Expertise-based decision support for managing food quality in agri-food companies

Patrice Buche, Bernard Cuq, Jérôme Fortin, Clément Sipieter

Expertise-based decision support for managing food quality in agri-food companies

Patrice Buche*[1], Bernard Cuq[1], Jérôme Fortin[1], and Clément Sipieter[2]

[1] *INRA, Montpellier Supagro, University of Montpellier, IATE, France*

*Corresponding author: patrice.buche@inra.fr, cuq@supagro.fr, fortin@umontpellier.fr

[2] *CNRS LIRMM, University of Montpellier, France*

clement.sipieter@lirmm.fr

Abstract

In many agri-food companies, food quality is often managed using expertise gained through experience. Overall quality enhancement may come from sharing collective expertise. In this paper, we describe the design and implementation of a complete methodology allowing an expert knowledge base to be created and used to recommend the technical action to take to maintain food quality. We present its functional specifications, defined in cooperation with several industrial partners and technical centres over the course of several projects carried out in recent years. We propose a systematic methodology for collecting the knowledge on a given food process, from the design of a questionnaire to the synthesis of the information from completed questionnaires using a mind map approach. We then propose an original core ontology for structuring knowledge as possible causal relationships between situations of interest. We describe how mind map files generated by mind map tools are automatically imported into a conceptual graph knowledge base, before being validated and finally automatically processed in a graph-based visual tool. A specific end-user interface has been designed to ensure that end-user experts in agri-food companies can use the tool in a convenient way. Finally, our approach is compared with current research.

Keywords. Knowledge acquisition, knowledge extraction, knowledge representation, conceptual graphs, decision support systems

1 Introduction

In many agri-food companies, food quality is often managed using expertise gained from experience. For example, cheese-making chains that showcase their terroir are an economically and agriculturally important industry in France, there being around 17,900 milk producers, 1,290 farm producers and 432 processing companies. Cheese-making companies with a "geographical indication", such as the appellation d'origine protégée (AOP) or indication géographique protégée (IGP), market their products by promoting local resources produced in their terroir and communicating their expertise in terms of milk production and processing. Internal evolutions to appellations, especially in terms of turnover and difficulties encountered in the training of operators, greatly weaken the preservation and transmission of this expertise. This kind of problem is not restricted to cheese-making companies that showcase their terroir. In other agri-food companies, production line management in factories depends to a great extent on the operator's experience. Consequently, overall quality enhancement may come from sharing collective expertise, which includes informal knowledge. Informal knowledge means knowledge that has not been acquired during learning classes, but rather through individual intentional or fortuitous experiences.

In this context, the development of knowledge engineering methods allowing knowledge bases to be exploited opens up new perspectives in terms of the preservation and data management of operational experience, by proposing complex automatic reasoning technics that go well beyond the description of standard processes [Buche et al., 2013a, Aceves Lara et al., 2017].

In this paper, we propose an original and complete methodology, as well as a dedicated software, for collecting formal and informal knowledge from operators and experts, collectively validating this knowledge, and codifying it in a well-founded language based on a core ontology that provides decision support. This decision support system (DSS) helps to control quality [1] and

[1] Food quality is the set of characteristics of food that are acceptable to consumers.

defects [2] of manufacture by recommending the most relevant technical action to take at the processing process level, with this process made up of several unit operations [3]. The DSS also allows all the defects and qualities impacted by a given action to be determined. These recommendations are based on formally representing the possible causal relationships linking defects/quality standards to actions by way of explanatory mechanisms.

Another type of application that this system could be put to is for training purposes. For example, it could help a new operator to get an overview of all the operations and get a better understanding of the different kinds of modifications that can be made to control a process (referred to here as levers).

A generic methodological approach for managing the different steps in the DSS design and implementation process has been developed in order to allow it to be used in different food environments (see Figure 1). The first step involves defining the scope of the study (a processing process and a set of product quality standards or defects of interest) and collecting associated sources of information (technical reports, etc.). In the second step, the processing process is broken down into a set of unit operations and associated controlled parameters [4]. In the third step, a systematic questionnaire is de-

This includes internal factors (chemical, physical, microbial) and external factors such as appearance (size, shape, colour, gloss and consistency), texture, and flavour.

[2] Food defects are the characteristics of food that are not acceptable to consumers. This includes the same factors as for food quality.

[3] A unit operation is a basic step in a process. Unit operations involve a physical change or chemical transformation, such as separation, crystallization, evaporation, filtration, polymerization, isomerization, and other reactions.

[4] A controlled parameter is the current measured value of a particular part of a process

rived from the description of the process in order to collect expert knowledge on the potential impact that each unit operation may have on the product in terms of defects and quality standards. In the fourth step, expert knowledge is collected through two kinds of interviews: on the one hand, individual interviews, and on the other hand, collective and contradictory ones. Collective interviews are organized in order to resolve potential contradictions detected when pooling the data from individual interviews in order to obtain a consensus. The expert knowledge resulting from these interviews is then represented in the fifth step as a tree structure using mind mapping software. As mind map tools are only equipped with standard scripting mechanisms, our approach in the sixth step involves automatically translating the knowledge from the mind map software into the conceptual graph formalism [Chein and Mugnier, 2009], which allows specific automatic reasoning tasks to be performed.The tool runs on CoGui software, which is a conceptual graph editor that, firstly, permits the terminology, facts, rules and constraints of an application domain in a knowledge base to be managed, and secondly, allows this knowledge base to be queried and reasoned. Finally, the DSS designed in the seventh step is an end-user interface with associated programs based on CoGui API, ensuring that end users of the application can easily use it without knowing anything about conceptual graph formalism. This seven-step workflow is an iterative one, as the processing process and/or the expert knowledge on it may evolve.

which is being controlled. For instance, temperature is a common controlled parameter.

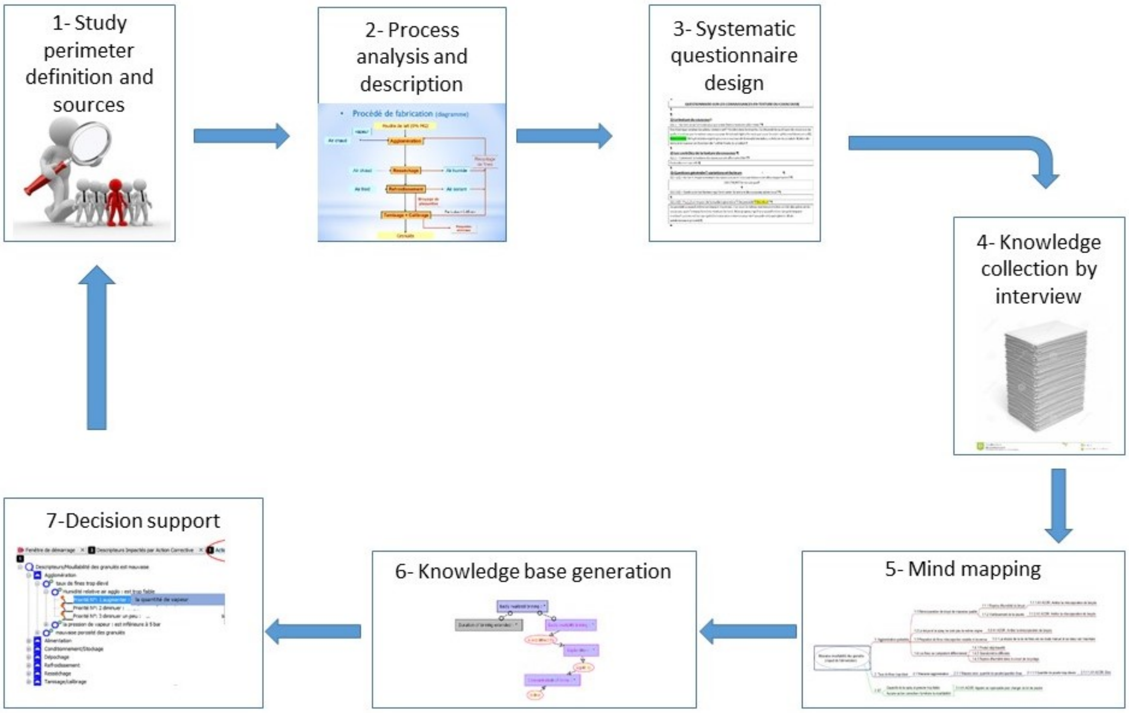

1- Study perimeter definition and sources
2- Process analysis and description
3- Systematic questionnaire design
4- Knowledge collection by interview
5- Mind mapping
6- Knowledge base generation
7-Decision support

Figure 1: Overview of the methodology

The sections in this paper are dedicated to the following topics:

- the functional specifications of the desired system (Section 2),
- the methodology used to collect the expert knowledge on the processing process, and the use of mind mapping to structure knowledge (Section 3),
- the automatic translation of the mind map into the conceptual graph model (Section 4),
- a presentation of the decision support system (Section 5),

- the decision support system validation process (Section 6),

- a comparison with current research(Section 7).

