ERDSS: Emerging Risk Detection Support System

2008 Project Report

Roos Groeneveld
Don Willems
Jeen Broekstra
Willie van den Broek
Jan Top

Report 1100

Colophon

Title	ERDSS: Emerging Risk Detection Support System
Author(s)	Roos Groeneveld, Don Willems, Jeen Broekstra, Willie van den Broek, Jan Top
AFSG number	
ISBN-number	
Date of publication	January 2009
Confidentiality	
OPD code	
Approved by	

Agrotechnology and Food Sciences Group
P.O. Box 17
NL-6700 AA Wageningen
Tel: +31 (0)317 475 024
E-mail: info.afsg@wur.nl
Internet: www.afsg.wur.nl

The quality management system of Agrotechnology and Food Innovations b.v. is certified by SGS International Certification Services EESV according to ISO 9001:2000.

Content

1 Introduction

This report is about the ERDS System as developed by Wageningen UR. This system is an emerging risk detection system, not an Early Warning System. An Early Warning System deals with the process of continuously monitoring parameters in food (related) chains and looking for violations of (legislative) thresholds. Emerging Risks detection, on the other hand, focuses on detecting risks related to food consumption that will emerge in the near or far future. The process of detecting risks requires the state-of-the-art of food safety research and knowledge on how food is processed, where it is obtained from, how it is processed, etc.

ERDSS stands for Emerging Risk Detection Support System. The system has as goal to detect emerging risks; risks that are not a risk yet, but that might be a threat for human health in the (nearby) future. The main goal of the system is to support experts and managers in their work in detecting and reducing emerging food safety risks. ERDS System applies the holistic approach that for example has been developed in EU projects Periapt, SAFE FOODS and EMRISK.. The holistic approach [Noteborn et al., 2005] enables the emerging risk detection system to take various expertise-fields and disciplines into account when detecting emerging risks. One can, for example, think of economics, international trade, climate changes and human factors next to knowledge about certain supply chains, areas of distribution, production chains and knowledge about breeding and plants. Thanks to the use of information from within these various expertises and disciplines, the detection of emerging risks can go beyond the constraints of these expertises and disciplines, by which potential risks can be found that otherwise would be missed.

One of the new qualities of the ERDS System is that the system is able to reason about the facts it knows about. After adding new facts, application of inference rules will create new (inferred) facts. Some of these new facts will define new emerging risks and these new facts will show up as emerging risks in the system. The ERDS System will alert the user of the system to possible risks at an early stage. In this way, action can be undertaken to prevent the development of possible risks. As the system searches for possible emerging risks by itself, in contrast with reactive systems where users have to give focussed input like a specified search command, it can be said that the system works pro-active.

In this report the ERDS System as developed by de Wageningen UR will be described. First we will outline the envisioned use of the ERDS System within the community and the possible users of the system (chapter 2). In chapter 3 the technical construction of the system and the multifarious segments, which constitute the system, will be expounded. Because the success of the ERDS System depends for a large part both on the quality and on the quantity of the content, the content currently incorporated in the prototype system will be described in a separate chapter (chapter 4). The mode of operation as it has been incorporated in the prototype system today, will be depicted in chapter five. Chapter six will bring the other chapters together by presenting the current prototype of the ERDS System. We will conclude this report with a discussion of the current system.

2 Envisioned Use of the ERDS System

The ERDS System can be of special help answering policy questions of the policy makers. By making it possible to fill-in specific scenario's in the ERDS System the consequences of certain choices can be analysed. Next, the ERDS System will, proactively, give warnings for possible emerging risks. Therefore, the system is particular helpful by formulating unasked advice by the policy support department to policy makers. Also, the system can be helpful for research and development. For example, researchers can define what information has been researched and they can examine scenarios and result of the ERDS System and use these as a guideline for future research.
An interesting option is to use automatic data and text mining techniques to add content (facts) to the ERDS System.

3 ERDSS Reasoning

The Emerging Risk Decision Support System supports the detection and evaluation of perceived emerging risks in the food chain by automatic reasoning. A knowledge based system (or expert system) tries to reproduce the reasoning done by human experts to infer new facts previously unknown (to the system). A working knowledge based system needs three basic types of content for reasoning; concepts, facts, and rules. This content information needs to be supplied and/or validated by human experts.
In this chapter we will give an overview of the reasoning strategy employed by the ERDS System.

3.1 Concepts, Facts, and Rules

Concepts, facts, and rules are the content that a knowledge based system needs to be able to make relevant deductions.

Concepts (also known as instances or things) are things like 'the Atlantic Ocean', 'the Yukon river', 'salmon', 'open net cages', and 'salmon filet'. Not only physical things are concepts, but also processes (e.g. 'salmon fishing in the Yukon', or 'transportation to the supermarket'), hazards, and aspects. Concepts can be applied to actual individual specimen of salmon, or a specific transportation event between China and the United States, but in the context of food safety, this approach is not very helpful. In ERDSS, therefore, concepts mostly refer to a collection of specimen, for instance 'salmon in the Yukon river', meaning all salmon specimen living in the Yukon river.

