Combining fuzzy querying of imprecise data and predictive microbiology
using category-based reasoning
for prediction of the possible microbial spoilage in foods:
application to Listeria monocytogenes.

PATRICE BUCHE*1, CATHERINE DERVIN1,
AGNES BROUILLAUD-DELATTRE3, NATHALIE GNANOU-BESSE2,

1UMR INA P-G/INRA, BIA Paris
16, rue Claude Bernard 75231 Paris Cédex 5 FRANCE
Tel 33 1 44 08 16 75, Fax 33 1 44 08 16 66
E-mail: Patrice.Buche@inapg.inra.fr

2Agence Française de Sécurité Sanitaire des Aliments, Laboratoire d'Etudes et de Recherches
sur l'Hygiène et la Qualité des Aliments, 39/41, rue du 11 novembre 94700 Maisons-Alfort

3Agence Française de Sécurité Sanitaire des Aliments/Ecole Nationale Vétérinaire de
Toulouse , 23 chemin des Capelles 31076 Toulouse cedex 3

* corresponding author

Abstract

Various predictive models of microbial behavior have been created and extensive data
collection has been done by numerous private or public laboratories. However, significant
differences between predicted and observed values in foods have been observed and need to
be stressed, understood and explained as much as possible. In this paper, we present a
software tool (currently at the level of a prototype) able: (i) to store in a database all relevant
information expressed on one hand as qualitative or quantitative data and on the other hand as
precise or imprecise data, (ii) to retrieve the more relevant information from the database
using queries where criteria may be expressed as fuzzy values in order to enhance the
flexibility of the search, (iii) to compute, in addition to the nearest data, an estimation of
searched values using statistical models. The architecture of this software tool is structured as
a category-based reasoning system. Example queries about *Listeria monocytogenes* (*L. monocytogenes*) illustrate functionalities of this tool.

**Key words**: Predictive microbiology, *Listeria monocytogenes*, Fuzzy logic, Information retrieval system

**Introduction**

Numerous factors affect microbial growth in food systems and often synergistic or antagonist interactions are significant. Thus, microbial growth in foods can be enhanced or inhibited depending on multiple food ingredients and storage conditions (Eifert et al., 1996). This can explain why an understanding of microbial behavior usually requires large-scale experiments under a wide range of conditions and involving extensive data collection.

Since data collection can be very time-consuming and labor intensive, numerous laboratories have developed predictive growth models, especially in Australia, the U.K., the U.S.A. and France with particular reference to *L. monocytogenes*.

*Listeria monocytogenes* has been associated in the last 15 years with several food-borne illness outbreaks involving a large range of foods such as raw milk, cheeses (particularly soft-ripened varieties), raw vegetables, fermented raw-meat sausages, raw or cooked meats. Microbial safety of many food products depends on refrigeration after commercialization, as most microorganisms do not grow at refrigeration temperatures. However this is not the case for *L. monocytogenes* because of its ability to grow at temperatures as low as 3°C (even 0.1°C and –0.4°C (Bajard et al., 1996).

In order to evaluate food safety relating to the *Listeria* growth hazard in "chilled prepared food products", temperature, pH and water activity are the main factors that must be taken into account in predictive models (Bajard et al., 1996). Various models are able to predict lag times and growth rates of *L. monocytogenes* (or other microorganisms) in many food matrices.
at constant temperatures (Ratkowsky et al., 1983; Zwietering et al., 1991; Duh and Schaffner, 1993; Rosso et al., 1993; Baranyi et al., 1993) and a small number of models also do so in a dynamic temperature environment (Baranyi et al., 1995; Van Impe et al., 1995; Alavi et al., 1999). Most of them have improved their predictions by including other environmental factors such as the presence of organic acids, the nature of packaging atmosphere (Buchanan and Phillips, 1990; Murphy et al., 1996; Fernandez et al., 1997) or some other biological factors such as the physiological state of inoculum cells and inoculum size (Mc Kellar et al., 1997).

However, some authors have observed significant differences between predicted values and observed values in foods and suggest that these differences are due to factors not taken into account in existing models (Brouillaud-Delattre et al., 1997; Dalgaard and Jorgensen, 1998). For example, the growth response of *L. monocytogenes* can be modified by the strain considered (Siswanto and Richard, 1992; Begot et al., 1997), the accumulation of inhibiting substances (Guinot-Thomas et al., 1995), initial contamination level (Gay et al. 1996), competitive interactions (Farrag and Marth, 1989) or microstructural texture (Guerzoni et al., 1994; Brouillaud-Delattre et al., 1997).

In some cases, failure of predictive models is due to insufficient knowledge of bacterial behavior in complex environments such as food matrices. However, particularly over the last few years, a lot of studies have provided better knowledge of new factors that may significantly influence the growth of bacteria. The main problem concerns the quality of these data: they might be imprecise and incomplete. On the one hand, experimental results are often synthesized as intervals of values obtained with or without statistical methods, in order to represent variability, then the data are imprecise. On the other hand, experimental results exist only for a limited amount of substrates, then the data are incomplete, in comparison to the set of substrates on which users may query. The use of an computerized logical reasoning...
may help to manage this incomplete and potentially imprecise information (Dubois and Prade, 1995).

The object of this study is to develop a "logical tool" to include the incomplete and imprecise information provided by the literature in the database of a "decision support system for prediction of microbial risk". Previous works have already proposed such a kind of system.

Zwietering et al., (1992) takes only into account the fact that information is incomplete for food products. He does not propose any solution to represent and manage imprecise information. Estimation is only proposed for growth phenomena using a universal model. Adair and Briggs (1993) presents only the concept and main stages of a dedicated expert system.