2 Functional specification of the system

The system's features are based on the experience we have acquired over the course of several projects with different industrial partners and technical centres, which are briefly presented here:

- industrial contract (2012-2014) with Panzani (France), for which we had to represent knowledge on durum wheat fractionation in the production of couscous;

- industrial contract (2014-2016) with Regilait (France), for which we had to represent knowledge on the fast hydration of milk powder;

- the CASDAR Docamex project (2017-2020), as part of which we are collaborating with several cheese-making companies that have a geographical indication (AOP or IGP) to develop a generic methodology and DSS which can be used by any company.

Our goal is to create an application that, firstly, allows the vocabulary used to express knowledge within a given community to be defined in a non-ambiguous way, secondly, allows this knowledge to be explored in two different ways, and thirdly, allows knowledge evolution to be managed. Both directions of exploration, from a defect to a corrective action and from a corrective action to defects, are relevant: the first when attempting to identify a possible corrective action in order to fix a defect, and the second when checking for the possible undesired consequences of this corrective action on other defects.

2.1 Definition of unambiguous vocabulary

In order to express common knowledge, domain experts have to share a common vocabulary. Indeed, in many domains, people do not use the same terms to denote the same concepts. To define this common vocabulary, synonyms must be identified. Sometimes, people use the same terms to denote different concepts. These terms must be identified too. Consequently, the system must allow a shared non-ambiguous vocabulary to be defined.

2.2 From a defect to a corrective action

The software is designed to be used in agri-food companies to deliver a recommendation when a defect is detected in a given production chain. A technician involved in the chain may consult the expert knowledge base to find some explanation of what is going wrong and why, and to get some suggestions of actions to solve the problem.

When a corrective action is considered by the technician, the system should display the key information about why this correction may solve the problem. It is important for any technician to have information on why a lever (an adjustment that can be made to control a process) may solve a problem. Technicians with less experience can improve their own knowledge and skills, while experienced technicians can use this information as a checklist to be sure that they have not forgotten anything. A given problem can be explained by a first situation, itself explained by another situation, and so on until the last explanatory situation, which can be corrected using a particular lever that leads to a corrective action. This information can be displayed concisely or with full details. The concise explanation involves providing access only to the first and last situation explaining the problem so that the suggested corrective action can be understood correctly. For more details, all the branches linking the defect by way of intermediate situations to the corrective action can be displayed.

2.3 From a corrective action to defects

Once a technician has identified a given corrective action that would fix a food defect in the process, he/she needs to obtain information on the potential impact of this corrective action on the set of defects managed in the knowledge base. Otherwise, the original problem may just be replaced by another problem.

2.4 Knowledge evolution

No one can ensure that all the possible situations linking defects to corrective action are fully described and understood in any version of the knowledge base. This means that the knowledge base may be updated iteratively throughout the life cycle of the DSS in order to take into account new experiences. Consequently, the DSS needs to include knowledge base maintenance features so that explanation trees can be easily added to or modified. Moreover, analytical values (for example, temperature level, pH value, etc.) or temporal information (for example, sequence of unit operations) can be associated with situations in order to be able to contextualize the querying of the knowledge base.

3 Obtaining and structuring expert knowledge

In this section, we will present the methodological approach that we propose for collecting and structuring expert knowledge.

3.1 Collecting expert knowledge through individual interviews

Several approaches have been proposed for collecting expert knowledge on skills development for training and educating professionals [Piot, 2012], including in the domain of agriculture [Cerf et al., 2011]. These methods, based on professional didactics, have a lot of potential. They are based on making video recordings of the daily operations carried out by cheese makers for each unit operation. When the films are shown to the cheese makers during face-to-face interviews, it allows their explanations of the gestures that they do implicitly to be captured. In our applications, we need to collect information about several dozen situations of interest. Therefore, such methods should be used only for certain complex unit operations, since their main drawback is that they are very time-consuming. Consequently, we took inspiration from [Depraz et al., 2003], who propose a method for interview management. In order to be as exhaustive as possible in collecting knowledge, we designed an interview guide based on a systematic analysis of the process at the unit operation scale. We assume that a preliminary study of the process has been conducted (Step 2 described in Figure 1) in order to identify the levers which may be used to control each unit operation.

For each unit operation, a series of questions has been devised, from the more general to the more specific. More precisely, for each unit operation O_i in the process, each of its associated levers L_{ij}, and for each defect D_k, questions have the following form:

- General question 1: How do you check that operation O_i is being carried out correctly?
- General question 2: Does operation O_i have an impact on the defect D_k?

    - Yes/No/I don’t know

- If yes, can you describe the impact?
- for each lever L_{ij}
    - Specific question 1: Does lever L_{ij} have an impact on the defect D_k?
        - Yes/No/I don’t know
    - If yes, can you describe the impact?
    - Specific question 2: Do you usually modify lever L_{ij} during the process?
        - Yes/No/I don’t know
    - If yes, can you describe how?
- For unit operation O_i and defect D_k, order the levers from the most to least efficient.

Once the series of questions has been established for all unit operations, the interviews can be conducted. A list of people with recognized expertise in relation to the process must be drawn up (for example, line operators, maintenance staff, quality staff, research and development staff, scientific experts, etc.). These people must be interviewed one by one. Each interview, which may last between one and three hours, is recorded in order to avoid losing information.

3.2 Structuring expert knowledge using mind mapping

Once recorded, the interviews are analysed and consolidated in order to create a first version of an explanation tree using mind map software. The root of each tree is a given situation of interest representing a defect, each arc linking two situations is an explanation between them, and the leaves of the tree are corrective actions (see Figure 2).

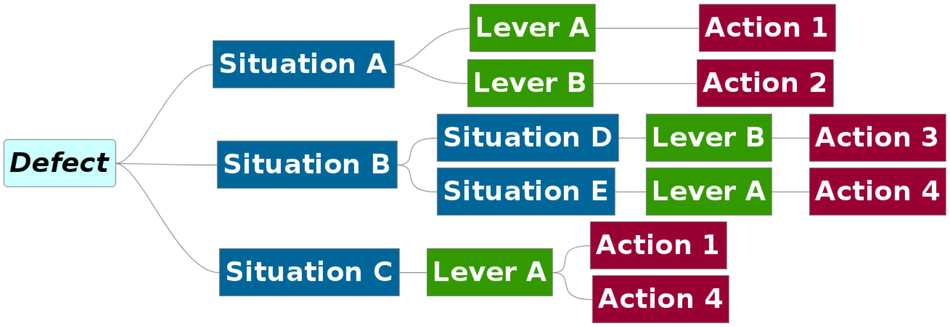


Figure 2: Explanation tree expressed using mind map software

More complex explanations for a situation, called joint effects, must also be taken into account. Joint effects occur when two or more situations at level n must occur in combination if they are to affect a situation at level $n-1$. The effect is expressed in the mind map explanation tree by the creation of an "*AND*" node.

An example of a joint effect is given in Figure 3, which shows how the situation "Physico-chemical composition of cheese on demolding too high" is jointly explained by "Cheese draining level at 20h after demolding too high" and "Hygrometry of the pre-refining cellar too low".

