability section, BRMS performance can be expressed in the number of logical decisions (consequences) made for a certain quantity of events based on a set of asserted facts or axioms. In order to provide for fast decision-making, nearly every BRMS is equipped with the Rete-algorithm, a powerful basis for a rule engine, which can substantially increase performance. When choosing a proper BRMS, Rete features should be assessed.

Most vendors have their BRMS equipped with the Rete algorithm. Furthermore nearly all software packages (e.g. from IBM, Oracle, Fair Isaac, Red Hat and Pegasystems) allow for upgrading a rule execution engine in addition to – or to simply replace – the current engine. In that way, if a newer or more advanced rule engine is developed that uses a more enhanced version of the Rete algorithm, rule processing can be accelerated to 10 to 1000 times faster, as happened when Rete-NT replaced Rete-2in 2010. The Rete-NT algorithm is, in most scenarios, at least 500 times faster than the original Rete and 10 times faster than Rete-2. [37]This is a typical example of software scalability that allows for future-proof performance.

The BRMS should be able to handle ever growing event loads. According to predicted estimations, the system would have to process a case load of approximately 110 TB a day in 2014, which comes down to processing 15 GB a second. This will result in handling 16TB of indexing data a day. The system would therefore be able to handle considerable loads of job events per second.

In this research, performance is indicated as a high instead of a vital importance principle. In practice and coexistence, reliability and stability principles have a profound effect on the performance of a system. For instance, fail-safe systems and checks to increase reliability inevitably affect performance. Because reliability principles are indicated to be of higher importance, due to organization and forensic standards, than performance and stability, the latter two are indicated as **high** instead of vital importance.



Compatibility (H)

Due to the fact that the Hansken system will mainly be Java-based, it could be helpful to use a Java-based rule engine that follows protocols similar to the Hansken system. In that way, the likelihood of system compatibility is far greater. Furthermore it would be easier to implement as there would be no need for translation agents. System developers would benefit from compatibility. The following BRMSs are compatible with the JSR 94 (Java Rule Engine API):

- *Oracle Business Rules*
- *IBM ILOG jRules*
- *Drools by Red Hat*
- *OpenRules*
- *Blaze Advisor*
- *Pega Rules Process Commander*

Compatibility is rated as a **high** level principle, because many developers as well as digital forensic scientists expressed the need for a Hansken compatible system, preferably Java-written or equipped with a proper API.

Flexibility (M)

Flexibility concerns the system's ability to adapt to new circumstances in a business. It should be possible to rapidly and frequently change business rules within the BRMS without disabling the system. Evolving business conditions and varying policies require certain flexibility. For example, recent developments, or changes in regulations could directly influence priority themes for cases.

Therefore it should be possible to dynamically add, remove and alter business rules. For example, Drools has its own Rules Language Engine (DRL), which is quite simple to modify using a standard set of rule terms. Rules are stored as .DRL files. Such files are quite easily modified using the GUI that comes with the BRMS, largely controlled by means of a web-based administration console.

Flexibility was rated as a **medium** principle due to the fact that a business rule management system owes its very existence to making IT systems flexible because of the separation of computer code and policy rules. Therefore one can assume that almost every BRMS is inherently flexible. Flexibility comes naturally with the implementation of this system and thus does not need further priority.

Example of DRL

```
rule "name"
attributes
when
        statement
then
        statement
end
```

Complex event processing (CEP)

When selecting an appropriate BRMS it is vital to asses goals concerning the manner in which events are processed. Most modern BRMS are equipped with artificial intelligence that allows them to search for patterns of events, enabling complex event processing. These patterns can be correlated, so that patterns can be manipulated and bound to certain rules. Also, CEP allows for combining data from various resources and extracting threats or opportunities from them. Complex event handling allows businesses to add value to decision-making in comparison to ‘normal’ event handling.

Usability (M)

Usability stands for a wide range of principles; in essence it defines the elegance and clarity of the interaction between human and machine. In order for users to simply and quickly modify business rules, the system should provide a logical (GUI) interface. Furthermore it should be consistent, provide proper feedback in the event of false input and most importantly be user-friendly.

However, prior to selecting a BRMS, attention should be given to the kind of user that will handle the systems business rules. Requirements concerning usability are defined according to their level of expertise. In the case where a technically skilled operator implements business rules, it might not be essential that the system has a modern GUI. Because it is quite likely that the rules will be defined by skilled personnel capable of defining architecture business rules, requirements regarding usability are of **medium** priority. However, if unskilled personnel are to change business rules, a more straightforward and easy to use method for defining business rules should be considered.