Our logical approach suggests a new kind of predictive response. First, we propose a unified framework based on fuzzy logic and theory of possibility which permits to: (i) express the user’s query preferences (and therefore to control the possible enlargement process of database scanning) (ii) represent and manage the imprecise information. Second, the system presents the ability of delivering several hierarchical responses (experimental results obtained from bibliography, unpublished experimental results, …) brought by the database with an indication of the respective degree of matching for each response. Third, the system is able to propose a fuzzified estimated response using predictive models stored in its knowledge base. Fourth, a case based reasoning system identifies the category of queried data. This support system is part of a national project on predictive microbiology in food systems.

The originality of our approach is demonstrated using as an example to L. monocytogenes in this paper. But, our method and our tool are completely independent to any particular bacterium and can be used whatever studied bacterium.

**Materials and methods**
In this paper, we are concerned on one hand with the storage and handling of imprecise information and on the other hand with the expression of querying using fuzzy values to retrieve the more relevant information available in the database. In the first section, we introduce the relational model chosen to structure the information in the database. In the second section, we present the unique framework selected to represent on one hand query preferences and on the other hand precise or imprecise values. In the third section, we present statistical models we use to compute estimated values delivered to the user in addition to the closest information found in the database. In the fourth section, we introduce resemblance relations used to retrieve the closest information stored in the database in addition to query preferences. Finally, in the fifth section, we present the main concepts of the information retrieval system we have designed.

**Information structured using the relational model.**

The relational model is the most popular model used all over the world to represent information in databases. It was proposed by Codd (1970) in the early 70’s and it has been available as a commercial product for 20 years. It manages large quantities of data using a DBMS - Database Management System- such as Oracle for instance as well as personal databases using a DBMS such as Microsoft Access. The basic concept is very simple: elementary information is gathered in relations which represent objects of the modeled world or links between these objects. The concept of relation is implemented as a table. For instance, a bacterium is an object of the world we want to model. Elementary information such as the name of the bacterium, the name of the strain, its identification (unique) number in the database are gathered in a table called *Bacterium* and represent the object Bacterium in the database. These 3 pieces of information are called attributes of the table. A domain of values is associated to each attribute. For instance, the attribute *BacteriumName* must be a string as...
the attribute \textit{BacteriumId} must be a positive integer. All the bacteria represented in the
database are described by the values associated to these 3 attributes and are called tuples of
the table \textit{Bacterium}. Now, if we want to represent the fact that a bacterium produces a
bacteriocin, we will add two new tables. The table called \textit{Bacteriocin} will represent the object
Bacteriocin. Another table will represent the semantic link between the bacterium and the
bacteriocin; this table will be called \textit{IsProducedBy}.

The relational model is built on solid theoretical foundations and useful normalization rules
have been provided to help the administrator of the database to define the table structures.
But, an important drawback of the relational model is that the modeled world is split into a
great number of tables and the database structure composed of a long list of tables is very
complex to understand. This is the reason why the concept of view has been introduced. A
view is a \textit{virtual} table in which all the information needed by the user is brought together. For
instance, we can define the view \textit{BacteriocinProducedByBacterium} which brings together the
information split into the 3 tables \textit{Bacterium}, \textit{Bacteriocin} and \textit{IsProducedBy}. Only the
administrator of the database needs to know the real and entire structure of the database. A
user only knows the view designed by the administrator to access the data he needs.

\textbf{Query preferences and imprecision representation.}

The fuzzy logic theory base approach presented in Zadeh (1965) has been largely used to
represent values associated to selection attributes in order to enhance query flexibility (Cox,
1995; Bosc et al., 1999). More precisely, fuzzy sets are mathematical objects here used with a
semantic of preferences. For example, if one queries a database with the selection attribute
Duration associated to the fuzzy value \textit{HighDuration} (presented in fig. 1), the database engine
will first retrieve information corresponding to Duration value in the interval [50, 70] and
second, with a decreasing degree of preference, information corresponding to duration in the
interval [20, 50[ or ]70, 100].

Formally, for each element in a fuzzy set, there is a grade of membership score that represents
the degree to which the element belongs to the set. These scores are assumed to vary between
0 and 1 inclusive. A 0 indicates that the element has no membership in the set and a 1
indicates that the element has complete, or crisp, membership in the set.

It is a common think to find imprecise data in the bibliography, expressed as intervals of
value, without any reference to the manner it has been obtained. The reason why we have
chosen the possibility theory is that we want to represent imprecision that is characterized by
an inherent vagueness (as high duration) and not uncertainty which occurs when one can
accurately predict the behavior of many similar experiments (Laviolette et al., 1995). The
framework of fuzzy logic restated in the possibility theory base approach presented in Zadeh
(1978) has been used to represent precise or imprecise values of data and to realize fuzzy
pattern matching. A lot of works have focussed on the management of this kind of data in
databases (Dubois and Prade, 1995; Bosc and Pivert, 1997; Bosc and Pivert, 1998; Galindo et
al., 1998).

