# Case Base Mining for Adaptation Knowledge Acquisition

**M. d'Aquin[1, 2], F. Badra[1], S. Lafrogne[1], J. Lieber[1], A. Napoli[1], L. Szathmary[1]**

[1] LORIA (CNRS, INRIA, Nancy Universities) BP 239, 54506 Vandœuvre-lès-Nancy, France,
{daquin, badra, lafrogne, lieber, napoli, szathmar}@loria.fr

[2] Knowledge Media Institute (KMi), the Open University, Milton Keynes, United Kingdom,
m.daquin@open.ac.uk

## Abstract

In case-based reasoning, the adaptation of a source case in order to solve the target problem is at the same time crucial and difficult to implement. The reason for this difficulty is that, in general, adaptation strongly depends on domain-dependent knowledge. This fact motivates research on adaptation knowledge acquisition (AKA). This paper presents an approach to AKA based on the principles and techniques of knowledge discovery from databases and data-mining. It is implemented in CABAMAKA, a system that explores the variations within the case base to elicit adaptation knowledge. This system has been successfully tested in an application of case-based reasoning to decision support in the domain of breast cancer treatment.

## 1 Introduction

Case-based reasoning (CBR [Riesbeck and Schank, 1989]) aims at solving a target problem thanks to a case base. A case represents a previously solved problem and may be seen as a pair (problem, solution). A CBR system selects a case from the case base and then adapts the associated solution, requiring domain-dependent knowledge for adaptation. The goal of adaptation knowledge acquisition (AKA) is to detect and extract this knowledge. This is the function of the semi-automatic system CABAMAKA, which applies principles of knowledge discovery from databases (KDD) to AKA, in particular frequent itemset extraction. This paper presents the system CABAMAKA: its principles, its implementation and an example of adaptation rule discovered in the framework of an application to breast cancer treatment. The originality of CABAMAKA lies essentially in the approach of AKA that uses a powerful learning technique that is guided by a domain expert, according to the spirit of KDD. This paper proposes an original and working approach to AKA, based on KDD techniques. In addition, the KDD process is performed on a knowledge base itself, leading to the extraction of *meta-knowledge*, i.e. knowledge units for manipulating other knowledge units. This is also one of the rare papers trying to build an effective bridge between knowledge discovery and case-based reasoning.

The paper is organized as follows. Section 2 presents basic notions about CBR and adaptation. Section 3 summarizes researches on AKA. Section 4 describes the system CABAMAKA: its main principles, its implementation and examples of adaptation knowledge acquired from it. Finally, section 5 draws some conclusions and points out future work.

## 2 CBR and Adaptation

A case in a given CBR application encodes a problem-solving episode that is represented by a problem statement $\texttt{pb}$ and an associated solution $\texttt{Sol}(\texttt{pb})$. The case is denoted by the pair $(\texttt{pb}, \texttt{Sol}(\texttt{pb}))$ in the following. Let $\texttt{Problems}$ and $\texttt{Solutions}$ be the set of problems and the set of solutions of the application domain, and "is a solution of" be a binary relation on $\texttt{Problems} \times \texttt{Solutions}$. In general, this relation is not known in the whole but at least a finite number of its instances $(\texttt{pb}, \texttt{Sol}(\texttt{pb}))$ is known and constitutes the case base $\texttt{CB}$. An element of $\texttt{CB}$ is called a *source case* and is denoted by $\texttt{srce-case} = (\texttt{srce}, \texttt{Sol}(\texttt{srce}))$, where $\texttt{srce}$ is a *source problem*. In a particular CBR session, the problem to be solved is called *target problem*, denoted by $\texttt{tgt}$.

A case-based inference associates to $\texttt{tgt}$ a solution $\texttt{Sol}(\texttt{tgt})$, with respect to the case base $\texttt{CB}$ and to additional knowledge bases, in particular $\mathcal{O}$, the *domain ontology* (also known as domain theory or domain knowledge) that usually introduces the concepts and terms used to represent the cases. It can be noticed that the research work presented in this paper is based on the assumption that there exists a domain ontology associated with the case base, in the spirit of knowledge-intensive CBR [Aamodt, 1990].