Scalability (M)

In the event that the NFI decides to introduce the Hansken system in other countries, proper scalability is a must. In order to accommodate for future growth, the system should be able to cope with high customer demands and ever increasing event handling. Clustering, caching, failover, load balancing, and distributed deployment features might be of great importance and should be carefully considered. Secondly, in the near future, the system might attract attention of foreign forensic investigation bureaus. Reselling the system is a likelihood that should be considered.

Scalability was rated as a **medium** priority requirement because a BRMS is already inherently scalable. A decision rule repository, which contains rule sets that can be modified from any remote pc, often exists as only one instead of a distributed set of servers, which is in fact why the system is strong in scalability. Only the rule engine is distributed locally, which assesses events and triggers a certain rule if needed. Every rule engine checks the repository server to determine whether a rule has changed over a certain time period. In short, BRMS scalability should not be a problem.

Conclusion

All requirement principles have been defined to be of **vital**, **high** or **medium** importance according to the evaluation of user feedback, forensic standards and the Hansken high-level design document. Privacy, security, transparency and reliability principles were marked as vital, whereas performance, stability, compatibility were marked as high priority principles. Medium principles were usability, flexibility and scalability. Predefined principles are part of the selection criteria for software and hardware. Because rule handling is no longer performed within the code of the system, but by a BRMS, certain requirements are needed, which this chapter discussed in depth.

Before selecting a package, a list of specific requirements is needed that are extracted from the requirement principles in this section. For example, concerning compatibility, a specific requirement can be that the system should be provided with the JSR 94 API. In the recommendations section, a number of different examples are given that provide a basis for follow-up research for actually selecting a BRMS.

Although ultimately all requirements are important to achieve a suitable BRMS, it is up to the NFI to make a final decision in prioritizing the various principles. This chapter can serve as a guideline to do so.

Recommendation for specific requirements based on principles

The next step would be to translate the requirement principles into more concrete features that allow a single BRMS to be chosen. In figure 13, several BRMS systems have been set against several feature-based requirements. The diagram has not been filled in, because it is outside the scope of this project. However, it could provide a suitable framework for ultimately choosing a BRMS.

BRMS requirements (specific)

	Pega Rules Commander	Drools	Hammurapi Rules	ILOG JRules	Jess programming	Blaze Advisor	Oracle Business Rules	OpenRules
JSR 94 API(compatibility)	✗							
Mean time between failure (reliability)	2 h							
Rete 1/2 OO or NT (performance)	NT							
Backup / restore procedures (stability)	✓							
License type (ASL / LGPL)	ASL							
Commercial support (duration)	5 y							
Elaborate documentation (usability)	✓							
Rule Editor GUI (usability)	✓							
Active R&D (patching)	✗							
Pattern recognition (CEP) (flexibility)	✗							
Event logging options (transparency)	✓							
Event monitoring options (transparency)	✓							

Figure 13, an example of a requirement table that could be used as a basis for follow-up research.

5. Evaluation of results

This section will assess the results of the present research, which are the collected business statements, as well as the intentional and operational rules extracted from them. The results were **validated** according to the following criteria:

- The results should provide an answer to how business rules are defined, modeled, expressed and managed in order to achieve a suitable set of business rules for the Hansken job scheduler.
    - How does one guarantee that the business rules that have been defined are suitable?
- The results should provide an answer to what requirements are needed for selecting a suitable BRMS for Hansken.
    - How does one guarantee that defined requirements are suitable?

This means that an answer must be provided as to whether the results are beneficial or not.

Rule validation

Before this research project started, it was unclear which job scheduling processes would have to be configured and according to which rules. However, the provision of a set of primary business rule statements constituted the first step towards acquiring a set of business rules. The business rules were broken down into intentional and operational rules. The first rule set was defined according to user input from developers, front-end users and administrators. This input was subsequently compared and only viable rules were extracted from this data, as well as from the Hansken high-level design document. It was crucial that the wishes of individuals were carefully assessed, as they might not represent or be consistent the objectives of the organization. Moreover, features that did not yet exist had to be assessed for technical feasibility.

When designing and implementing new systems, it is vital that rules taken from previous systems are validated, which means they have to be checked and adapted accordingly if necessary. Furthermore it is essential to assess whether new rules can in fact be implemented into the system. For example, XIRAF allows 5 cases to be indexed simultaneously, so should Hansken be provided with the same rule, or should it be changed to a different value? Simply copying rules can produce a system that is unable to function properly. However, if the defined rules prove to be inadequate during the implementation phase, they can easily be modified. Changing rules at different stages is also common practice, due to changing policies and trial-and-error.

In order to prove or disapprove a business rule, it should be implemented in a test environment. The only way to properly validate a rule is to have it work with actual case data and to analyze its behavior on the basis of the events that emerge. It is recommended that a system for testing business rules be designed.