Facts give meaning to concepts. The information contained in concepts only gives us the information that something exists. The concept 'Atlantic ocean', only tells us that there exists something that we refer to with the name Atlantic ocean. It does not tell us what its function is in relation to other concepts, nor does it even tell us what it is. (Of course, we as humans might say that the Atlantic ocean is an ocean, but that involves reasoning with respect to the name, or prior factual knowledge about the concept). With facts we add information to concepts. Facts are for instance, 'the Atlantic ocean is an ocean', 'salmon is contained in an open cage net', 'the Atlantic ocean has a volume of 354,700,000 km^3', or 'an open cage net is located in Golfo de Coronadas'. Facts adhere to the structure: subject-predicate-object. In the fact: 'the Atlantic ocean is an ocean', the 'Atlantic ocean' is the subject, the predicate is 'is an' and 'ocean' is the object. Predicates are also often called properties '354,700,000 km^3' is the value of the property 'volume' for the concept 'Atlantic ocean'.

Inference Rules are used for reasoning. Concepts and facts represent the knowledge that is present in a knowledge based system, but they do not enable the system to reason about those concepts or facts. Rules are used to combine facts that are present in the system and create new facts. Rules are often represented in 'if-then' form. For instance: '*if* country A borders country B *then* country B borders country A', or '*if* a contamination X is present in river A *and* river A flows

out into lake B *then* contamination X is present in lake B'. The facts in the antecedent of the rule are premises to the consequent (another fact) of the rule. In:

if a contamination X is present in river A
and river A flows out into lake B
then contamination X is present in lake B',

the premises are:

- X is a contamination
- A is a river
- X is present in A
- B is a lake
- A flows out in B

where A, B, and X are variables that represent concepts.
The consequent of this rule is the fact:

- X is present in B

Not only new facts, but also new concepts can be the consequent of a rule. For instance the concept of a new contamination may be the consequent of a rule that states that every electronic factory has a flame retardant hazard.
Concepts and facts represent the information that is currently available to a knowledge based system. By using rules the system can reason about the available information and deduce new information in the form of facts or concepts, which is the process that we call reasoning.

3.2 Reasoning

Reasoning in knowledge based systems is the application of rules on the information (concepts and facts) that are available to the system. In principle, two methods of reasoning can be used: forward chaining and backward chaining.
With **backward chaining**, the system starts with a goal and tries to work back from the goal to see if the premises for that goal are satisfied. Such a system would search the rule base (in which all the rules are contained) until it finds a rule where the consequent is equal to the goal. So for instance: if the user wants to know if salmon in Gulfo de Coronadas is contaminated with flame retardants, the system would start (if the fact is not already present as a fact in the knowledge base) with searching for rules where the consequent satisfies the goal. The rule: '*if* a contamination X is present in water area A *and* organism Y is living in water area A *then* contamination X is present in Y' satisfies the goal if 'X' is the concept: flame retardant contamination and 'Y' is the concept: 'salmon in Golfo de Coronadas'. The system then tries to fulfil the premises to the rule:

- X is a contamination
- A is a water area
- X is present in A
- Y is an organism
- Y lives in A

Suppose it knows that 'flame retardants' is a contamination, that 'Golfo de Coronadas' is a water area, that 'salmon in Golfo de Coronadas' is an organism, and that 'salmon in Golfo de Coronadas' lives in 'Golfo de Coronadas'. Only the fact whether 'flame retardant contamination' is present in 'Golfo de Coronadas' is unknown to the system. The system then adds this fact as a new goal to be satisfied. It then might try to satisfy another rule: '*if* a contamination X is present in river A *and* river A flows out into water area B *then* contamination X is present in water area B', and it starts to check whether the premises to this rule are satisfied. Optionally, such a system may ask the user if a certain fact that needs to be satisfied is true. If all premises to the rules used in the backward chaining process are satisfied, the system may answer the user that the initial goal is achieved.

It is important to note that backward chaining is initiated by a question from the user. The goal of the process is to answer that question. Backward chaining is therefore also known as goal driven.

On the other hand, **forward chaining**, which is the method used in the current ERDSS prototype, is data driven. A forward chaining reasoning system starts its reasoning process when new facts are added to the knowledge base. The system then tries to find rules which premises are satisfied by the facts which are known to the system. If the premises of a rule are satisfied the consequent is applied. If the facts:

- 'Golfo de Coronadas' is a water area
- 'Salmon' is an organism
- 'Salmon in Golfo de Coronadas' lives in 'Golfo de Coronadas'

are known to the system, and the facts:

- 'flame retardant contamination' is a contamination
- 'flame retardant contamination' is present in 'Golfo de Coronadas'

are added to the system by a user, then after applying the rule:

- '*if* a contamination X is present in water area A *and* organism Y is living in water area A *then* contamination X is present in Y'

the new fact:

- 'flame retardant contamination' is present in 'salmon in Golfo de Coronadas"

is added to the knowledge base. The system then tries to find other rules whose premises are now satisfied (after the addition of the *inferred* fact) and applies them.

After new facts are added to the knowledge base, whether it is done by an expert user, or it is the consequence of automatic application of rules by the system (reasoning), the system tries to infer new facts by repeated application of the rules in its rule base.

Both methods of reasoning have their uses for ERDSS. Forward chaining is especially suited for monitoring, where the system is proactive and tries to find emerging risks without intervention by the user. Backward chaining might be used for identifying facts that may lead to signals from observed symptoms that are as yet unknown. The system might then request the user for more information (whether a specific fact that may lead to knew emerging risks being found is true or not).

In the current demonstration prototype, which implements a monitoring system, forward chaining is used. In the future version of the system, both methods will be implemented in a hybrid system.