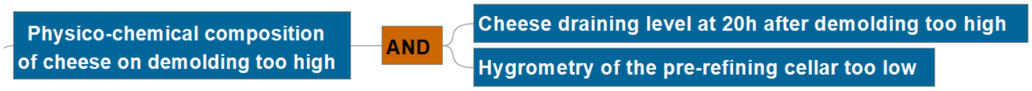

Physico-chemical composition of cheese on demolding too high
AND
Cheese draining level at 20h after demolding too high
Hygrometry of the pre-refining cellar too low

Figure 3: Representation of a joint effect

3.3 Collective interviews

Once the first version of the explanation tree is complete, collective interviews should be organized in order to achieve two different goals. The first is to present the preliminary results of the study to all the interviewed experts. This is an important step towards involving everyone in the success of the developed tool. The second goal is to validate the collected knowledge, and correct as soon as possible any misunderstandings. It is at this step that a lack of knowledge on certain points may be identified. This may happen when two different experts have expressed contradictory knowledge on a given part of the tree. In this case, a collective consensus is sought. In some cases, experiments could be planned in order to acquire new knowledge and resolve the contradiction.

4 From mind mapping to formal knowledge representation

Mind mapping tools are well suited to quickly capturing the experts' knowledge of a process [Buzan, 2004]. However, they are not sufficient for ensuring the consistency of a large data set, as they lack a formal representation model to ensure data consistency. To allow efficient automatic reasoning, the same kind of knowledge must always be represented in the same way, regardless of who inserts the information or when it was inserted. In addition, we need an easy way to avoid duplicate data across several trees, since the same explanatory situations may appear in different trees (concerning different defects). The use of mind mapping tools may results in possible duplication of information. Thus, when one wants to update a part of a tree, one has to manually find and update all duplicates in other trees (other defects). Some mind mapping tools provide script mechanisms (e.g. Freeplane), but they do not fit well with our needs.

In the next section, we will define the notion of ontology, which is well suited to overcoming the weaknesses of mind mapping tools mentioned above. We will then explain why we have chosen the conceptual graph (CG) model as a specific ontology model, and we will review briefly its main principles. Afterwards, we will present the new core ontology, dedicated to the representation of explanation relations between situations, used in this DSS. This core ontology has been published on the INRA Dataverse repository to be shared with the food processing community (https://doi.org/10.15454/9Z4PS3). We will also describe the algorithm we have developed to automatically translate mind map explanation trees into the CG formalism using this core ontology. Finally, we will present the domain ontology, which represents specific concepts associated with a given process in a non-ambiguous way.

4.1 Ontology

An ontology is a form of formal knowledge representation which is well suited to our purposes [Staab and Studer, 2009]. It is a hierarchically structured set of concepts thanks to the *kind of* relation as well as the relationships between these concepts. Ontologies allow similar pieces of knowledge to be structured in the same way by defining core ontologies dedicated to a specific task. They provide powerful querying and reasoning mechanisms for exploiting the knowledge and managing changes in this knowledge. Moreover, domain ontologies allow the concepts of a given application domain to be defined in a non-ambiguous way. Finally, core and domain ontologies may be shared by communities thanks to public repositories. In the agri-food and food processing domains in particular, many ontologies have been proposed in recent years [Buche et al., 2013b],[Muljarto et al., 2014],[Lousteau-Cazalet et al., 2016], [Ibanescu et al., 2016],[Poveda-Villalón et al., 2018] and published on AgroPortal [Jonquet et al., 2018].

4.2 The conceptual graph formalism

In order to encode ontologies, we chose to use conceptual graphs (CGs) [Sowa, 1984] with CoGui, a software tool which allows CGs to be managed. We chose CGs for several reasons: (i) their terminological support described below allows ontologies to be defined; (ii) CG models provide querying and reasoning mechanisms to retrieve knowledge; and finally, (iii) while CGs have a logical translation in first-order logic as described in [Chein and Mugnier, 2009], they are usually expressed graphically, which is very important so that end users can easily interact with them.

CGs are composed of two parts: the terminological support and a set of facts (data). Below, we review the main principles of CG formalism (readers should refer to [Chein and Mugnier, 2009] for a complete overview).

The terminological support is composed of a set of concept types and a set of relation types. Each relation has a given signature that defines its arity and the type of concepts with which it can be associated. Concept type and relation type sets are partially ordered by a *kind of* relation. Two relation types can be compared only if they have the same arity. An example of terminological support is given in Figure 4. It contains the concept type "Situation" and the relation "can be explained by" which links a "Situation" to another "Situation".

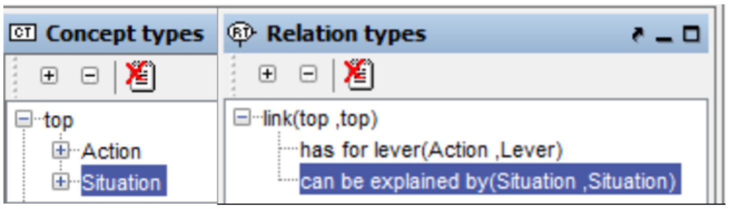

Concept types
Relation types
top
Action
Situation
link(top ,top)
has for lever(Action ,Lever)
can be explained by(Situation ,Situation)

Figure 4: Terminological support: an example of the hierarchy of concept types and relation types

A fact (or fact graph) allows data to be encoded based on the vocabulary defined in the ontology. This is a bipartite graph composed of:

- concept nodes represented by rectangles which define entities. Each concept is labelled by a pair $t : m$, where t is a concept type of the ontology and m is a marker. Either m is used as the name of an individual marker, or the symbol $*$, which denotes an unspecified individual marker called the generic marker. A $t : *$ concept node means that there is an individual belonging to concept type t which is defined in the knowledge base.

- relation nodes, represented by ovals linked to concept nodes, express some relationships between concept nodes. Each relation has to satisfy its signature: the number of incident edges is equal to the relation's arity, and the concept type of a node linked by the i^{th} edge is more specific or equal (depending on the terminological support's *kind of* relation) to the i^{th} element of the relation's signature.

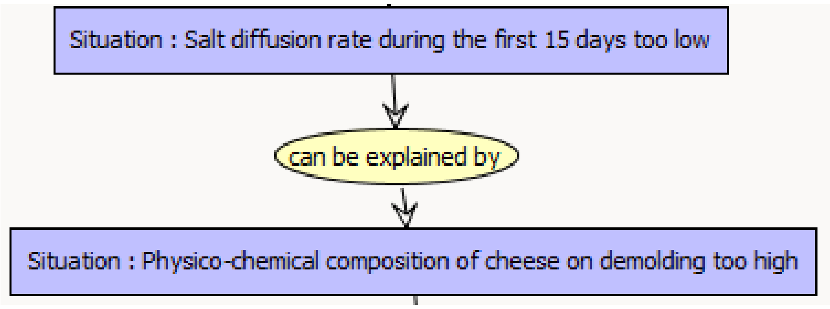

Situation : Salt diffusion rate during the first 15 days too low
can be explained by
Situation : Physico-chemical composition of cheese on demolding too high

Figure 5: Example of basic conceptual graph

In Figure 5, the example CG graph shows that the situation "Salt diffusion rate during first 15 days too low" can be explained by the situation "Physico-chemical composition of cheese on demolding too high". In terms of conceptual graphs, the first situation is represented by the concept node type “Situation” and has as its marker “Salt diffusion rate during first 15 days too low”. It is linked to the concept node type “Situation” and the marker “Physico-chemical composition of cheese on demolding too high” by the relation node “can be explained by”.

Finally, CG models created in CoGui offer a query mechanism that allows us to retrieve situations using their relationships with other concepts of the ontology.

4.3 Core ontology of the DSS

In this section, we will present the core ontology of the DSS, which is dedicated to the representation of explanatory relations between situations and recommendations. It is not specific to any one application domain. It has been designed to fulfil the needs defined in Section 2, and more specifically Sections 2.2 and 2.3.