Requirement validation

In essence, the requirements defined should represent the goals of the NFI regarding Hansken, although they can also be applied in similar applications in the digital forensic community.

The requirements will likely change when the project reaches the implementation phase, due to changing user policies and demands. The current requirements were validated according to the organization's standard for several other forensic applications that are currently being used. However, since Hansken is a fairly new concept, requirements are experimental and cannot be validated according to predefined organizational standards. Nor can it be sufficiently guaranteed that the current requirements will reflect the final BRMS requirements. The main goal is to inspire and empower users to further brainstorm on the importance of all the aspects defined and translate them into specific requirements. For example, in the field of performance, the system should be equipped with the latest Rete-NT algorithm for quick and efficient event handling.

6. Overall conclusion and recommendations

Hansken will be the system selected to precede a digital investigation as a service. It will give digital forensics a new impetus by putting innovative and ground-breaking new ideas into practice, something previous solutions were unable to do. However, several essential steps need to be taken if ideas are to be properly translated into technology. One of the most vital subsystems of a digital forensic application as a service is the job scheduling system. This research took the most important ideas regarding a job scheduler system and converted them into a well defined and structured perspective, not only for the NFI but also for the forensic community as a whole.

The need for capturing, defining, modeling, expressing and overall management of business rules was essential to take the initial steps towards implementing a business rules management system for a job scheduler. As a result, a literature study was conducted that assessed a wide variety of methods in relation to the needs mentioned. Finally all methods were evaluated and a selection of suitable methodologies was chosen to work with. Based on a number of criteria, four methods were chosen for each part of the project.

The project required a management methodology with a clear set of stages regarding rule definition and capturing, which was provided by the MBRM method. Although BRADES and PROTEUS approaches could have provided a suitable basis for the project, MBRM was chosen for its clear separation of rule definition into three stages and the possibility of constructing rule sets in a chronological and homogeneous perspective. It provided an ideal basis for the project by allowing business rules to be defined in natural language, derived from business rule statements. As a follow-up, the method included the possibility to translate intentional rules into operational rules, which in turn allows the creation of architecture (implementation) rules.

All defined rules demonstrate the strength of MBRM, which enables expressing, structuring and organizing the business rule statements of many completely different types (e.g. integrity constraints, derivations, workflow rules) in a clear manner. The natural language expressions are easy to understand for all levels of users. The operational rules were defined in a Rule Interchange Format (RIF) that allowed for a uniform and standardized approach towards implementation. By not expressing rules in a vendor-specific format, a broad range of rule management systems can be chosen from. The same applies to modeling rules in a uniform format; therefore Unified Modeling Language (UML) was used to generate use cases and a flow diagram.

In the first stage of the MBRM method, business rule statements were defined that included actor roles and activities. On the basis of these statements, individual business rules were extracted in the second, operational stage. Furthermore, a process flow diagram and use case were constructed. The final stage that links business rules to exact system implementation specifications was outside the scope of this research. This step establishes a correlation between the initial analysis of the system and the proposed design. It transforms the implementation of free requirements to the implementation of specific requirements and specifications. A method for this transition could be the creation of class diagrams on the basis of earlier contextual work.

This research demonstrates the need for a forensic application that functions as a distributed and widely accessible service, capable of handling large and complex data sets in the form of images, as well as incorporating a job scheduler. Moreover, it provides the forensic community with a clear overview of many suitable business rules, including their motivation for implementation. Furthermore, several key requirements for an appropriate business rules management system were evaluated. As this document only serves as a guideline, in the end, it is up to the NFI to decide which factors are the most important for the introduction of a BRMS for the Hansken system and which business rules are to be implemented.

The next step will be assessing suitable business rules management systems and applying business rules to a rule language that is interpretable for a wide range of BRMSs. The implementation of business rules is an ongoing process; as the rules fluctuate, new rules are bound to manifest themselves.

This research has proven that it is possible to produce a uniform and standardized approach to defining, expressing, modeling and managing business rules by utilizing a unique collection of well known methodologies. As opposed to the usage of formats that are bound to vendor-specifications and systems, this research demonstrates that it is possible to use a selection of suitable methods according to predefined criteria. In turn this provides system users with the freedom they need to adjust the direction of a project in a later phase if necessary.