A classical decomposition of CBR relies on the steps of retrieval and adaptation. *Retrieval* selects $(\texttt{srce}, \texttt{Sol}(\texttt{srce})) \in \texttt{CB}$ such that $\texttt{srce}$ is similar to $\texttt{tgt}$ according to some similarity criterion. The goal of adaptation is to solve $\texttt{tgt}$ by modifying $\texttt{Sol}(\texttt{srce})$ accordingly. Thus, the profile of the adaptation function is

$$\texttt{Adaptation} : ((\texttt{srce}, \texttt{Sol}(\texttt{srce})), \texttt{tgt}) \mapsto \texttt{Sol}(\texttt{tgt})$$

The work presented hereafter is based on the following model of adaptation, similar to *transformational analogy* [Carbonell, 1983]:

① $(\texttt{srce}, \texttt{tgt}) \mapsto \Delta\texttt{pb}$, where $\Delta\texttt{pb}$ encodes the similarities and dissimilarities of the problems $\texttt{srce}$ and $\texttt{tgt}$.

② $(\Delta\texttt{pb}, \texttt{AK}) \mapsto \Delta\texttt{sol}$, where AK is the adaptation knowledge and where $\Delta\texttt{sol}$ encodes the similarities and dissimilarities of $\texttt{Sol}(\texttt{srce})$ and the forthcoming $\texttt{Sol}(\texttt{tgt})$.

③ $(\texttt{Sol}(\texttt{srce}), \Delta\texttt{sol}) \mapsto \texttt{Sol}(\texttt{tgt})$, $\texttt{Sol}(\texttt{srce})$ is modified into $\texttt{Sol}(\texttt{tgt})$ according to $\Delta\texttt{sol}$.

Adaptation is generally supposed to be domain-dependent in the sense that it relies on domain-specific adaptation knowledge. Therefore, this knowledge has to be acquired. This is the purpose of *adaptation knowledge acquisition* (AKA).

## 3 Related Work in AKA

The notion of adaptation case is introduced in [Leake *et al.*, 1996]. The system DIAL is a case-based planner in the domain of disaster response planning. Disaster response planning is the initial strategic planning used to determine how to assess damage, evacuate victims, etc. in response to natural and man-made disasters such as earthquakes and chemical spills. To adapt a case, the DIAL system performs either a case-based adaptation or a rule-based adaptation. The case-based adaptation attempts to retrieve an adaptation case describing the successful adaptation of a similar previous adaptation problem. An adaptation case represents an adaptation as the combination of transformations (e.g. addition, deletion, substitution) plus memory search for the knowledge needed to operationalize the transformation (e.g. to find what to add or substitute), thus reifying the principle: *adaptation = transformations + memory search*. An adaptation case in DIAL packages information about the context of an adaptation, the derivation of its solution, and the effort involved in the derivation process. The context information includes characteristics of the problem for which adaptation was generated, such as the type of problem, the value being adapted, and the roles that value fills in the response plan. The derivation records the operations needed to find appropriate values in memory, e.g. operations to extract role-fillers or other information to guide the memory search process. Finally, the effort records the actual effort expended to find the solution path. It can be noticed that the core idea of "transformation" is also present in our own adaptation knowledge extraction.