Concepts

- a **Situation** describes a partial state in the agri-food chain process;
  - a **Situation of interest** is a particular situation for which an explanation tree has been created;
    - a **defect** is associated with a defect of the product in the agri-food chain process which must be corrected;

        - * **quality** is associated with a level of quality of the product which must be maintained;
    - a **Joint situation** is associated with an explanatory situation which is the combination of two or more other situations;

- **Action** is associated with actions to be taken to correct a defect or maintain a quality standard;
    - a **Corrective action** is an action that corrects a particular defect;
    - **Compensatory action** is an action that counteracts a particular situation: the problem will still exist, but its impact will be reduced;
    - a **Recommendation** is an action that should be carried out to maintain a particular quality standard of the product;
- a **Lever** (or controlled parameter) is an element that can be operated to control the agri-food chain process;
- a **Unit operation** represents a step in the process;
- a **State variable** is an analytical variable and associated value which defines when a situation occurs.

Relations

- **is composed of** allows several situations to be combined into a 'Joint situation';

- **can be explained by** links a situation to a potential cause of it;

- **cannot be explained by** makes it clear that a given situation cannot be the cause of another given situation. It is very helpful to indicate that a belief is false;

- **has for action/can be resolved by** links a situation to possible actions to solve it;

- **has for lever** allows actions to be grouped according to levers which are operated;

- **occurs in** allows a particular 'situation' to be attached to its 'unit operation';

- **occurs before** allows a particular 'situation' to be attached to a subsequent 'unit operation';

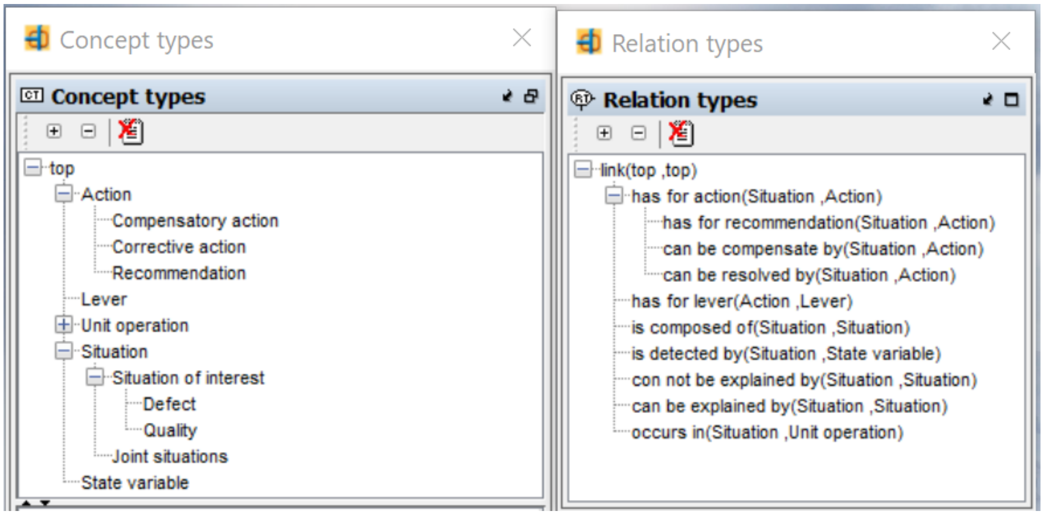
- **occurs after** allows a particular 'situation' to be attached to a preceding 'unit operation';

- **is detected by** links a 'state variable' to a 'situation' it highlights.

Using the CoGui graphical user interface (GUI), Figure 6 shows the core ontology's hierarchy of concept types on the left and its hierarchy of relations on the right.

The core ontology allows the explanation tree associated with a given situation of interest to be represented. Figure 7 shows a section of an explanation tree expressed using the core ontology. Both the core ontology and section of an explanation tree shown in Figure 7 can be downloaded from the INRA Dataverse repository (https://doi.org/10.15454/9Z4PS3).

Table 1 shows some statistics about knowledge bases for three different domains produced using the core ontology.



Concept types

Concept types

- top
  - Action
    - Compensatory action
    - Corrective action
    - Recommendation
  - Lever
  - Unit operation
  - Situation
    - Situation of interest
      - Defect
      - Quality
    - Joint situations
  - State variable

Relation types

Relation types

- link(top ,top)
  - has for action(Situation ,Action)
    - has for recommendation(Situation ,Action)
    - can be compensate by(Situation ,Action)
    - can be resolved by(Situation ,Action)
  - has for lever(Action ,Lever)
  - is composed of(Situation ,Situation)
  - is detected by(Situation ,State variable)
  - con not be explained by(Situation ,Situation)
  - can be explained by(Situation ,Situation)
  - occurs in(Situation ,Unit operation)

Figure 6: Hierarchy of the core ontology's concept types and relation types

	# concept instances	**# relation instances**	**# situations of interest**
couscous factory	305	418	3
milk powder factory	134	135	2
cheese-making technical centre	885	1041	17

Table 1: Statistics about knowledge bases produced for three different domains

Finally, it may be noted that the CG model ensures that two nodes with the same marker are equivalent. So, if two explanation trees associated with

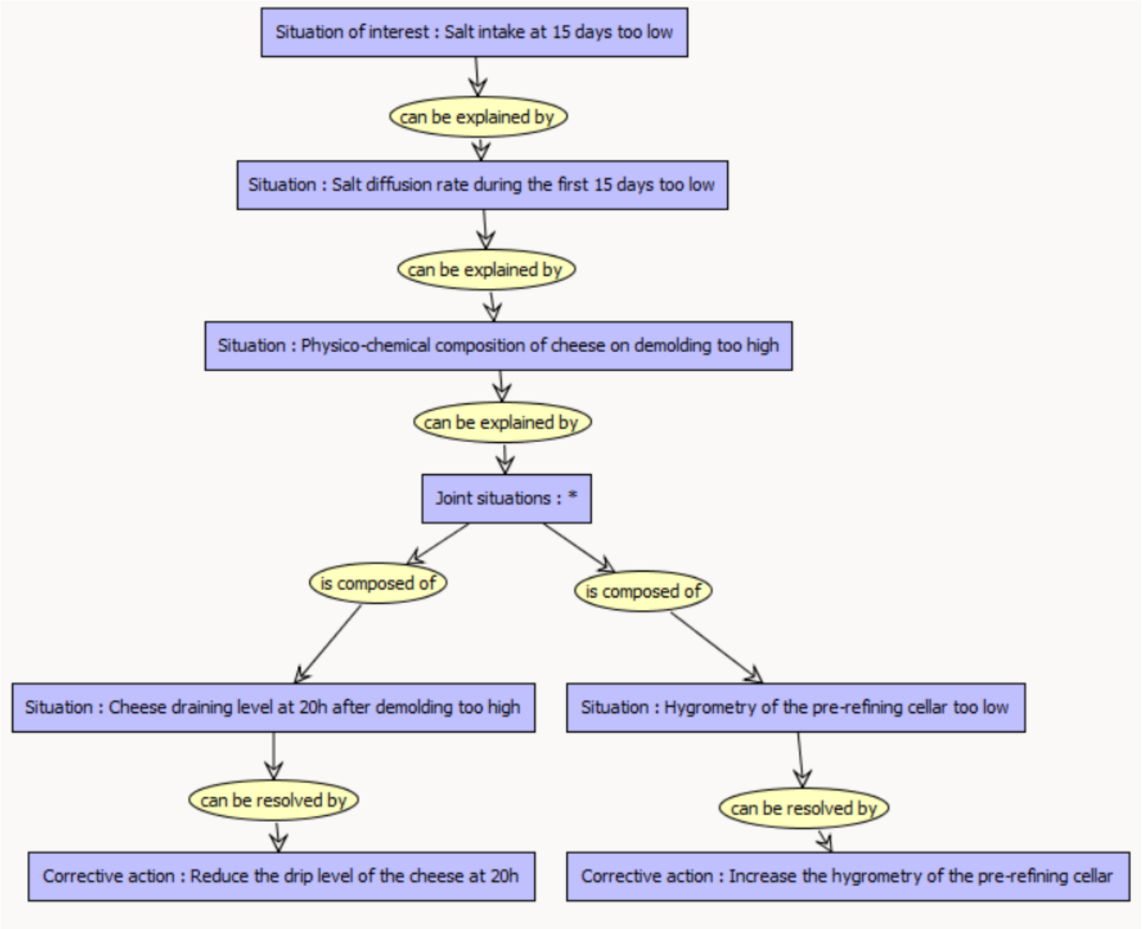

Situation of interest : Salt intake at 15 days too low
can be explained by
Situation : Salt diffusion rate during the first 15 days too low
can be explained by
Situation : Physico-chemical composition of cheese on demolding too high
can be explained by
Joint situations : *
is composed of
is composed of
Situation : Cheese draining level at 20h after demolding too high
Situation : Hygrometry of the pre-refining cellar too low
can be resolved by
can be resolved by
Corrective action : Reduce the drip level of the cheese at 20h
Corrective action : Increase the hygrometry of the pre-refining cellar

Figure 7: Section of an explanation tree expressed using the core ontology

two situations of interest share the same sub-tree of explanations, only one sub-tree has to be represented in the CG knowledge base. In this way, we avoid duplicates, which was one of the requirements set out in the introduction to Section 4.