Representation of precise or imprecise values on continuous or discrete domain are illustrated
here with 3 examples. In Figure 1, the fuzzy set HighDuration represents an imprecise value :
the duration is completely possible in the interval 50-70 seconds, but it is still possible with a
decreasing degree of possibility until the interval 20-100 seconds. Figure 2 illustrates the
imprecise value SterilizedMilk which can be whole sterilized milk with a degree of possibility
of 1 and which may be also skimmed sterilized milk with a degree of possibility of 0.5. The
figure 3 shows that the same framework may also represent a precise value (here the precise
value 80 seconds for the attribute Duration), it may be used to represent both kinds of
information (precise or imprecise).
More formally, the value associated to an attribute $A$ for a tuple $t$ is defined as a possibility distribution $\pi_{A(t)}$ on the underlying domain $D$. The possibility distribution $\pi_{A(t)}$ can be viewed as a fuzzy restriction of the possible value of $A(t)$ by a fuzzy subset $F$. Given $\chi$ the membership function associated to $F$:

$$\left\{ \forall d \in D, \chi_F(d) \in [0,1] \right\} \pi_{A(t)}(d) = \chi_F(d).$$ (1)

For instance, the information « thermisation $t$ has a high duration » will be represented by:

$$\pi_{\text{Duration}(t)}(d) = \chi_{\text{HighDuration}}(d).$$

Here, $\chi_{\text{HighDuration}}$ is a membership function which represents the vague predicate $\text{HighDuration}$ defined as a fuzzy subset. We consider two kinds of membership functions depending on the domain: (i) on an ordered and continuous underlying domain, a trapezoidal fuzzy subset is associated to each value (see fig. 1), (ii) on a discrete and non-ordered domain, a standard fuzzy subset is associated to each value (see fig. 2).

**Statistical models**

Two types of models are used in order to give an estimation of the data sought. The first one concerns growth model and the second one inactivation model.

The growth model comes from a bacterial growth simulation model with two levels (Rosso et al., 1993). Primary level model $F$ (adaptive model) describes how microbial density $x$ changes with time $t$ in a specified environment $x(t)=F(p_i, t)$ where $p_i$ are at most 6 characteristic parameters (maximum growth rate $\mu_{\text{max}}$ or generation time $t_g=\ln 2/\mu_{\text{max}}$, lag phase duration $\lambda$, inoculum $x_0$, maximal density $x_{\text{max}}$, and eventually initial growth rate $\mu_0$, adaptation rate $\rho_a$). Secondary level models indicate how parameters $\mu_{\text{max}}$ and $\lambda$ of primary model change with respect to cardinal values (maximum, minimum and optimal) of temperature $T$ and of pH and a fourth parameter $\mu_{\text{opt}}$ (value of $\mu$ at optimal temperature and optimal pH). With this model we
are able to give a fuzzy estimation of the parameter $\mu_{\text{max}}$ (or the generation time $t_g$) using fuzzy arithmetic (Pedrycz and Gomide, 1998, pp129-150).

Inactivation models (Augustin, 1996) used are linear models: primary modeling is a linear relation between $\log_{10}(\text{cell population } N)$ and time $t : \log_{10}(N)=at+b$, allowing us to define $D=1/a$, time of decimal reduction. Then a linear regression of $\log_{10}(D)$ on the temperature $T$:

$$\log_{10}(D)=cT+d,$$

is calculated from numerous data found in the literature. This model allows us to calculate the estimation of the duration $D$ for a given temperature value and moreover to calculate a fuzzy estimation of it. This calculus takes into account two kinds of variability: the limits of the fuzzy subset representing the user’s preferences and the confidence interval on the coefficients of the regression model (Buche and Dervin, 1999).

Two conditions must be fulfilled to obtain a fuzzy estimation using those both models: (i) values for each model parameter must be expressed in the selection attributes (for example, temperature and pH for the selected growth model and temperature for the selected inactivation model); (ii) values associated to each attribute expressed as fuzzy subset must be included in the limits of the model. If the second condition is not fulfilled, we deliver a precise estimation using precise values for model parameters (defined as the center of the trapezoidal fuzzy subsets).

However, our method is not dependent on those previous models because: first, our system incorporates in its knowledge base predictive models which can be replaced or enhanced if necessary; second, different models may be used for different categories of data (see section The Information Retrieval System for more details).

**Resemblance relations**

---

1 Defined as the set of values where the membership function is equal to 1
If no information stored in the database corresponds to the values specified for the selection attributes, it is still possible to retrieve the closest information if resemblance relations have been defined. Resemblance relations \( \text{res}_i \) are defined on the domain \( D_i \) of an attribute \( i \) which belongs to the query. Given a distance \( d_i \) on the domain \( D_i \), the resemblance relations between \( x \) and \( y \) are defined as follows:

\[
\text{res}_i : D_i \times D_i \rightarrow [0,1] \\
\forall (x, y) \in D_i \times D_i, \text{res}_i(x, y) = \frac{1}{1 + d_i(x, y)}.
\]

We have defined two kinds of distance \( d_i \):

(i) for biological qualitative information (e.g., substrates or serotypes), called concepts in the following, we have defined a semantic distance \( d_i \) computed in generalization/specialization trees, called GS trees. A GS tree is defined as follows : concept \( C_j \) is the father of \( C_2 \) (i.e. \( C_j \) is above \( C_2 \) in the tree) if \( C_j \) is more general than \( C_2 \) (\( C_2 \) is more specific than \( C_j \)). An example of GS tree is given in Fig. 4, representing the distance associated to the attribute Substrate. We drew upon the work of (Foo et al., 1989) to build a semantic distance function on GS trees. Given two concepts \( C_1 \) and \( C_2 \) we consider \( C_3 \), the more specific concept which generalizes \( C_1 \) and \( C_2 \); then the semantic distance \( d_i(C_1, C_2) \) is equal to \( d_i(C_1, C_3) + d_i(C_2, C_3) \). By example, in the fig.4, the distance between concepts Skimmed milk and Whole milk is 2 (\( d_i(\text{Skimmed milk, Sterilized Milk}) + d_i(\text{Whole Milk, Sterilized Milk}) = 1 + 1 = 2 \)) and the distance between Skimmed milk and Raw milk is 4 (\( d_i(\text{Skimmed milk, Milk}) + d_i(\text{Raw Milk, Milk}) = 3 + 1 = 4 \)).