7. References

[1] Thompson et al, Winds of Change and a Look at the Future of Digital Forensics, 2008

[2] Illustration, Netherlands Forensic Institute, Digital investigation as a service, 2012

[3] Computer Forensics, Past, Present and Future, EWA HUEBNER, DEREK BEM, AND OSCAR BEM, University of Western Sydney, 2007

[4] FTK Toolkit - http://accessdata.com/products/computer-forensics/ftk(viewed 29-7-2012)

[5] Zylab Forensic Analysis - http://www.zylab.nl/Software/LawEnforcementInvestigations.aspx(viewed 29-7-2012)

[6] Illustration, Netherlands Forensic Institute, XIRAF system, 2011

[7] Illustration, Netherlands Forensic Institute, Hansken Big Data documents (High end level design), 2012

[8] D. Rosca, Application of a Decision Support Mechanism to the Business Rules Lifecycle, 10th Knowledge-Based Software Engineering Conference (KBSE95), Boston, MA, 1995

[9] H. Herbst, Business Rule Oriented Conceptual Modelling, PhD Thesis, Physica-Verlag, 1996.

[10] M.I. Krammer, Business rules: automating business policies and practices, Distributed Computing Monitor May (1997) 1997.

[11]The Business Rules Manifesto: http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf (viewed 3-7-2012)

[12]Defining business rules: http://www.businessrulesgroup.org/first_paper/br01c0.htm (viewed 5-7-2012)

[13] Ross, R. G. The business Rule Book (2nd ed.). Business Rule Solutions, Houston, 1997.

[14] Modeling languages for business processes and business rules: A representational analysis Michael zur Muehlen a, Marta Indulska b, information Systems 35 (2010) 379–390

[15] Rosca et al, Towards a flexible deployment of business rules, Department of Software Engineering, Monmouth University, Department of Computer Science, Old Dominion University, 2002

[16] Kardasis, Expressing and organizing business rules, Deloitte and Touche Consulting, Department of Computation,

[17] José L. Martínez-Fernández, K-SITE RULES: Integrating Business Rules in the mainstream software engineering practice, 2006

[18] I. Petrounias, P. Loucopoulos, A rule based approach for the design and implementation of information systems, in: M. Jarke (Ed.), Proceedings EDBT '94, Springer, Cam-bridge, UK, 1994.

[19] Bajec, A methodology and tool support for managing business rules in organizations
Faculty of Computer and Information Science, University of Ljubljana, 2001

[20] H. Herbst, The specification of business rules: a comparison of selected methodologies, Methods and Associated Tools for the Information Systems Life Cycle, Elsevier, Amsterdam, 1994

[21] Bajec, A methodology and tool support for managing business rules in organizations
Faculty of Computer and Information Science, University of Ljubljana, 2001

[22] H. Herbst, Business Rules in Systems Analysis: A Meta-Model and Repository System (1996)

[23] E. Gottesdiener, Capturing Business Rules, Software Develop. Mag., (1999).

[24] R. Ross, The Business Rule Book: Classifying, Defining and Modelling Rules, 2nd Edition (Ross Method), 1997.

[25] D.L. Struck, Business rule continuous requirements environment, Colorado Technical University, 1999.

[26] Bajec, A methodology and tool support for managing business rules in organizations, Faculty of Computer and Information Science, University of Ljubljana, 2001

[27] Bajec, A methodology and tool support for managing business rules in organizations
Faculty of Computer and Information Science, University of Ljubljana, 2001

[28] Kardasis, Expressing and organizing business rules, Deloitte and Touche Consulting, Department of Computation

[29] Omara et all, Performance Analysis and Design of A Wireless Networks, Computer science department, International Journal of Engineering Science and Technology Vol. 2(6), 2010

[30] Kardasis, Expressing and organizing business rules, Deloitte and Touche Consulting, Department of Computation

[31] P. Kardasis, Managing business rules during the requirements engineering process in rule-intensive IT projects, in: W. Abramowicz, G. Klein (Eds.), Proceedings of 6th International Conference on Business Information Systems, 2003

[32] Kilov, H., & Simmonds, I. (1997). Business rules: From business specification to design. Technical Report RC 20754, IBM TJ Watson.

[33] Feuerlicht, G., & Blair, A. (1990). An architecture for managing business rules using knowledge engineering techniques in RDBMS environment. Proceedings of the Fourth Australian Joint Conference on Artificial Intellligence, 384–393.

[34] Krovvidy, Business Rules for Automating Business Policy, AAAI Technical Report WS-99-09.

[35] Kestutis, Repository for Business Rules Based IS Requirements, Department of Information Systems, Kaunas University of Technology

[36] Muehlen, Modeling languages for business processes and business rules: A representational analysis, Howe School of Technology Management, Stevens Institute of Technology

[37] Rule Engine Performance - http://www.infoworld.com/t/business-rule-management-systems/worlds-fastest-rules-engine-822?page=0,1(viewed 31-7-2012)