4.4 Importation from mind mapping tool

In order to reduce the business user's workload, we developed an automatic import module which takes as its input the explanatory tree edited using mind mapping tool, and generates as its output a knowledge base using the core ontology described in the previous section. This importation module makes use of mind mapping tool's plain text export feature and the CoGui Core library. By reading the mind mapping tool text export, each mind map node can be processed by creating a concept vertex whose marker is the text associated with the mind map node and whose concept type is a core ontology concept which depends on the mind map node's position in the tree or its specific prefix in the mind map node text. At the same time, the import module links these nodes together with corresponding relations defined by the core ontology. We finally obtain a conceptual graph representing the explanatory tree which can incorporate additional knowledge, as we will see in Section 4.5, and which is queryable using native conceptual graph querying operators. Below, we will present the main rules used by the import module to generate an CG explanation tree from the mind mapping tool export.

Importation rules based on the type of mind map node:

- root nodes are translated into a Situation of interest;
- by default a child node is a Situation connected to its parent node by a can be explained by relation;

- prefixed nodes have special treatment:
  - nodes prefixed by LEVER: are translated into a lever connected to their subsequent node by a has for lever relation,
  - nodes prefixed by ACOR: are translated into a corrective action connected to the previous situation by a can be resolved by relation;
- finally, AND nodes are translated into a Joint situation concept connected to their sons by a is composed of relation.

Example From the mind mapped explanation tree shown in Figure 8, and using the mind mapping tool's built-in export, we are able to obtain the plain text representation that can be seen in Figure 9. The import module then uses this export to construct the conceptual graph shown in Figure 7 which represents an excerpt of the explanation tree shown in Figure 8.

Importing also allows us to check the coherence between the different occurrences of the same situation in the mapped explanation trees. If a situation occurs more than once, we record each occurrence with its son nodes. In the final CG, all occurrences are merged into a single situation referenced by several trees.

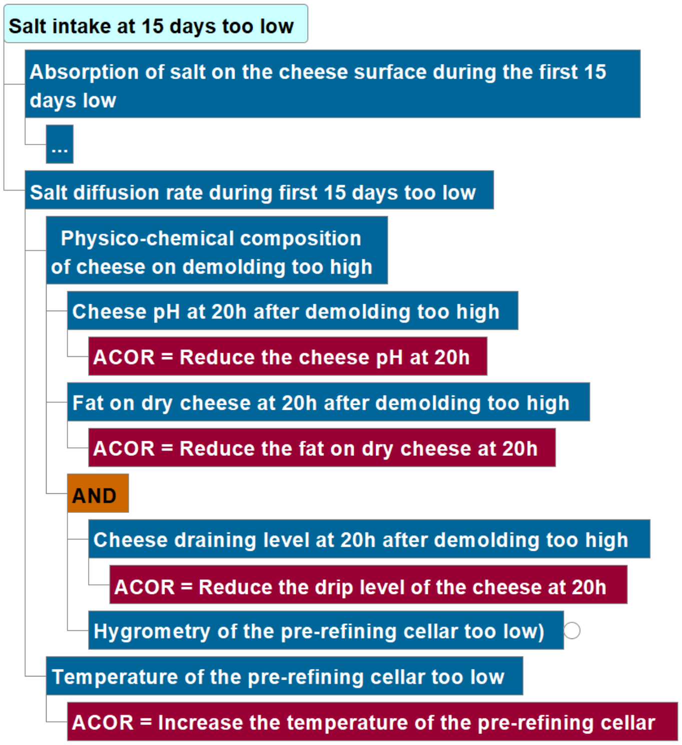
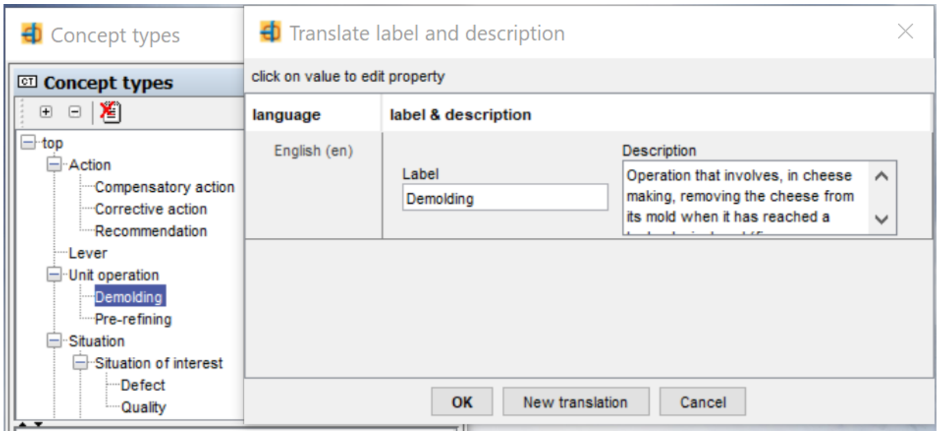

Salt intake at 15 days too low
Absorption of salt on the cheese surface during the first 15 days low
...
Salt diffusion rate during first 15 days too low
Physico-chemical composition of cheese on demolding too high
Cheese pH at 20h after demolding too high
ACOR = Reduce the cheese pH at 20h
Fat on dry cheese at 20h after demolding too high
ACOR = Reduce the fat on dry cheese at 20h
AND
Cheese draining level at 20h after demolding too high
ACOR = Reduce the drip level of the cheese at 20h
Hygrometry of the pre-refining cellar too low)
Temperature of the pre-refining cellar too low
ACOR = Increase the temperature of the pre-refining cellar

Figure 8: Example of a mind mapped explanation tree

Figure 9: Mind mapping plain text export

4.5 Domain ontology

As mentioned in Section 2.1, vocabulary must be defined in an unambiguous way before being used to represent domain knowledge. Figure 10 shows how this definition can be easily associated with concepts from the domain ontology. Moreover, the knowledge base must be easily modifiable, as described in Section 2.4, to allow knowledge evolution to be taken into account. More precisely, in order to identify the actual piece of knowledge which must be updated in case of knowledge evolution, one needs to be able to contextualize the querying of the knowledge base using, for example, analytical values (e.g. temperature level, pH value, etc) or temporal information (e.g. unit operation). Contextual querying requires the knowledge base to be added to with specific concepts which depend on the application domain. These concepts are defined in the domain ontology for each application and are specializations of the core ontology concepts.



English (en)

Label: Demolding

Description: Operation that involves, in cheese making, removing the cheese from its mold when it has reached a

Figure 10: Definition associated with a concept belonging to the domain ontology

For example, Figure 11 shows an enriched section of the explanation tree provided in Figure 7 to which has been added the unit operations occurring during a given situation. In this explanation tree, the concept types **Pre-refining** and **Demolding** belong to the cheese-making domain ontology. There are also subtypes of the **Unit operation** concept which belongs to the core ontology.