(ii) for information defined on an ordered domain, we have defined a distance \( d_i \) between two values \( x \) and \( y \) of an attribute \( i \), as follows : given an increasing ordered series of values \( \Delta_n \) on the domain \( D_i \) associated to increasing numerical values \( \alpha_n \), \( d_i(x,y) = \alpha_n \) where \( \alpha_n \) is the value associated to \( \Delta_n \) such as \( \Delta_n \leq |x-y| < \Delta_{n+1} \)

Submitted to Special Issue of the International Journal of Food Microbiology
By example, the following values $\alpha_n$ may be associated to the values $\Delta_n$ on the domain $D_i$ of the attribute Temperature

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta_n$</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>...</td>
</tr>
<tr>
<td>$\alpha_n$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
</tr>
</tbody>
</table>

Then for values 2, 3, 7 of attribute Temperature

\[ d_{\text{température}}(2, 3) = 1 \text{ because } \Delta_2 \leq |2-3| < \Delta_3 \]
\[ d_{\text{température}}(3, 7) = 2 \text{ because } \Delta_3 \leq |3-7| < \Delta_4 \]

The information retrieval system

Using a graphical user interface called GUI in the following (see fig 7), the user queries the database specifying selection criteria using the form attribute/value. The value associated to each selection attribute is a fuzzy subset. It represents the user’s preferences. Result attributes are the closest values retrieved from the database for the selection attributes and specific attributes depending on the view through which the query is executed.

The system scans the database through the different views available in its knowledge base and retrieves the results associating an adequation degree to each of them. More precisely, the GUI translates and submits to the engine the user’s preferences as a set of queries $Q_V$. For each available view $V$, the syntax of $Q_V$ is a set $\{V, a_1, \ldots, a_n, <a_{n+1}, V_{n+1}, r_{n+1}, f_{n+1}>, \ldots, <a_m, V_m, r_m, f_m>, nb, t\}$ where $V$ is the name of the view through which the query is executed; $a_1, \ldots, a_n$ are the expected attributes (limited to 1 in the examples), $<a_{n+1}, V_{n+1}, r_{n+1}, f_{n+1}>, \ldots, <a_m, V_m, r_m, f_m>$ are quadruplets defining the selection criteria, nb and t define respectively the maximum number of tuples and the minimum degree of matching of each tuple in the result (t in $[0, 1]$). The quadruplets defining the selection criteria have the following meaning:

For all i in $[n+1, m]$:
• $a_i$ is a selection attribute and $v_i$ is a fuzzy set defining the value associated with $a_i$. We distinguish two kinds of labels depending on the underlying domain of the attribute: discrete or continuous. Two examples of fuzzy sets are given in fig 1 and fig 2.

• $r_i$ is a boolean value specifying to the engine whether the resemblance relations available in the engine knowledge base may be used or not.

• $ft_i$ is a boolean value specifying to the engine whether the users’ preferences expressed in the fuzzy sets may be enlarged or not.

For instance, we consider the following query $Q_{ex}$:

$$V = \text{HeatResistance}$$

$$a_1 = \text{DecimalReductionTime},$$

$$<a_2 = \text{Substrate}, v_2 = \text{MyEggProductPreferences}, r_2 = \text{true}, ft_2 = \text{false}>,$$

$$<a_3 = \text{DecimalReductionTime}, v_3 = \text{HighDuration}, r_3 = \text{false}, ft_3 = \text{true}>,$$

$$nb = 40, t = 0.3$$

The value represented by the fuzzy subset $\text{MyEggProductPreferences} = \{(\text{egg yolk}, 1), (\text{whole egg}, 0.5)\}$ indicates that the user is first interested by information about egg yolk and second by whole egg. The definition of the fuzzy set is made by the user using the GUI. This is the first way to enlarge the query, completely controlled by the user. The second way of enlargement is also controlled by the user: he may activate or deactivate the automatic enlargement based on the set of resemblance relations defined in the previous section and stored in the knowledge base of the engine. The boolean value $r_i$, specified in the above quadruplet definition, permits to control this mechanism. In the above example $Q_{ex}$ ($r_2 = \text{true}$) the engine is authorized to realize enlargement using the resemblance relation defined on the domain of values of the attribute $\text{Substrate}$. The third way of enlargement is also controlled by the user: he may activate or deactivate the automatic enlargement of his preferences based on fuzzy tolerance coefficients defined in the knowledge base of the engine for each category.
of data. The boolean value $f_{l_1}$, specified in the above quadruplet definition, permits to control this mechanism. In the above example $Q_{ex}$ ($f_{l_3} = true$) the engine is authorized to realize enlargement on the user’s preferences represented by the fuzzy set $HighDuration$ for the attribute $DecimalReductionTime$.

In the running version of the system prototype, the selection attributes available for each view are substrate, conditions of the process step (temperature and pH) and bacterium. For the moment, views available concern growth, heat resistance (inactivation) and bacteriocin interaction. In this information retrieval system, we have introduced the new concept of category-based fuzzy view (Buche and Loiseau, 1999; Buche and Dervin, 1999). This notion is an extension of the classical view concept in databases, e.g. a virtual table in which all the information needed by the user is brought together. This view is fuzzy because the selection attributes are expressed as fuzzy predicates as in (Galindo et al., 1998; Zadrozny and Kacprzyk, 1998). This view is category-based because the system identifies the category of queried information (defined in the information retrieval system knowledge base). Three advantages are provided by this category-based reasoning system implemented in the category-based fuzzy view. First, if the user authorized it, the view engine enlarges user’s preferences expressed in his query if they are too restrictive due to the real content of the database: it provides an additional degree of flexibility not available in the previous systems. Second, the view engine is able to propose, in addition to the nearest information retrieved from the database, an estimation of the data sought. For this purpose, statistical models well-adapted to each category of data (choice of the model and associated parameters) are incorporated into the knowledge base of the engine. Third, the view engine optimizes the fuzzy matching processing because the user’s preferences are only compared to the information associated to the category selected: in the previous systems, all the information stored in the queried tables are compared to the user’s preferences.
Adequation between, on one hand, sought values and their category definition and, on the other hand, sought values and values from the database are computed using pattern matching techniques presented in (Dubois and Prade, 1995) and fuzzy object comparison measures proposed in (Bosc and Pivert, 1997). A detailed description of the engine processing can be find in (Buche and Loiseau, 1999).