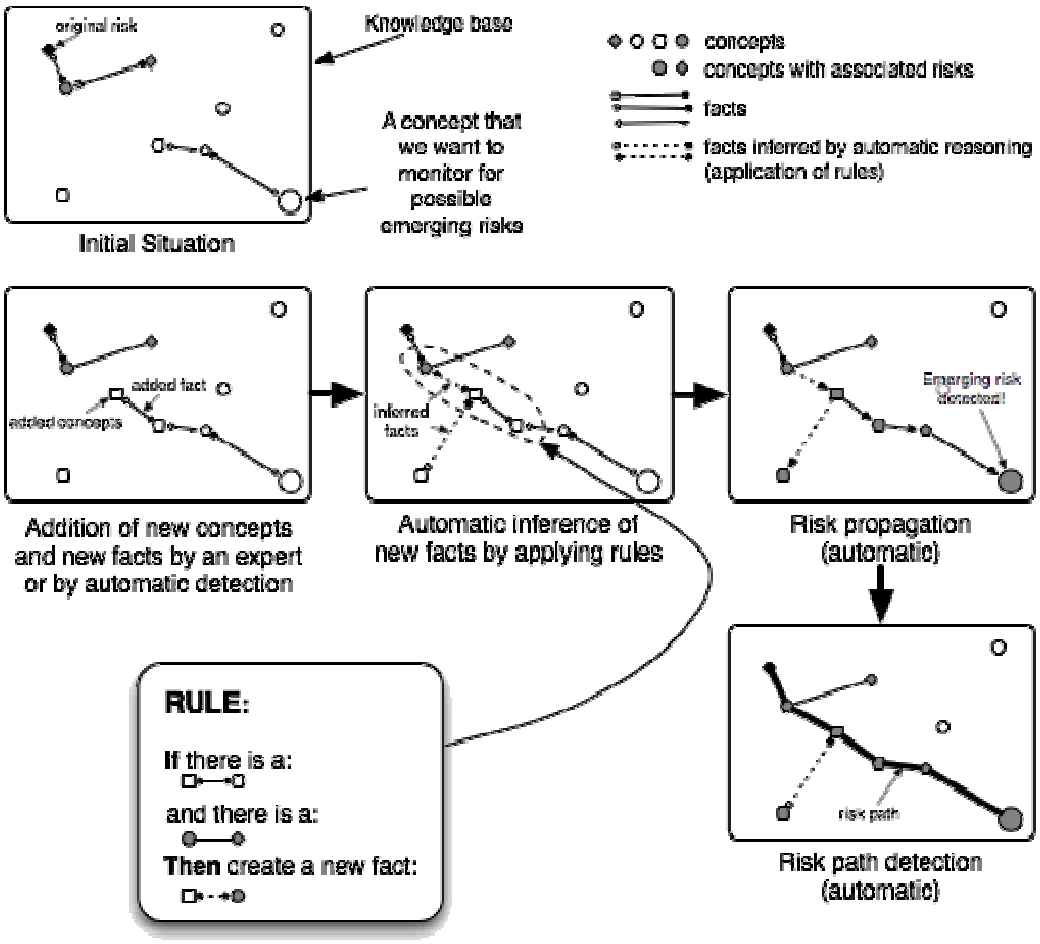

Figure 3.1. Reasoning in the ERDS System. Initially the knowledge base consists of a (large) number of concepts and facts. Concepts are connected to each other by facts, and some of them may have risks associated with them. When the user adds a fact (or concept), the system starts its reasoning and infers new facts. Inferred facts may lead to new (or existing) risks to be associated with concepts. If the concept is being monitored (because it is for instance a food product in The Netherlands), a new risk path is detected.

3.3 Risk Paths

One essential part of the ERDSS monitoring system is the ability to evaluate the emerging risks that are found by the system. The user has to be able to judge the decisions made by the reasoning system. The user might then decide whether or not an emerging risk is realistic and

needs acting upon. The user needs access to the reasoning path as found by the reasoning system, e.g. the facts (the premises) and rules that led to the emerging risk.
This means that the system needs to retain per inferred fact, references to the facts that are used as premises and the rule that led to the inferred fact. With that information the reasoning path for each inferred fact, and therefore for each emerging risk that is found, can be recreated and presented to the user of the ERDS System.

4 Incorporated Content

The quality of a knowledge based system's abilities depends for a large part on the content incorporated in the knowledge base and in the rule base. It is obvious that the quality of the content is very important. If facts are incorrect, or when rules are to general or to specific, the deductions made by the system will be of low quality. It is therefore of prime importance to safeguard the quality of the facts and rules being added to the system. The number of facts and rules also plays an important part in the validity of the emerging risks being found. If the number of facts or rules is too small, not all relevant risks will be found. In this chapter we will describe the content (facts and rules) that are currently incorporated in the prototype of the ERDS System. The number of facts and rules incorporated in the prototype is limited at this moment, as we were focusing on the development of the system itself, in order to provide a proof of concept.

4.1 Ontology

An ontology in computer science is a formal representation of a set of classes of concepts and relations between those classes. It constitutes an (often) hierarchical description of a domain. For ERDSS, an ontology is important because it allows for reasoning on classes of concepts instead of on the concepts themselves. For instance: 'salmon in the Bering Sea' is of class 'Salmon'. According to the ERDSS ontology the class 'Salmon' is a subclass of class 'Fish', which in its turn is a subclass of 'Organism', therefore 'salmon in the Bering Sea' is also of class 'Fish' and of class 'Organism'. This allows us to use both general rules like '*if* a contamination X is present in water area A *and* organism Y is living in water area A *then* contamination X is present in Y' and more specific rules like '*if* Salmon X is located in Bering Sea *then* Salmon X is fished by Fishing Boat Y'. The first rule would apply on both salmon living in the Bering Sea and whales living in the Bering Sea, while the second rule would only apply to the salmon.