To update the knowledge available on the **Pre-refining** unit operation,

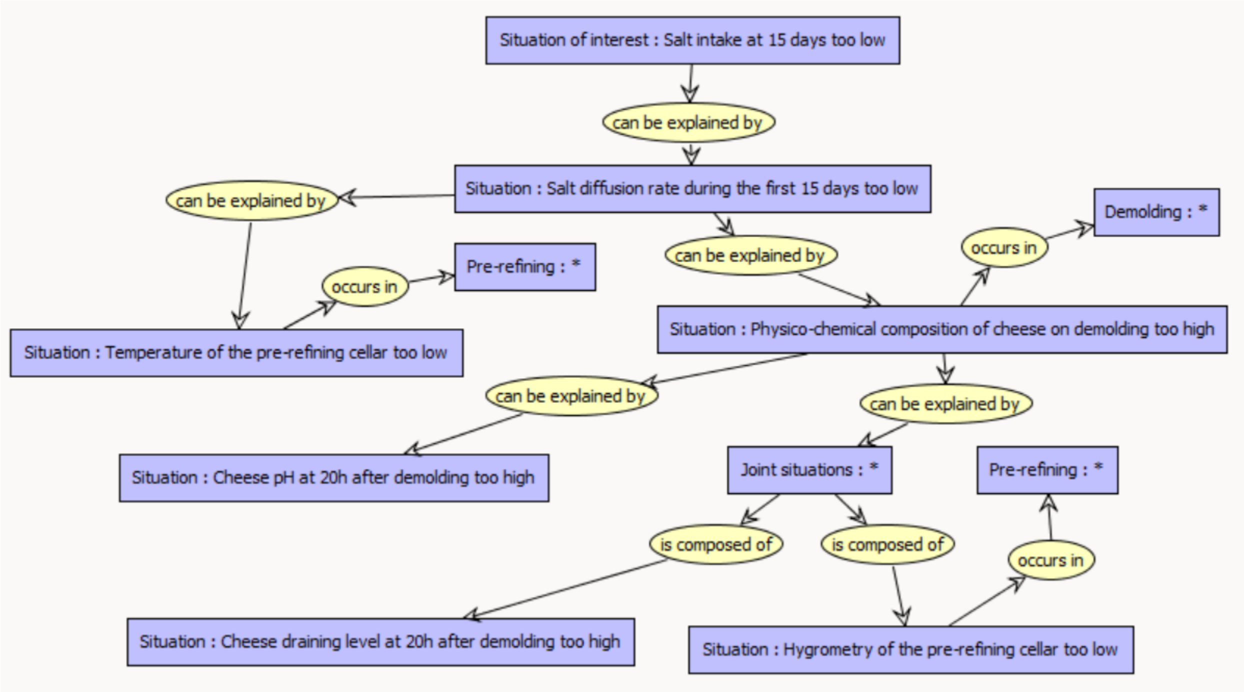

Situation of interest : Salt intake at 15 days too low
can be explained by
Situation : Salt diffusion rate during the first 15 days too low
can be explained by
Demolding : *
can be explained by
occurs in
Pre-refining : *
occurs in
Situation : Physico-chemical composition of cheese on demolding too high
Situation : Temperature of the pre-refining cellar too low
can be explained by
can be explained by
Situation : Cheese pH at 20h after demolding too high
Joint situations : *
Pre-refining : *
is composed of
is composed of
occurs in
Situation : Cheese draining level at 20h after demolding too high
Situation : Hygrometry of the pre-refining cellar too low

Figure 11: Section of an explanation tree expressed using the core ontology and the cheese-making domain ontology

the list of situations already available in the knowledge base (and more precisely, in the explanation tree presented in Figure 11) can be obtained using the query shown in Figure 12. This will retrieve both answers shown in Figure 12 using the CG model's native querying operator.

5 Decision support system

The DSS application allows business users to quickly access the relevant information stored in the knowledge base and compare data from all the explanation trees. The application has been developed on top of the CoGui

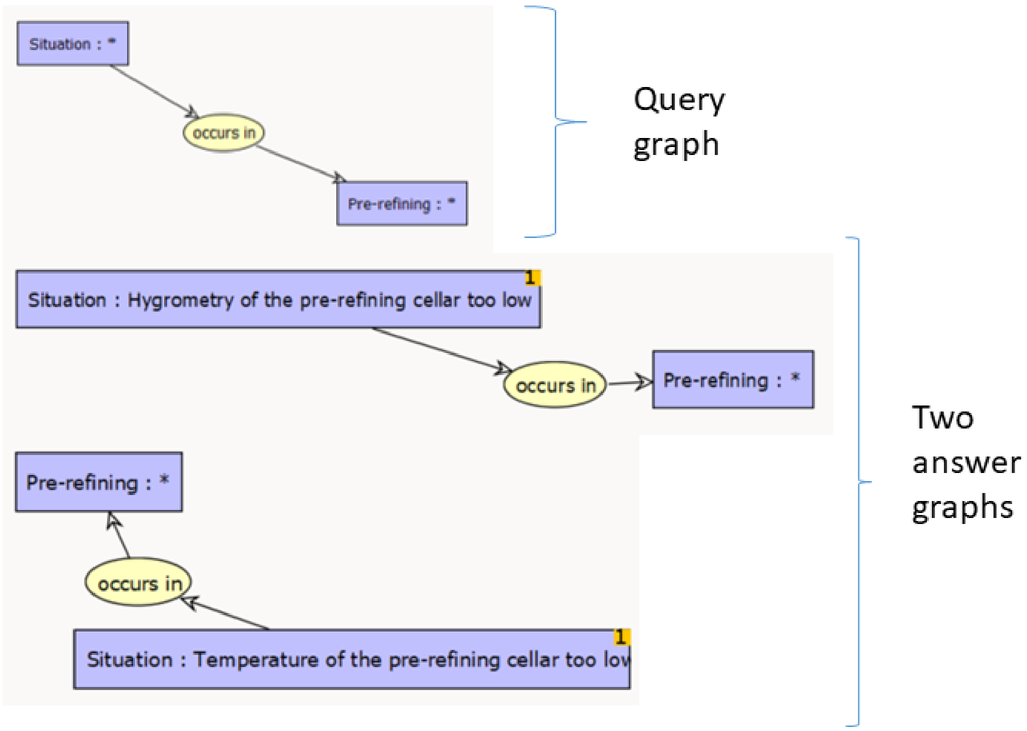
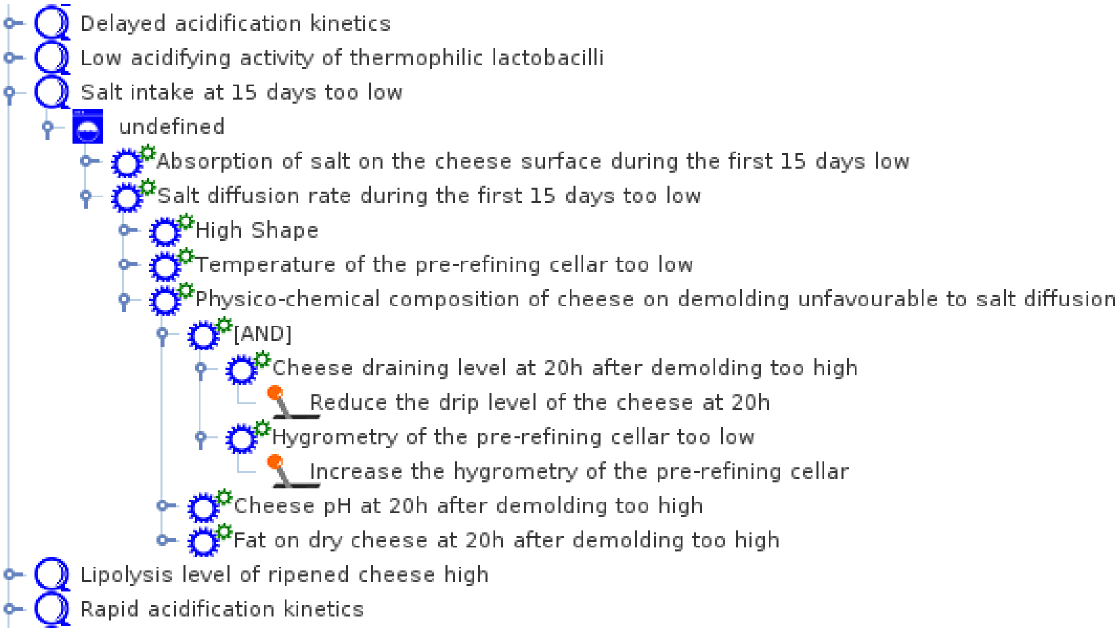


Figure 12: Example of a query graph allowing the situations occurring during the **Pre-refining** unit operation to be retrieved, and two answers retrieved from the knowledge base using this query

Core library using the NetBeans platform framework.