We have implemented the concept of category-based fuzzy view in a prototype called CFQ (for Category-based Fuzzy Querying). This software is written in Java language. Thus, CFQ is available on the Web. The information is stored in an Oracle DBMS using the imprecision representation model of FSQ (Galindo et al., 1998). A CFQ query is executed using a three-tier process architecture (see fig 8): (i) the CFQ Java client applet running under a usual Web browser, (ii) a database connectivity middleware (running on the Web server due to security checking on applet execution), (iii) an Oracle/FSQ database server. First, the user loads the CFQ engine applet from the Web server on his Web browser. Second, he expresses his preferences using the GUI of CFQ and asks for information scanning. Third, the GUI transfers the user’s query to the CFQ engine. The engine identifies the queried category of data and builds the fuzzy filter. To retrieve the closest tuples stored in the database, the engine submits SQL queries to the Oracle database server through the Web server.

**Results**

In this section, we present four results provided by our system CFQ corresponding to four different examples of queries. Each query illustrates functionalities provided by the system: in example 1, automatic category-based enlargement of data sought by the user to take into account variability of data values stored in the database; in the second and third examples, comparison between estimated values and closest experimental values retrieved from the database may provide on one hand a validation of the estimation or on the other hand
important differences which need to be explained by a sound bibliographical analysis; in the
fourth example, proposal of values unknown by the user but essential to obtain an estimation.
The first example is about the behavior of L. monocytogenes in egg products under
inactivation conditions. It is a sort of process validation query. The user specifies as querying
attributes, substrate, temperature conditions and decimal reduction time of his process and
wants to compare it to the decimal reduction time values stored in the database. Values sought
for decimal reduction time are represented by the fuzzy set HighDuration (see fig 1). Values
sought for temperature are in the interval [60, 64]. Values sought for substrates are
represented by the following fuzzy set called MyEggProductPreferences={(egg yolk, 1),
(whole egg, 0.5)}. It means that the user is first interested by information about egg yolk and
second by information about whole egg. The 3 views available in the system are executed and
only the view HeatResistance provides results (with a adequation degree superior to a
threshold, set arbitrarily to 0.3). The syntax of the query Q generated by the GUI
corresponding to the view HeatResistance is the following:

\[ Q = \{ V=HeatResistance, a_1 = DecimalReductionTime, \]
\[ <a_2=Substrate, v_2=MyEggProductPreferences , r_2= true, f_2=false>, \]
\[ <a_3= DecimalReductionTime, v_3=HighDuration, r_3= true, f_3= true > \]
\[ , nb = 40, t = 0.3 \} \]

In this view, the system has determined that the category of queried data is Egg Product. For
this category, the system has found in its knowledge base an enlargement coefficient (called
fuzzy tolerance coefficient) for the selection attribute DecimalReductionTime. Fig. 5 shows
user’s preferences enlargement for this selection attribute. The values sought for decimal
reduction time are compared to the values stored in the database which are represented as
imprecise values defined by the interval given by the two attributes Duration Min and
Duration Max. Table 2 shows results obtained without or with enlargement of preferences
computed for the decimal reduction time attribute. The first five tuples appear in the two
query results. An additional tuple is provided in the second result set thanks to the
enlargement mechanism even if the threshold set on the adequation degree is the same for the
two queries (set arbitrarily to 0.3 as said previously). This additional datum concerns a
substrate which does not belong to the user’s preferences expressed by
MyEggProductPreferences. Therefore, it has been retrieved thanks to the fuzzy resemblance
relation defined on the domain of values associated to the attribute Substrate and thanks to the
category-based enlargement mechanism adapted to the category EggProduct. In the context of
this example, it may be a relevant piece of information which indicates that the addition of salt
or sugar to eggs provides a high increase in the decimal reduction time.

The second example is about the behavior of L. monocytogenes in dairy products under
growth conditions. The user wants to retrieve the generation time and queries on substrate,
temperature and pH attributes. Values sought are the following : for substrate, the fuzzy set
MyDairyProductPreferences is \{(Sterilized milk, 1)\} which means that user queries sterilized
milk only; for temperature, the fuzzy set MyTemperaturePreferences is the interval [2.2, 2.8]
; for the pH, the fuzzy set MyPHPreferences is the precise value 6.6. The 3 views available in
the system are executed and only the view Growth provides results. The syntax of the query

\( Q \) generated by the GUI corresponding to the view Growth is the following :

\[ Q = \{V = \text{Growth}, a_1 = \text{GenerationTime}, \]
\[ <a_2 = \text{Substrate}, v_2 = \text{MyDairyProductPreferences}, r_2 = \text{true}, ft_2 = \text{false}>, \]
\[ <a_3 = \text{Temperature}, v_3 = \text{MyTemperaturePreferences}, r_3 = \text{true}, ft_3 = \text{false}>, \]
\[ <a_4 = \text{pH}, v_4 = \text{MyPHPreferences}, r_4 = \text{true}, ft_4 = \text{false}> \]
\[ , nb = 40, t = 0.3 \} \]

In this case, the system CFQ is able to compute a fuzzified estimation thanks to the statistical
growth model stored in the system knowledge base for the queried category Dairy Product.
The generation time estimation (in hours) computed with the sought values is the interval [66.18, 105.27]. Simultaneously, CFQ system retrieves the closest experimental data stored in the database (see table 3). The closest experimental value retrieved from the database (identified by number 33) is out of the estimated interval of values (duration = 28.5 hours). This value has been published in Walker et al. (1990), this bibliographical reference may also be provided by the querying system. In this case, this experimental value tends to invalidate the estimation.