An ontology not only defines the classes that can be used, but also the connections between those classes (properties). These properties are often restricted in the classes they connect. For instance, the property 'sub-area-of' only makes sense in the context of the 'Area' class.

The ERDSS ontology contains classes from different fields, it contains a (partial) tree of life, an ontology for geographic features, and classes more specific to ERDSS, like Process, Aspect, ObjectInContext, and hazards. Instances of the Hazard class ('Melamine hazard' for instance) are connected to an aspect (an instance of the 'Aspect' class, for instance 'Melamine contamination'). This aspect may in turn be connected to a resource (like 'salmon in the Bering Sea'). In this manner, hazards are connected to resources. Aspects do not have to be connected to hazards. The corruption of a country for instance, is also an aspect of a resource. All these connections are facts, and instances of classes are concepts.

ObjectInContext is another imported class. It defines objects (like resources, ore processes) as related to a context (for instance a geographic location).

4.2 Concepts, Facts, and Rules

The ERDS System is a system which will grow in power with the passing of time. The more content the system contains, the more powerful it will be, as possible risks can be evaluated better, because more areas of expertise are included.

In the current prototype of the ERDS System certain information, content, has been included and processed. In an earlier stage of the project, the decision was made what content would first be included in the system. Decided was to focus on the fish chain (and more specifically in the salmon chain), for various reasons. First, in this sector especially, many problems appeared throughout the production chain. Second, this sector deals with a fast growing market, so many changes occur, which often lead to the creation of new problems. Third, this project is one segment of four projects within the Wageningen UR and one of the other projects focuses on the salmon chain [van der Roest et al., 2007]. Because the focus of that project is on extracting knowledge about the salmon chain, knowledge developed by that project constituted a great opportunity to add that knowledge to the current ERDSS prototype. In this way, the focus of the ERDS System prototype came to lay on the salmon chain.

In this project the salmon chain was divided in the fish feed chain, in the salmon production chain and in the salmon processing chain. For these three areas of expertise, information and knowledge were extracted from several sources. Reports from RIKILT, IMARES for salmon chain information and for more general information several sources available on the internet were used.

Besides specific information on fish, it was required to add knowledge from other fields. Geographic information, logistical information and 'general' information were added to the knowledge base as well.

When adding new information in the ERDS System, certain annotations are always made to make it possible to verify certain facts. This includes information about the source of the information (report/interview/journal paper/etc.), the date on which the fact or rule has been added or modified, and the person or institute that added the item. The addition of these annotations enables the expert who is evaluating the emerging risks found by the system to better ascertain the reliability of these risks.

5 Mode of Operation

In the current prototype of the ERDS System, a few specific functionalities were implemented. These functionalities are to a large extend developed upon the requirements and wishes of some of the intended users of the system. The requirements of the intended users will keep evolving, so this outline is limited to the current version of the ERDSS prototype. In chapter seven various possible new features and functionalities will be discussed, which might get developed in the future. In this chapter the recent methods of working with ERDSS will be outlined.
In the prototype of the ERDS System the user has two main possibilities; monitoring current emerging risks and working with what-if scenario's. When a user starts the ERDS System, a screen with a list of all new emerging risks will appear. The user can verify and inspect these risks, which is what is referred to as monitoring. When the user selects a risk, he or she can:

- see how the risk has been built up
- judge the degree of threat from the risk on human health
- revise and edit the risk, so alternative situations can be explored as well.

Besides monitoring the up to date, current emerging risks, it is also possible for a user to create what-if scenario's. This can be done by changing existing emerging risks or by creating a complete new scenario. Either way, the user can influence the facts that form an emerging risk in these what-if scenario's in various ways. With the what-if facts the user can:

- add real and fictitious ('play') facts
- remove facts
- change facts

As can be seen, there is an overlap in the functionalities that are available with both monitoring and working with what-if scenarios. In both situations one can edit risks, resulting in alternative situations coming forward. As the objective of each procedure is different, this distinction is made to clarify the possibilities for the user of the ERDS System.

6 Prototype

In this chapter we will describe the capabilities of the demonstration prototype that was developed in 2008. In our description of the prototype we will distinguish between the back-end (the reasoning system) and the front-end (the user interface).