It provides three main features:

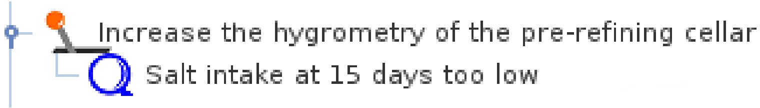
- **the first** implements the functional specification described in Section 2.2. The DSS displays and allows you to explore each explanation tree from a situation of interest through explanatory situations to a possible action that could be taken. Figure 13 illustrates this feature on the cheese-making application. Five defects are displayed, and two possible explanations are given for the “Salt intake at 15 days too low” defect. One of these is “Physico-chemical composition of cheese on demolding unfavorable to salt diffusion”, which can be explained by two joint situations. One of these situations, “Hygrometry of the pre-refining cellar too low”, may be solved by increasing the hygrometry of the pre-refining cellar.



```
Delayed acidification kinetics
Low acidifying activity of thermophilic lactobacilli
Salt intake at 15 days too low
    undefined
        Absorption of salt on the cheese surface during the first 15 days low
        Salt diffusion rate during the first 15 days too low
            High Shape
            Temperature of the pre-refining cellar too low
            Physico-chemical composition of cheese on demolding unfavourable to salt diffusion
                [AND]
                    Cheese draining level at 20h after demolding too high
                        Reduce the drip level of the cheese at 20h
                    Hygrometry of the pre-refining cellar too low
                        Increase the hygrometry of the pre-refining cellar
                Cheese pH at 20h after demolding too high
                Fat on dry cheese at 20h after demolding too high
Lipolysis level of ripened cheese high
Rapid acidification kinetics
```

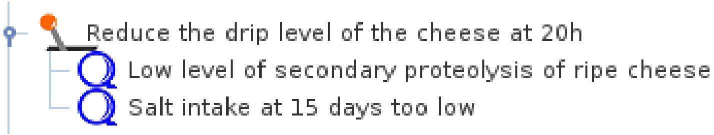
Figure 13: First main feature in the cheese process application

- **the second** implements the functional specification described in Section 2.3. The DSS displays the list of all situations of interest potentially impacted by an action. In Figure 14, the lever “Increase the hygrometry of the pre-refining cellar” is only associated with the defect “Salt intake at 15 days too low”, whereas in Figure 15 the lever “Reduce the drip level of the cheese at 20h” is associated with the situations of interest “Salt intake at 15 days too low” and “Low level of secondary proteolysis of ripe cheese”.



Increase the hygrometry of the pre-refining cellar
Salt intake at 15 days too low

Figure 14: Second feature: only one situation of interest impacted by this lever in the cheese-making process application



Reduce the drip level of the cheese at 20h
Low level of secondary proteolysis of ripe cheese
Salt intake at 15 days too low

Figure 15: Second feature: two situations of interest impacted by this lever in the cheese process application

The third functional specification described in Section 2.4 is realized by the knowledge manager directly using the CoGui software application thanks to predefined CG queries such as the one shown in Figure 12 which allow them to identify which explanation trees have to be updated.

6 Validation process

There are two parts to the process of validating the results delivered by the DSS: validation of the knowledge base content and validation of the DSS functional specifications defined in Section 2.

The knowledge base content has been validated for the cheese application. The main characteristics of this validation process are presented below. Each explanation tree associated with a situation of interest has been validated. Expert technicians were chosen to carry out this validation. The validation team was composed of the group of 10 technical advisers belonging to the cheese technical centre. As each participant is the adviser for 10 to 15 dairies in his/her everyday activities, the collective expertise of the validation team is based on the acquired experience in cheese processing of 100 to 150 dairies, which has been judged representative enough for this cheese-making process food chain. One collective session was organized for each explanation tree. Each tree was validated by the group of participants over the course of half a day. A branch of the tree is considered validated when all the participants validate it. Due to lack of time in the other projects, this validation has not been carried out for the couscous and milk powder applications.

The features of the DSS defined in Section 2 were validated using the classic use-case testing procedure. This validation was carried out by involving participants of three different food processing processes (cheese, couscous, milk powder) as part of different projects. Some collective testing sessions were organized in order to have the DSS's features validated by its potential users. Following these testing sessions, the users requested that graphical user interfaces (GUI) be introduced for the functional specifications described in Sections 2.2 and 2.3, inspired by the file explorer GUI that they are used to using in their everyday work.

7 Comparison with current research

Despite increased numbers of scientific publications in the field of food science & technology, capitalization on technological knowledge remains fragmented and incomplete [Perrot et al., 2011, Aceves Lara et al., 2017]. Several approaches have been proposed to pool technical knowledge and the available data, but they do not generally exceed the scale of a unit operation. For example, [Ndiaye et al., 2009] propose a method of qualitative modelling of the kneading unit operation, making it possible to predict descriptors of the states of the dough in the field of breadmaking. [Baudrit et al., 2010] model the dynamics of microbial growth from dynamic Bayesian networks by combining several scales of organization during the cheese-ripening process. Although [Thomopoulos et al., 2009] propose a model for collecting available data and knowledge for the durum wheat sector, use of this information is restricted to the "pasta cooking" unit operation [Thomopoulos et al., 2013]. [Guillard et al., 2015] and [Tamani et al., 2015] propose a decision support system which retrieves the best packaging for a given fresh food, taking into account several criteria (food packaging permeability optimization, consumer's preferences in terms of transparency, etc.), which is restricted to the stage in the process of processing fresh food when the food is preserved in packaging. [Muljarto et al., 2017] propose an ontological model which allows experimental data on the entire set of unit operations, from field to the final wine product, to be pooled. While this ontological model facilitates the analysis of the impact of vine operations (for example irrigation) on the final quality of the wine, the approach requires a huge set of numerical data to obtain robust statistical results. These approaches do not use the available expertise on the cause-and-effect relationships between all the possible descriptors (product qualities or defects) of interest and intervention levers in the processing process, allowing these relationships to be represented and reasoned, which is the purpose and originality of our paper.

Our approach should be compared to the fault tree analysis (FTA), which has been developed as a failure analysis tool [Baig et al., 2013, Lee et al., 1985]. It allows the level of risk in terms of the probability of an undesired event occurring to be calculated, and can help to identify safety critical components. A fault tree is a tree with a root labelled as an undesired event. The leaves of the tree are basic events representing minor failures that likely contribute to the global failure expressed in the root. Intermediate nodes of the trees allow some kinds of logical gates to be modelled in order to express how several failures have an impact on a global failure. Several aggregation nodes are possible, for example, at least one or at least k from n. A whole fault tree can be seen as a Boolean function that provides the global failure state of the system based on the failure of its subsystems. Our core ontology is rather similar to the tree structure used in FTA. Our contribution consists mainly in proposing a semantic representation of the tree structure using the conceptual graph model. This allows the knowledge base to be queried thanks to the core ontology and the domain ontology, which can be specifically defined for each application. Moreover, it should also be noted that FTA focuses mainly on probabilistic risk assessment of failures, which requires a huge amount of numerical data to be implemented, whereas our approach is mainly focused on gathering the collective qualitative technical expertise available for a given domain (company, factory, etc.) in order to reuse it to propose recommendations.

8 Conclusion and perspectives

We are proposing a complete methodology and associated software pipeline which allows collective knowledge on technical expertise to be collected. The method is able to take into account diverse sources of information (interviews of experts and technicians, scientific papers, technical reports, etc.). This expertise is recorded in a knowledge base using a core ontology and a domain ontology. The knowledge base is a collection of explanatory trees which link situations of interest (product quality or defects) to actions by way of explanatory situations. A GUI has been designed and implemented that takes into account feedback from end users. It has been tested successfully on three different applications (production of cheese, couscous and milk powder), showing that our method and tool are generic and could be applied to a large variety of production sectors.