The third example is about the behavior of L. monocytogenes in dairy products under inactivation conditions. The user queries using substrate and temperature as selection attributes. Values sought are the following: for substrate, the fuzzy set MyDairyProductPreferences is {(Raw milk, 1)}; for temperature, the value is defined by the fuzzy set ImpreciseTemperature (see fig. 6). The 3 views available in the system are executed and only the view HeatResistance (due essentially to the temperature values) provides results. The syntax of the query $Q$ generated by the GUI corresponding to the view HeatResistance is the following:

$$Q = \{V=HeatResistance, a_1 = DecimalReductionTime, $$

$$<a_2=Substrate, v_2=MyDairyProductPreferences, r_2 = true, ft_2 = false>, $$

$$<a_3=Temperature, v_3=ImpreciseTemperature, r_3 = true, ft_3 = false> $$

$$, nb = 40, t = 0.3\}$$

As in the previous example, the CFQ system is also able to compute a fuzzified estimation thanks to the statistical inactivation model stored in its knowledge base for the category of queried data, here Dairy Products. The fuzzy kernel (values with a membership degree equal to 1) of the estimated decimal reduction time (expressed in seconds) is the interval [4.83, 287.18]. The fuzzy support (values with a membership degree > 0) of the estimated decimal reduction time is the interval [1.99, 697.99]. Most experimental values retrieved from the
database (see table 4) are included in the fuzzy estimation support. Only one experimental value (ID = 8 in table 4) is far from the kernel (but still belongs to the support). Therefore, the great majority of experimental values confirm the estimation.

The fourth example is about the behavior of *L. monocytogenes* in meat products under growth conditions. In this case, we suppose that the user only knows the values for the substrate and the temperature. He has no idea about the value of the pH. It is then possible to use the CFQ system to perform an initial screening to retrieve possible values for the unknown parameter (here the pH) and a second scan to obtain simultaneously an estimated value of the growth (which can be computed only if temperature and pH are simultaneously provided in the query) and a comparison with the closest experimental values.

For instance, in the first query, values sought are the following: for substrate, the fuzzy set *MyMeatProductPreferences* is \{(Cuts of beef, 1)\}; for temperature, the fuzzy set *MyTemperaturePreferences* is the precise value 4°C. The 3 views available in the system are executed and only the view *Growth* provides results (due to the temperature values). The syntax of the query *Q* generated by the GUI corresponding to the view *Growth* is the following:

\[
Q = \{V = \text{Growth}, a_1 = \text{pH}, a_2 = \text{GenerationTime} \\
\quad <a_3 = \text{Substrate}, v_3 = \text{MyMeatProductPreferences}, r_3 = \text{true}, ft_3 = \text{false}> , \\
\quad <a_4 = \text{Temperature}, v_4 = \text{MyTemperaturePreferences}, r_4 = \text{true}, ft_4 = \text{false}> , \\
\quad , nb = 40, t = 0.3\}
\]

The results of the interrogation are given in table 5. In the entire set of experiments, the pH is included in the interval [5.5, 6.7]. In the second query, the user considers the same values for substrate and temperature and adds the first screening for the pH value, i.e. the fuzzy set...
The syntax of the query $Q$ generated by the GUI corresponding to the view $Growth$ is the following:

$$Q = \{ V = \text{Growth}, a_1 = \text{GenerationTime},$$

$$<a_2 = \text{Substrate}, v_2 = \text{MyMeatProductPreferences}, r_2 = \text{true}, ft_2 = \text{false}>,$$

$$<a_3 = \text{Temperature}, v_3 = \text{MyTemperaturePreferences}, r_3 = \text{true}, ft_3 = \text{false}>,$$

$$<a_4 = \text{pH}, v_4 = \text{MypHPreferences}, r_4 = \text{true}, ft_4 = \text{false}>,$$

$$, nb = 40, t = 0.3 \}$$

The estimated generation time (expressed in hours) obtained during the second screening is the interval $[6.38, 76.3]$. The closest set of experimental values is identical to the one obtained in the first screening.

**Discussion**

One of the great difficulties in predicting microbial behavior in complex food matrix concerns the fact that we have to take into account the variability of biological systems (bacteria and food matrices) in a large range of environmental conditions (experimental or industrial conditions). Moreover, in most cases, we are not able to determine all of the factors that could interfere with the bacterial response. First, quantification cannot be done precisely (chemical or physical gradients are not easy to establish), second, we do not always know the influence of one factor (ingredient, other bacteria, physical conditions, etc.) on bacterial behavior. For example, the ability of $L. \text{monocytogenes}$ to grow in many food products is mainly evaluated with regards to a small number of factors such as temperature, pH, etc. Nevertheless, other factors (physiological, biological or other physical factors) could significantly modify the growth response of $L. \text{monocytogenes}$. Because our knowledge in microbiology and particularly in microbial ecology is growing, predictive microbiology models will be able to take into account this new information in order to improve prediction. This new knowledge is incomplete and/or can be imprecise. Our information retrieval system allows us to take into
account these data that can (i) modulate or specify the prediction obtained with mathematical models or (ii) can propose a response when mathematical models have failed to calculate growth or inactivation parameters.