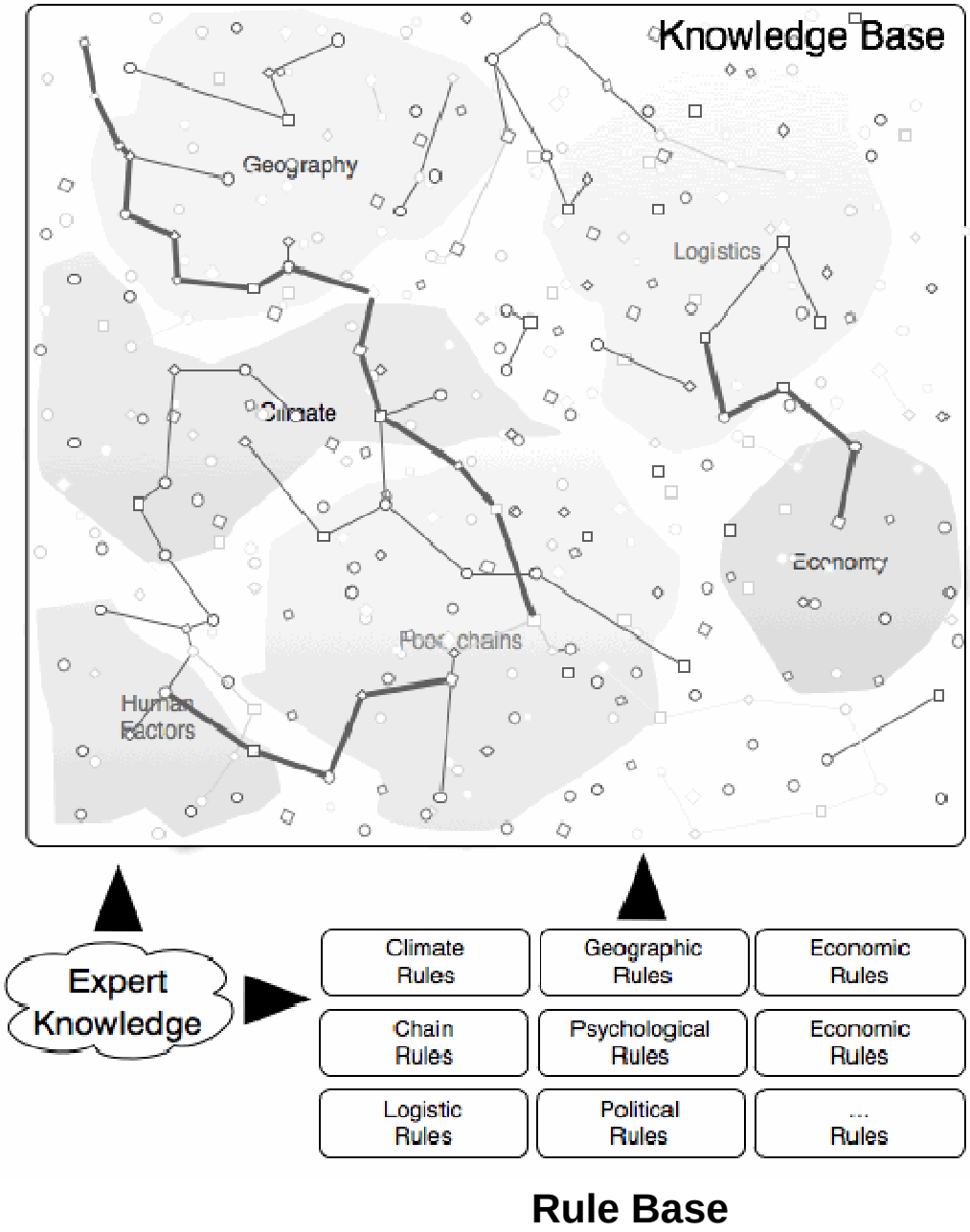

Figure 6.1. The back-end of the ERDS System consists of a knowledge base implemented by the Sesame data base, a rule base and a rule engine (not shown), both implemented by the JESS rule engine (see text).

6.1 Back-end

The back-end of the system contains the knowledge base, the rule base, and the reasoning engine (see Figure 6.1). The knowledge base contains the concepts and the facts that are present in the system. When the system starts up, the facts are loaded into the knowledge base. In the prototype we use Sesame [Broekstra et al., 2002], which is an open source framework for storage, inferencing and querying of Resource Description Framework (RDF) data. RDF [Beckett, 2004] is a set of specifications from the World Wide Web Consortium (W3C), which is used as a general method for resource description. In RDF facts are represented as triplets of the form <subject, predicate, object>. Concepts and facts in ERDSS can easily be described using RDF. The other parts of a knowledge based system, the rule base, and the rule engine, are implemented using the JESS [Friedman-Hill, 2008] rule engine, which is one of the fastest Java-based rule engines available. When new facts are added to the Sesame knowledge base, the rule engine (JESS) starts to reason using the rules that are available in the rule base. If new facts (or new concepts) are inferred, those facts (or concepts) are added to the knowledge base. The front-end is notified by any addition (or retraction) of facts from the knowledge base. Furthermore, the back-end keeps track of all rules that are fired, which premises led to the rules being fired, and the consequences (inferred facts) of those rules that were fired. This enables the ERDS System to recreate the reasoning path for any inferred fact and therefore the risk paths for all emerging risks.

6.2 Front-end

The front-end or Graphical User Interface (GUI) presents the emerging risks to the user and provides a simple interface for the addition of new facts. The GUI consists of one window (see Figure 6.2) containing a menu bar with which one can add new facts, add RDF files containing multiple facts, or add predefined facts to the knowledge base. Below the menu bar, three panels are visible. The left panel presents different views that are available to the user. The middle panel presents the emerging risks that were found by the system and the rightmost panel presents the risk path (hazard chain) or a geographical map of the selected risk path (hazard map).