To reuse this methodology in new studies, the steps presented in Figure 1 must be followed. The main actions to be taken are:

- Definition of the scope (processing process for which the expertise needs to be collected)
- Design of the systematic questionnaire on the processing process
- Interviews using the questionnaire to collect knowledge
- Synthesis of collected knowledge to build the mind maps
- Automatic generation of the CG knowledge base
- Loading of the knowledge base in the DSS to obtain recommendations

The very next step would be to take into account the uncertainty associated with <situation of interest, explanatory situation, action> triplets. This would be based on the frequency of the situations and the effectiveness of this action in this situation.

Acknowledgments

The research leading to these results has received partial funding from the CASDAR Docamex Programme from the French Ministry of Agriculture (2016-2020). It has also benefited from preliminary works performed during one project with the Technical Centre of Cheeses in Poligny (CTFC) and two projects with French industrial partners. The authors would also like to thank the CTFC, who authorized us to present in this paper some of

the knowledge provided by their technical experts and extracted from their knowledge base.

References

[Aceves Lara et al., 2017] Aceves Lara, C., Athes, V., Buche, P., Della Valle, G., Farines, V., Fonseca, F., Guillard, V., Kansou, K., Kristiawan, M., Monclus, V., Mouret, J., Ndiaye, A., Neveu, P., Passot, S., Penicaud, C., Sablayrolles, J., Salmon, J., Thomopoulos, R., and Trelea, I. (2017). The virtual food system: Innovative models and experiential feedback in technologies for winemaking, the cereals chain, food packaging and ecodesigned starter production. Innovative Food Science and Emerging Technologies, 46:54–64.

[Baig et al., 2013] Baig, A., Ruzli, R., and Buang, A. (2013). Reliability Analysis Using Fault Tree Analysis: A Review. International Journal of Chemical Engineering and Applications, 4(3):169—-173.

[Baudrit et al., 2010] Baudrit, C., Sicard, M., G., Willemin, P. H., and Perrot, N. (2010). Towards a global modelling of the Camembert-type cheese ripening process by coupling heterogeneous knowledge with dynamic Bayesian networks. Journal of Food Engineering, 98:283–293.

[Buche et al., 2013a] Buche, P., Cucheval, V., Diattara, A., Fortin, J., and Gutierrez, A. (2013a). Implementation of a knowledge representation and reasoning tool using default rules for a decision support system in agronomy applications. In Graph Structures for Knowledge Representation and Reasoning - Third International Workshop, GKR 2013, Beijing, China, August 3, 2013. Revised Selected Papers, pages 1–12.

[Buche et al., 2013b] Buche, P., Dibie-Barthélemy, J., Ibanescu, L., and Soler, L. (2013b). Fuzzy web data tables integration guided by an ontological and terminological resource. IEEE Trans. Knowl. Data Eng., 25(4):805–819.

[Buzan, 2004] Buzan, T. (2004). The Mind Map Handbook.

[Cerf et al., 2011] Cerf, M., Guillot, M., and Olry, P. (2011). Acting as a Change Agent in Supporting Sustainable Agriculture: How to Cope with New Professional Situations? The Journal of Agricultural Education and Extension, 17:1:7–19.

[Chein and Mugnier, 2009] Chein, M. and Mugnier, M.-L. (2009). Graph-based Knowledge Representation and Reasoning. Computational Foundations of Conceptual Graphs. Springer, Advanced Information and Knowledge Processing Series, London.

[Depraz et al., 2003] Depraz, N., J., V. F., and Vermersch, P. (2003). On Becoming Aware, A pragmatics of experiencing. John Benjamins Publishing Company.

[Guillard et al., 2015] Guillard, V., Buche, P., Destercke, S., Tamani, N., Croitoru, M., Menut, L., Guillaume, C., and Gontard, N. (2015). A decision support system to design modified atmosphere packaging for fresh produce based on a bipolar flexible querying approach. Computers and Electronics in Agriculture, 111:131–139.

[Ibanescu et al., 2016] Ibanescu, L., Dibie, J., Dervaux, S., Guichard, E., and Raad, J. (2016). Po^2 - A process and observation ontology in food science. application to dairy gels. In Metadata and Semantics Research - 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings, pages 155–165.

[Jonquet et al., 2018] Jonquet, C., Toulet, A., Arnaud, E., Aubin, S., Kaboré, E. D. Y., Emonet, V., Graybeal, J., Laporte, M., Musen, M. A., Pesce, V., and Larmande, P. (2018). Agroportal: A vocabulary and ontology repository for agronomy. Computers and Electronics in Agriculture, 144:126–143.

[Lee et al., 1985] Lee, W., Grosh, D., Tillman, F., and Lie, C. (1985). Fault Tree Analysis, Methods, and Applications A Review. IEEE Transactions on Reliability, 34(3):194—-203.

[Lousteau-Cazalet et al., 2016] Lousteau-Cazalet, C., Barakat, A., Belaud, J. P., Buche, P., Busset, G., Charnomordic, B., Dervaux, S., Destercke, S., Dibie, J., Sablayrolles, C., and Vialle, C. (2016). A decision support system for eco-efficient biorefinery process comparison using a semantic approach. Computers and Electronics in Agriculture, 127:351–367.

[Muljarto et al., 2017] Muljarto, A., Salmon, J., Charnomordic, B., Buche, P., Tireau, A., and Neveu, P. (2017). A generic ontological network for agri-food experiment integration - application to viticulture and winemaking. Computers and Electronics in Agriculture, 140:433–442.

[Muljarto et al., 2014] Muljarto, A., Salmon, J., Neveu, P., Charnomordic, B., and Buche, P. (2014). Ontology-based model for food transformation processes - application to winemaking. In Metadata and Semantics Research - 8th Research Conference, MTSR 2014, Karlsruhe, Germany, November 27-29, 2014. Proceedings, pages 329–343.

[Ndiaye et al., 2009] Ndiaye, A., Della Valle, G., and Roussel, P. (2009). Qualitative modelling of a multi-step process: the case of French breadmaking. Expert Systems with Applications, 36:1020–1038.

[Perrot et al., 2011] Perrot, N., Trelea, I. C., Baudrit, C., Trystram, G., and P, B. (2011). Modelling and analysis of complex food systems: state of the art and new trends. Trends in Food Science and Technology, 22:304–314.

[Piot, 2012] Piot, T. (2012). The role of professional didactics in skills development for training and education professionals. In Cohen-Scali, V., editor, Competence and competence development., pages 53–72. Verlag Barbara Budrich.

[Poveda-Villalón et al., 2018] Poveda-Villalón, M., Nguyen, Q., Roussey, C., Vaulx, C. D., and Chanet, J. (2018). Ontological requirement specification for smart irrigation systems: A SOSA/SSN and SAREF comparison. In Proceedings of the 9th International Semantic Sensor Networks Workshop co-located with 17th International Semantic Web Conference, SSN@ISWC 2018, Monterey, CA, United States, October 9, 2018., pages 1–16.

[Sowa, 1984] Sowa, J. F. (1984). Conceptual Structures: Information Proc. in Mind and Machine. Addison–Wesley.

[Staab and Studer, 2009] Staab, S. and Studer, R., editors (2009). Handbook on Ontologies, International Handbooks on Information Systems. Springer.

[Tamani et al., 2015] Tamani, N., Mosse, P., Croitoru, M., Buche, P., Guillard, V., Guillaume, C., and Gontard, N. (2015). An argumentation system for eco-efficient packaging material selection. Computers and Electronics in Agriculture, 113:174–192.

[Thomopoulos et al., 2009] Thomopoulos, R., Charnomordic, B., Cuq, B., and Abecassis, J. (2009). Artificial intelligence-based decision support system to manage quality of durum wheat products. Quality Assurance and Safety of Crops and Foods, 1757–8361:179–190.

[Thomopoulos et al., 2013] Thomopoulos, R., Destercke, S., Charnomordic, B., Johnson, L., and Abecassis, J. (2013). An iterative approach to build relevant ontology-aware data-driven models. Inf. Sci., 221:452–472.