We have also to mention two main drawbacks the user will have to manage with this new querying approach. First, the adequation degree of a retrieved information may be the combining result of three different enlargement mechanisms (user’s preferences expression, category-based enlargement of user’s preferences, enlargement by resemblance relation definition on the domain of values). Then, it is not easy to explain to the user how it has been computed by the system. Second, the user has to be careful if he realizes comparisons between closest experimental data retrieved from the database and fuzzy estimation using statistical models. He has to check that retrieved experimental data are close enough compared to his selection criteria, knowing that the adequation degree only provides a relative ranking of retrieved information. For instance, in the result of the second query presented in the previous section, only the first closest information (ID 33) checks completely the selection criteria and can be really used to be compared to the fuzzified estimation.

**Conclusion**

In this paper, we have introduced the concept of category-based fuzzy view to design an information retrieval system adapted to microbial risk analysis of food products. In the query, the user has to specify his fuzzy preferences. Then the CFQ engine scans the database through the views available in its knowledge base. For each view, the engine builds a fuzzy filter with the user preferences adapted to the category of queried data - thanks to knowledge associated to each category. It brings three benefits: (i) The CFQ engine optimizes the processing time of the filtering because it only scans the data associated to the selected category; (ii) the CFQ engine can deliver an estimate of the queried data if an estimation method has been defined for the selected category of data; (iii) the CFQ engine can enlarge users’ preferences to adapt

Submitted to Special Issue of the *International Journal of Food Microbiology*
them to the real content of the database. The CFQ engine has been developed in Java
language. Therefore, the information retrieval system is available through the WEB. It has
been coupled to Oracle extended for fuzzy SQL (Buche and Loiseau, 1999; Buche and Dervin,
1999). CFQ is complementary with classical simulation software because :
- It can be used when classical software does not work; for example when parameters are
  outside the limits of the model.
- It allows fuzzy expression of the parameters (selection attribute values) and takes into
  account this imprecision in the database scan and in the calculation of the estimated
  values.
- It allows comparison between estimated and experimental data, underlining possible
  contradictions and then acting as a basic tool to understand unknown phenomena.
- It may provide information on an original domain, such as interaction between bacteria.
- It produces synthesized information, and provides bibliographical references to the
  corresponding publications.
- It is a flexible tool able to evolve by incorporating new experimental results and/or new
  simulation models.

In order to improve this information retrieval system, it is now necessary to add data that
come from more complete bibliography references. In the future, the database will have to
integrate other pathogens in order to carry out a better microbial prediction of food products,
predictive microbiology being an interesting tool for risk analysis.

Acknowledgements
This work was partially supported by the French Ministry of Agriculture’s Food Department
(DGAL) and Research and Education Department (DGER).

References


Figure 1. A trapezoidal fuzzy subset representing an imprecise value

(MD stands for Membership function)

Figure 2. A fuzzy subset defined on a discrete domain

Figure 3. A fuzzy subset representing a crisp (precise) value

Figure 4. Example of Generalization-Specialization (GS) tree for milk products
Figure 5. Enlargement mechanism of the user’s preferences using fuzzy tolerance coefficient for the category EggProduct (first example of query)

Figure 6. Temperature value for the third query
Figure 7. The Graphical User Interface permits to express the selection criteria of a query and to display the results. It is organized as a thumbnail formular. In the above example, the selected thumbnail shows the value (expressed as a fuzzy set) associated by the user to the attribute temperature.
Figure 8. Category-based Fuzzy Querying engine is a software based on a three tier architecture: the Java applet is available on the Web server, the Java applet is loaded on the client to run in the Web browser, the CFQ engine running in the client applet queries the database through the Web server to retrieve the closest information.
Table 1. View BacteriocinProducedByBacterium

<table>
<thead>
<tr>
<th>BACTERIOCIN NAME</th>
<th>BACTERIUM NAME</th>
<th>STRAIN NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pediocin ACH</td>
<td>Lactobacillus Plantarum</td>
<td>WHE92</td>
</tr>
<tr>
<td>Pediocin ACH</td>
<td>Pediococcus Acidilacti</td>
<td>H</td>
</tr>
</tbody>
</table>
Table 2. Closest data retrieved through the view *Heat Resistance* corresponding to the first query without enlargement of preferences (Adequation Degree (AD) > 0.3) are the first five tuples. With enlargement of preferences, same first five tuples are obtained plus the additionnal sixth tuple.

<table>
<thead>
<tr>
<th>ID</th>
<th>AD</th>
<th>SUBSTRATE</th>
<th>TEMP (°C)</th>
<th>DURATION MIN (Sec)</th>
<th>DURATION MAX (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>0.905</td>
<td>egg yolk</td>
<td>60</td>
<td>55.2</td>
<td>101.4</td>
</tr>
<tr>
<td>82</td>
<td>0.905</td>
<td>egg yolk</td>
<td>60</td>
<td>55.2</td>
<td>101.4</td>
</tr>
<tr>
<td>85</td>
<td>0.786</td>
<td>egg yolk</td>
<td>64.4</td>
<td>26.4</td>
<td>26.4</td>
</tr>
<tr>
<td>78</td>
<td>0.5</td>
<td>whole egg</td>
<td>60</td>
<td>78</td>
<td>84</td>
</tr>
<tr>
<td>81</td>
<td>0.5</td>
<td>whole egg</td>
<td>60</td>
<td>90</td>
<td>126</td>
</tr>
<tr>
<td>86</td>
<td>0.319</td>
<td>egg + salt or sugar</td>
<td>64.4</td>
<td>366.6</td>
<td>366.6</td>
</tr>
</tbody>
</table>
Table 3. Closest data retrieved through the view Growth corresponding to the second query

(AD means Adequation Degree)