In [Jarmulak *et al.*, 2001], an approach to AKA is presented that produces a set of *adaptation cases*, where an adaptation case is the representation of a particular adaptation process. The adaptation case base, $\texttt{CB}_\texttt{A}$, is then used for further adaptation steps: an adaptation step itself is based on CBR, reusing the adaptation cases of $\texttt{CB}_\texttt{A}$. $\texttt{CB}_\texttt{A}$ is built as follows. For each $(\texttt{srce}_1, \texttt{Sol}(\texttt{srce}_1)) \in \texttt{CB}$, the retrieval step of the CBR system using the case base CB without $(\texttt{srce}_1, \texttt{Sol}(\texttt{srce}_1))$ returns a case $(\texttt{srce}_2, \texttt{Sol}(\texttt{srce}_2))$. Then, an adaptation case is built based on both source cases and is added to $\texttt{CB}_\texttt{A}$. This adaptation case encodes $\texttt{srce}_1$, $\texttt{Sol}(\texttt{srce}_1)$, the difference between $\texttt{srce}_1$ and $\texttt{srce}_2$ ($\Delta\texttt{pb}$, with the notations of this paper) and the difference between $\texttt{Sol}(\texttt{srce}_1)$ and $\texttt{Sol}(\texttt{srce}_2)$ ($\Delta\texttt{sol}$). This approach to AKA and CBR has been successfully tested for an application to the design of tablet formulation.

The idea of the research presented in [Hanney and Keane, 1996; Hanney, 1997] is to exploit the variations between source cases to learn adaptation rules. These rules compute variations on solutions from variations on problems. More precisely, ordered pairs $(\texttt{srce-case}_1, \texttt{srce-case}_2)$ of similar source cases are formed. Then, for each of these pairs, the variations between the problems $\texttt{srce}_1$ and $\texttt{srce}_2$ and the solutions $\texttt{Sol}(\texttt{srce}_1)$ and $\texttt{Sol}(\texttt{srce}_2)$ are represented ($\Delta\texttt{pb}$ and $\Delta\texttt{sol}$). Finally, the adaptation rules are learned, using as training set the set of the input-output pairs $(\Delta\texttt{pb}, \Delta\texttt{sol})$. This approach has been tested in two domains: the estimation of the price of flats and houses, and the prediction of the rise time of a servo mechanism. The experiments have shown that the CBR system using the adaptation knowledge acquired from the automatic system of AKA shows a better performance compared to the CBR system working without adaptation. This research has influenced our work that is globally based on similar ideas.

[Shiu *et al.*, 2001] proposes a method for case base maintenance that reduces the case base to a set of representative cases together with a set of general adaptation rules. These rules handle the perturbation between representative cases and the other ones. They are generated by a fuzzy decision tree algorithm using the pairs of similar source cases as a training set.

In [Wiratunga *et al.*, 2002], the idea of [Hanney and Keane, 1996] is reused to extend the approach of [Jarmulak *et al.*, 2001]: some learning algorithms (in particular, C4.5) are applied to the adaptation cases of $\texttt{CB}_\texttt{A}$, to induce general adaptation knowledge.

These approaches to AKA share the idea of exploiting adaptation cases. For some of them ([Jarmulak *et al.*, 2001; Leake *et al.*, 1996]), the adaptation cases themselves constitute the adaptation knowledge (and adaptation is itself a CBR process). For the other ones ([Hanney and Keane, 1996; Shiu *et al.*, 2001; Wiratunga *et al.*, 2002]), as for the approach presented in this paper, the adaptation cases are the input of a learning process.

## 4 CABAMAKA

We now present the CABAMAKA system, for acquiring adaptation knowledge. The CABAMAKA system is at present working in the medical domain of cancer treatment, but it may be reused in other application domains where there exist problems to be solved by a CBR system.

### 4.1 Principles

CABAMAKA deals with case base mining for AKA. Although the main ideas underlying CABAMAKA are shared with those presented in [Hanney and Keane, 1996], the followings are original ones. The adaptation knowledge that is mined has to be validated by experts and has to be associated with explanations making it understandable by the user. In this way, CABAMAKA may be considered as a semi-automated (or interactive) learning system. This is a necessary requirement for the medical domain for which CABAMAKA has been initially designed.