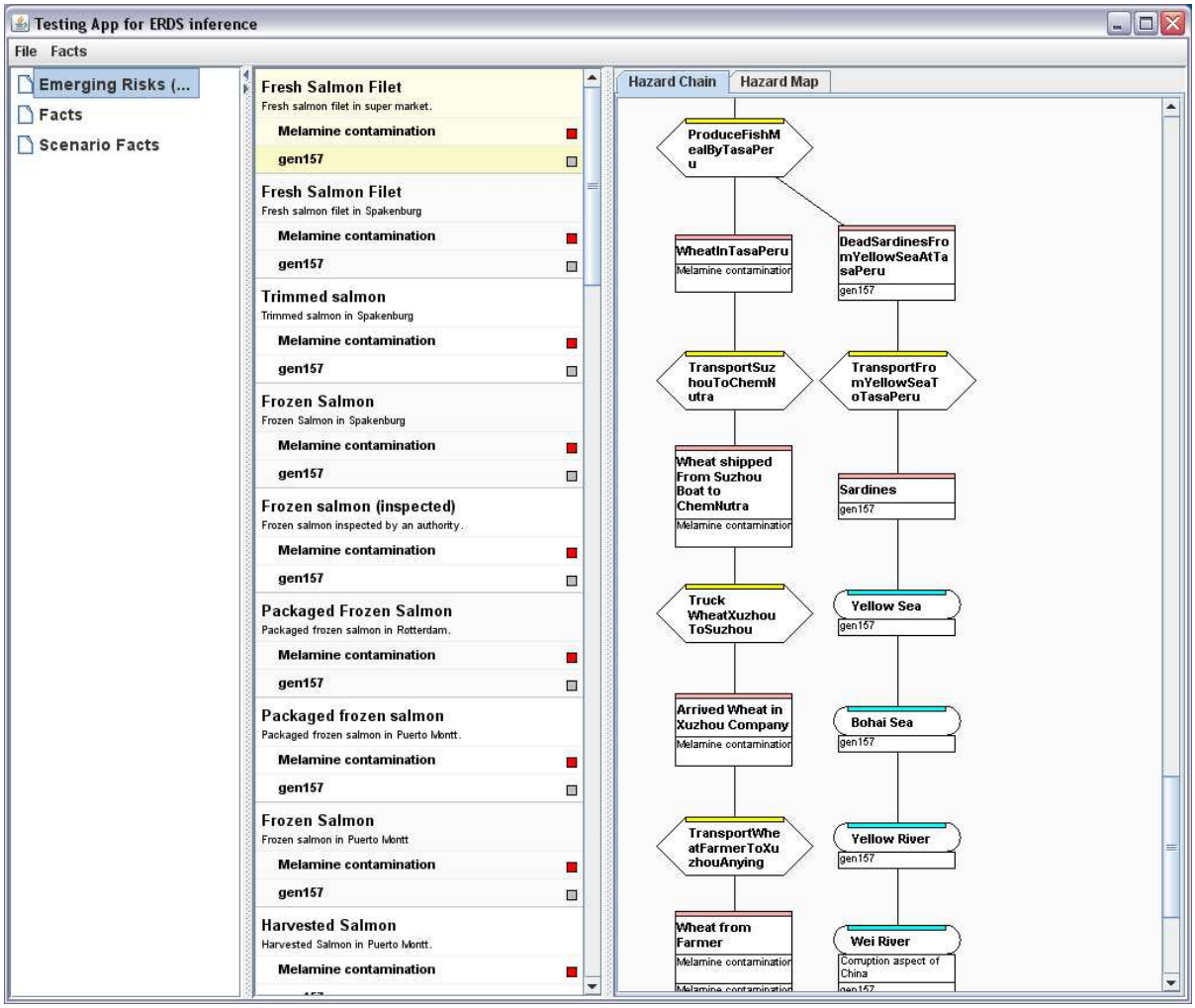

Figure 6.2. The main window of the ERDSS prototype. In the left panel, the user can select a view on the data. In the middle panel, all concepts with emerging risks are presented, and in the right panel, the risk path can be evaluated.

One view that is available to the user (and which are visible in the left panel) is the main monitoring view identified by the label 'Emerging Risks' concatenated with the number of emerging risks found by the system. A 'Facts' view, which presents all facts to the user, is also available. Finally, a 'Scenario Facts' view can be selected in which the user can add 'play facts'. In this manner, the user can evaluate what consequences the addition or retraction of facts has on the emerging risks that are found. For instance if a user does not agree that a fact known to the system is true, the user might retract the fact and see whether that retraction has any consequences on the emerging risks that are found. This functionality will be implemented in a future version of the system.

In the middle panel (see Figure 6.3), the list of emerging risks is presented. In the current version of the prototype, all concepts with attached risks are shown in this list. In a final version of the

system, filters should be applied to this list so that users see only the risks that they are interested in (for instance only micro-biological hazards in The Netherlands).

Figure 6.3. The list in which emerging risks are presented to the user. Each concept associated with a risk is represented in the list. Below each concept is the associated risk and an evaluation of the severity of the emerging risk. Red indicates a very serious risk, orange is less serious and yellow is the least serious. Grey denotes a severity that is unknown.