<table>
<thead>
<tr>
<th>ID</th>
<th>AD</th>
<th>SUBSTRAT</th>
<th>TEMP</th>
<th>PH</th>
<th>GENERATION TIME (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>0.76</td>
<td>Sterilized_milk_(UHT)</td>
<td>2.5</td>
<td>6.6</td>
<td>28.5</td>
</tr>
<tr>
<td>4</td>
<td>0.35</td>
<td>Skimmed_milk</td>
<td>4</td>
<td>6.4</td>
<td>32.5</td>
</tr>
<tr>
<td>7</td>
<td>0.35</td>
<td>Skimmed_milk</td>
<td>4</td>
<td>6.4</td>
<td>30.6</td>
</tr>
<tr>
<td>10</td>
<td>0.35</td>
<td>Skimmed_milk</td>
<td>4</td>
<td>6.4</td>
<td>37.5</td>
</tr>
<tr>
<td>13</td>
<td>0.35</td>
<td>Skimmed_milk</td>
<td>4</td>
<td>6.4</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>0.35</td>
<td>Skimmed_milk</td>
<td>4</td>
<td>6.6</td>
<td>48.9</td>
</tr>
<tr>
<td>17</td>
<td>0.35</td>
<td>Skimmed_milk</td>
<td>4</td>
<td>6.6</td>
<td>53.5</td>
</tr>
<tr>
<td>20</td>
<td>0.35</td>
<td>Sterilized_whole_milk</td>
<td>4</td>
<td>6.4</td>
<td>30.6</td>
</tr>
<tr>
<td>23</td>
<td>0.35</td>
<td>Sterilized_whole_milk</td>
<td>4</td>
<td>6.4</td>
<td>30.5</td>
</tr>
<tr>
<td>26</td>
<td>0.35</td>
<td>Sterilized_whole_milk</td>
<td>4</td>
<td>6.4</td>
<td>36</td>
</tr>
<tr>
<td>29</td>
<td>0.35</td>
<td>Sterilized_whole_milk</td>
<td>4</td>
<td>6.4</td>
<td>36.5</td>
</tr>
</tbody>
</table>
Table 4. Closest experimental values for the third query

(AD means Adequation Degree)

<table>
<thead>
<tr>
<th>ID</th>
<th>AD</th>
<th>SUBSTRATE</th>
<th>TEMP</th>
<th>DURATION MIN (sec)</th>
<th>DURATION MAX (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.76</td>
<td>Raw_milk</td>
<td>63.3</td>
<td>19.8</td>
<td>19.8</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>Milk</td>
<td>61.7</td>
<td>570</td>
<td>570</td>
</tr>
<tr>
<td>11</td>
<td>0.2</td>
<td>Raw_milk</td>
<td>71.7</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>12</td>
<td>0.2</td>
<td>Raw_milk</td>
<td>71.7</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td>13</td>
<td>0.2</td>
<td>Skimmed_milk</td>
<td>62.7</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>14</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>62.7</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>15</td>
<td>0.2</td>
<td>Half_skimmed_milk</td>
<td>62.7</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>16</td>
<td>0.2</td>
<td>Half_skimmed_milk</td>
<td>62</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>17</td>
<td>0.2</td>
<td>Half_skimmed_milk</td>
<td>62</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>18</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>63.3</td>
<td>19.8</td>
<td>34.8</td>
</tr>
<tr>
<td>19</td>
<td>0.2</td>
<td>Skimmed_milk</td>
<td>63.3</td>
<td>25.8</td>
<td>28.2</td>
</tr>
<tr>
<td>21</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>3.1</td>
<td>3.1</td>
</tr>
<tr>
<td>22</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>23</td>
<td>0.2</td>
<td>Half_skimmed_milk</td>
<td>72</td>
<td>0.05</td>
<td>1.5</td>
</tr>
<tr>
<td>24</td>
<td>0.2</td>
<td>Half_skimmed_milk</td>
<td>72</td>
<td>0.02</td>
<td>1.4</td>
</tr>
<tr>
<td>25</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>2.7</td>
<td>2.7</td>
</tr>
<tr>
<td>26</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>27</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>28</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>30</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>31</td>
<td>0.2</td>
<td>Sterilized_whole_milk</td>
<td>71.7</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>32</td>
<td>0.2</td>
<td>Pasteurized_whole_milk</td>
<td>57.8</td>
<td>138</td>
<td>138</td>
</tr>
</tbody>
</table>
Table 5. Closest experimental values for the fourth query  
(AD means Adequation Degree)

<table>
<thead>
<tr>
<th>ID</th>
<th>AD</th>
<th>SUBSTRATE</th>
<th>TEMP</th>
<th>PH</th>
<th>GENERATION TIME (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>0.58</td>
<td>Cuts_of_fat_bee</td>
<td>4.7</td>
<td>5.6</td>
<td>11.9</td>
</tr>
<tr>
<td>81</td>
<td>0.5</td>
<td>Cuts_of_lean_bee</td>
<td>5</td>
<td>6.3</td>
<td>17.1</td>
</tr>
<tr>
<td>79</td>
<td>0.4</td>
<td>Cuts_of_lean_bee</td>
<td>5.5</td>
<td>5.5</td>
<td>22.1</td>
</tr>
<tr>
<td>77</td>
<td>0.38</td>
<td>Cuts_of_lean_bee</td>
<td>5.6</td>
<td>5.6</td>
<td>22.9</td>
</tr>
<tr>
<td>87</td>
<td>0.24</td>
<td>Cuts_of_fat_bee</td>
<td>7.5</td>
<td>5.6</td>
<td>7.1</td>
</tr>
<tr>
<td>85</td>
<td>0.21</td>
<td>Cuts_of_lean_bee</td>
<td>10</td>
<td>6.7</td>
<td>4.08</td>
</tr>
<tr>
<td>88</td>
<td>0.20</td>
<td>Cuts_of_fat_bee</td>
<td>10.1</td>
<td>5.6</td>
<td>4.62</td>
</tr>
</tbody>
</table>