Moreover, the system takes into account every ordered pair $(\texttt{srce-case}_1, \texttt{srce-case}_2)$ with $\texttt{srce-case}_1 \neq$

$\texttt{srce-case}_2$, leading to examine $n(n-1)$ pairs of cases for a case base CB where $|\texttt{CB}| = n$. In practice, this number may be rather large since in the present application $n \simeq 650$ ($n(n-1) \simeq 4 \cdot 10^5$). This is one reason for choosing for this system efficient KDD techniques such as CHARM [Zaki and Hsiao, 2002]. This is different from the approach of [Hanney and Keane, 1996], where only pairs of *similar* source cases are considered, according to a fixed criterion. In CABAMAKA, there is no similarity criterion on which a selection of pairs of cases to be compared could be carried out. Indeed, the CBR process in CABAMAKA relies on the adaptation-guided retrieval principle [Smyth and Keane, 1996], where only adaptable cases are retrieved. Thus, every pair of cases may be of interest, and two cases may appear to be similar w.r.t. a given point of view, and dissimilar w.r.t. another one.

**Principles of KDD.** The goal of KDD is to discover knowledge from databases, under the supervision of an analyst (expert of the domain). A KDD session usually relies on three main steps: data preparation, data-mining, and interpretation of the extracted pieces of information.

*Data preparation* is mainly based on formatting and filtering operations. The formatting operations are used to transform the data into a form allowing the application of the chosen data-mining operations. The filtering operations are used for removing noisy data and for focusing the data-mining operation on special subsets of objects and/or attributes.

*Data-mining* algorithms are applied for extracting from data information units showing some regularities [Hand *et al.*, 2001]. In the present experiment, the CHARM data-mining algorithm that efficiently performs the extraction of *frequent closed itemsets* (*FCIs*) has been used [Zaki and Hsiao, 2002]. CHARM inputs a formal database, i.e. a set of binary *transactions*, where each transaction $T$ is a set of binary *items*. An *itemset* $I$ is a set of items, and the support of $I$, $\texttt{support}(I)$, is the proportion of transactions $T$ of the database possessing $I$ ($I \subseteq T$). $I$ is frequent, with respect to a threshold $\sigma \in [0;1]$, whenever $\texttt{support}(I) \geq \sigma$. $I$ is closed if it has no proper superset $J$ ($I \subsetneq J$) with the same support.

The *interpretation* step aims at interpreting the extracted pieces of information, i.e. the FCIs in the present case, with the help of an analyst. In this way, the interpretation step produces new knowledge units (e.g. rules).

The CABAMAKA system relies on these main KDD steps as explained below.

**Formatting.** The formatting step of CABAMAKA inputs the case base CB and outputs a set of transactions obtained from the pairs $(\texttt{srce-case}_1, \texttt{srce-case}_2)$. It is composed of two substeps. During the first substep, each $\texttt{srce-case} = (\texttt{srce}, \texttt{Sol}(\texttt{srce})) \in \texttt{CB}$ is formatted in two sets of boolean properties: $\Phi(\texttt{srce})$ and $\Phi(\texttt{Sol}(\texttt{srce}))$. The computation of $\Phi(\texttt{srce})$ consists in translating srce from the problem representation formalism to $2^{\mathcal{P}}$, $\mathcal{P}$ being a set of boolean properties. Some information may be lost during this translation, for example, when translating a continuous property into a set of boolean properties, but this loss has to be minimized. Now, this translation formats an expression srce expressed in the framework of the domain ontology $\mathcal{O}$ to an expression $\Phi(\texttt{srce})$ that will be manipulated as data, i.e. without the use of a reasoning process. Therefore, in order to minimize the translation loss, it is assumed that

$$\text{if } p \in \Phi(\texttt{srce}) \text{ and } p \models_{\mathcal{O}} q \quad \text{then } q \in \Phi(\texttt{srce}) \qquad (1)$$

for each $p, q \in \mathcal{P}$ (where $p \models_{\mathcal{O}} q$ stands for "$q$ is a consequence of $p$ in the ontology $\mathcal{O}$"). In other words, $\Phi(\texttt{srce})$ is assumed to be deductively closed given $\mathcal{O}$ in the set $\mathcal{P}$. The same assumption is made for $\Phi(\texttt{Sol}(\texttt{srce}))$. How this first substep of formatting is computed in practice depends heavily on the representation formalism of the cases and is presented, for our application, in section 4.2.