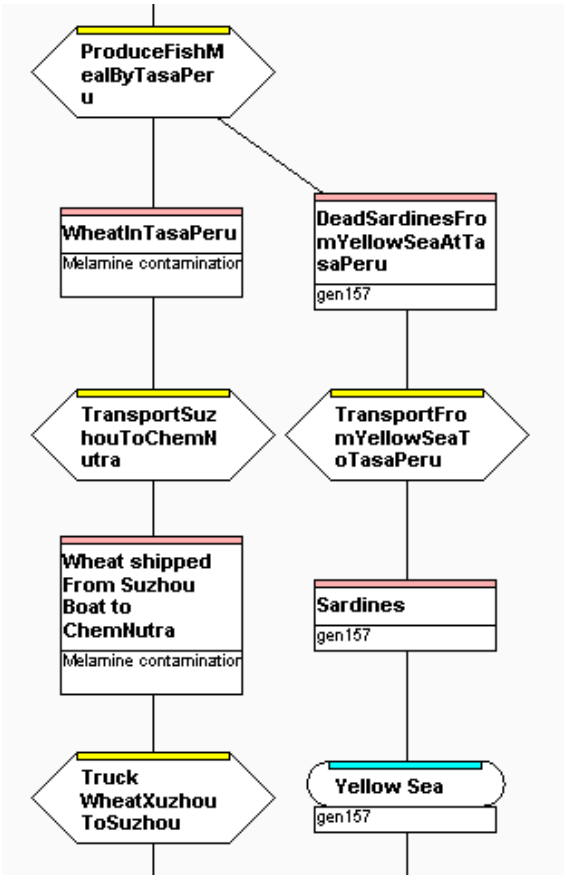

Figure 6.4. An example of a risk path as it is presented to the user. Different types of concepts are displayed differently. Physical concepts are presented in a rectangle, processes are displayed in a hexagon, and geographical concepts in an oval.
The direction of the risk path is from bottom to top. The concept with the initial risk is at the bottom, while the relevant concept is at the top.

When the user selects a risk in the list, a graph of the risk path (or hazard path) is shown in the rightmost side of the window (see Figure 6.4). The upper item in the graph is the risk that was selected in the emerging risks list. The second item in the graph is the item that led to the risk being associated to the concept selected in the list. This continues until the item on the bottom of the graph, which is the item in which (part of) the risk originated, is reached. In this manner one can see how the emerging risk propagated along the different concepts.
If one clicks with the mouse on one of the risks in the risk path, that risk is selected and a window specifying the rule that led to the selected risk pops up (see Figure 6.5). The selected risk is encircled in red (as being the consequence of the rule) and the premises are encircled in cyan. The rest of the risk path is greyed out. In the rule window, a human readable representation of the rule is presented together with the premises that led to the rule being used, and the consequences of the rule are specified.
The user can also add facts to the system. The 'Facts' menu in the menu bar provides options to add predefined facts (used to test the system) or to add a new concept. If the user chooses to add a new concept, the window depicted in Figure 6.6 pops up. In this window the user is initially presented with a text field in where she/he can provide a label for the new concept. A combo box (a text field with a drop down menu) is also presented in which the user can select a type for the new concept. This combo box uses text completion to enable faster selection of the requested type by the user. Only classes defined in the ERDSS ontology can be selected as type. When the user has selected a type, new text fields appear, enabling the user to provide values for the properties relevant for the selected type. Some properties require a literal value (a string or a number), and other properties require a reference to another concept. If a reference to a concept needs to be specified, the user is again presented with a combo box with text completion. The user can only select references to concepts that are already present in the knowledge base.
After the user has clicked the 'Add Fact' button, the concept and all defined facts are added to the knowledge base, and the system starts reasoning, which may lead to new emerging risks being detected.
To enable the user to evaluate the emerging risks that are found, source information such as the document from which the information (facts or rules) was extracted, the expert who added the information, and the date when the information was added to the system are also available to the user. Source information for facts are presented in the 'Facts' view of the user interface (select 'Facts' in the left panel in the main window), and source information for rules are presented in the rule pop-up window (see Figure 6.5).

The current prototype allows the user to monitor for emerging risks and evaluate the risks that are found by reviewing the risk path and the rules that fired resulting in the risk path. Furthermore the user can add facts enabling her/him to evaluate the consequences of the addition of unknown facts to the knowledge base.

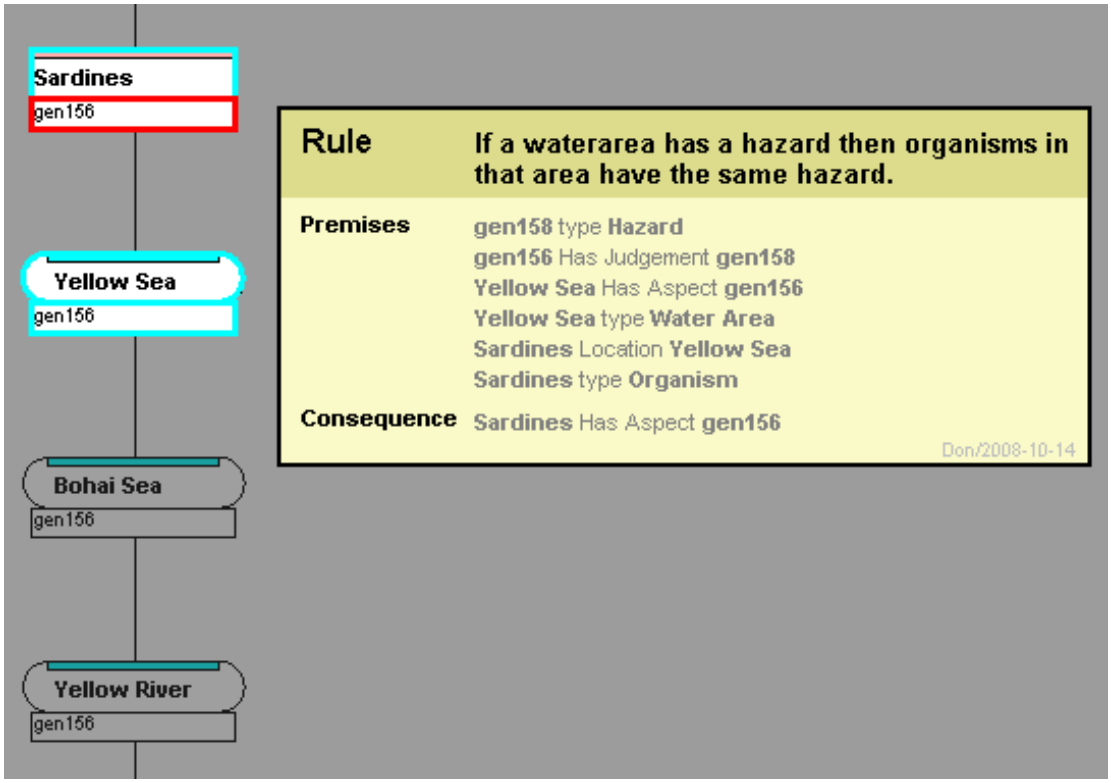

Figure 6.5. This is an example of the selection of a risk, showing the rule that led to the fact that the risk is associated with its attached concept. In the pop-up window the human readable description of the rule, the premises that led to the rule being fired, and the consequence of the rule are presented. In the risk path the consequence is encircled in red and the premises are encircled in cyan.

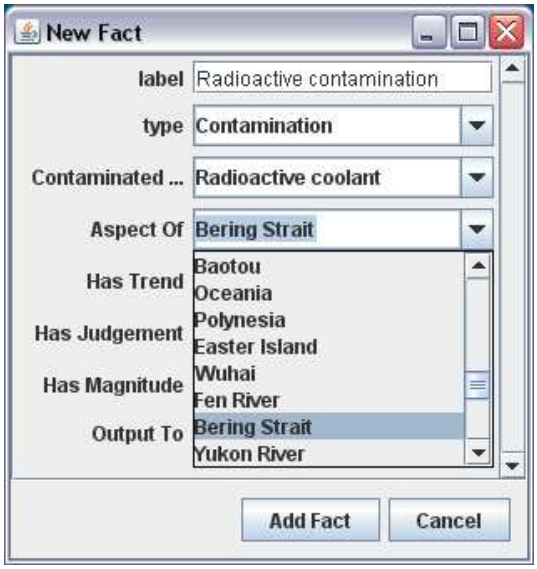

Figure 6.6. The 'New Fact' window in which the user can add facts to a concept that is created. The text fields, all implement text completion, which allows the user to quickly select the relevant information that is valid for the specified property.

7 Discussion

With the current ERDSS prototype, we have shown that a decision support system for emerging risks in food, can work. A self-reasoning, pro-active system has been achieved. With this prototype we are able to monitor emerging risks, evaluate the reasoning done by the knowledge based system, and test the system by adding new facts into the knowledge base. The possibilities of such a system in supporting policy multiply when new content is added to the system, which will make ERDSS essential for recognising future risks to our food chain.

One caveat in the prototype is that it only supports the salmon and salmon feed chain at this moment. For a usable ERDS System, the content needs to be expanded considerably. Not only with respect to food chain knowledge but also with respect to knowledge in other domains such as the economic domain, the geographic domain, the logistic domain, and the human factors domain. Some of this knowledge may be added automatically from available databases. Geographic information, for instance, is readily available in databases and can be transformed in meaningful data for the ERDSS domain.

The number of rules also needs to be expanded. Rules may for instance be used to automatically create new food chains at specific locations, removing the need for experts to add detailed information about every possible food chain in every possible location, which would be an impossible task. At this moment rules are developed for the automatic generation of new salmon chains in different locations. The inclusion of these rules would allow a user to create a new chain on the fly by simply adding the concept of a few salmon eggs in a river where no salmon chain existed before.

Another useful addition to the prototype would be the inclusion of filtering. Users may want to restrict the emerging risks presented to them by location (only emerging risks in The Netherlands) or by speciality (only emerging risks due to chemical contaminants). Filtering would undoubtedly add value to the ERDS System.

An open issue at this moment is the handling of quantitative or probabilistic data. Possibilities for incorporating quantitative data will be researched in the near future.

In our opinion these modest though important steps will lead to a valuable detection system that supports and stimulates the correct recognition of future food safety risks.

References

[Beckett, 2004] Beckett, D. (editor) (2004). RDF/XML Syntax Specification (Revised), W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

[Broekstra et al., 2002] Broekstra, J., Kampman, A. and van Harmelen, F. (2002). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Horrocks, I. and Hendler, J., editors, Proceedings of the first International Semantic Web Conference (ISWC 2002), number 2342 in Lecture Notes in Computer Science, pages 54–68, Sardinia, Italy. Springer Verlag, Heidelberg Germany. See also http://www.openrdf.org/.

[Friedman-Hill, 2008] Friedman-Hill, E.J. (2008) Jess®, the Rule Engine for the Java™ Platform, Version 7.1p2 (5 November 2008), Sandia National Laboratories, http://www.jessrules.com/jess/docs/71/

[Noteborn et al., 2005] Noteborn, H.P.J.M., Oom, B.W., and de Prado, M. (2005) Emerging Risks Identification in Food and Feed for Human Health, VWA-Food and Consumer Product Safety Authority, Directory of Research and Risk Assessment, The Hague, The Netherlands.

[van der Roest et al., 2007] van der Roest, J., Kleter, G., Marvin, H.J.P., de Vos, B., Hurkens, R.R.C.M., Schelvis-Smit, A.A.M., and Booij, K. (2007) Options for pro-actively identifying emerging risk in the fish production chain. RIKILT Institute for food safety, Wageningen UR, Report 2007.006, Wageningen