The second substep of formatting produces a transaction $T = \Phi((\texttt{srce-case}_1, \texttt{srce-case}_2))$ for each ordered pair of distinct source cases, based on the sets of items $\Phi(\texttt{srce}_1)$, $\Phi(\texttt{srce}_2)$, $\Phi(\texttt{Sol}(\texttt{srce}_1))$ and $\Phi(\texttt{Sol}(\texttt{srce}_2))$. Following the model of adaptation presented in section 2 (items ①, ② and ③), $T$ has to encode the properties of $\Delta\texttt{pb}$ and $\Delta\texttt{sol}$. $\Delta\texttt{pb}$ encodes the similarities and dissimilarities of $\texttt{srce}_1$ and $\texttt{srce}_2$, i.e.:

- The properties common to $\texttt{srce}_1$ and $\texttt{srce}_2$ (marked by "="),
- The properties of $\texttt{srce}_1$ that $\texttt{srce}_2$ does not share ("−"), and
- The properties of $\texttt{srce}_2$ that $\texttt{srce}_1$ does not share ("+").

All these properties are related to problems and thus are marked by pb. $\Delta\texttt{sol}$ is computed in a similar way and $\Phi(T) = \Delta\texttt{pb} \cup \Delta\texttt{sol}$. For example,

$$\text{if } \begin{cases} \Phi(\texttt{srce}_1) = \{a, b, c\} & \Phi(\texttt{Sol}(\texttt{srce}_1)) = \{A, B\} \\ \Phi(\texttt{srce}_2) = \{b, c, d\} & \Phi(\texttt{Sol}(\texttt{srce}_2)) = \{B, C\} \end{cases}$$

$$\text{then } \quad T = \left\{a^{-}_{\texttt{pb}}, b^{=}_{\texttt{pb}}, c^{=}_{\texttt{pb}}, d^{+}_{\texttt{pb}}, A^{-}_{\texttt{sol}}, B^{=}_{\texttt{sol}}, C^{+}_{\texttt{sol}}\right\} \qquad (2)$$

More generally:

$$\begin{aligned} T = &\{p^{-}_{\texttt{pb}} \mid p \in \Phi(\texttt{srce}_1) \backslash \Phi(\texttt{srce}_2)\} \\ &\cup \{p^{=}_{\texttt{pb}} \mid p \in \Phi(\texttt{srce}_1) \cap \Phi(\texttt{srce}_2)\} \\ &\cup \{p^{+}_{\texttt{pb}} \mid p \in \Phi(\texttt{srce}_2) \backslash \Phi(\texttt{srce}_1)\} \\ &\cup \{p^{-}_{\texttt{sol}} \mid p \in \Phi(\texttt{Sol}(\texttt{srce}_1)) \backslash \Phi(\texttt{Sol}(\texttt{srce}_2))\} \\ &\cup \{p^{=}_{\texttt{sol}} \mid p \in \Phi(\texttt{Sol}(\texttt{srce}_1)) \cap \Phi(\texttt{Sol}(\texttt{srce}_2))\} \\ &\cup \{p^{+}_{\texttt{sol}} \mid p \in \Phi(\texttt{Sol}(\texttt{srce}_2)) \backslash \Phi(\texttt{Sol}(\texttt{srce}_1))\} \end{aligned}$$

**Filtering.** The filtering operations may take place before, between and after the formatting substeps, and also after the mining step. They are guided by the analyst.

**Mining.** The extraction of FCIs is computed thanks to CHARM (in fact, thanks to a tool based on a CHARM-like algorithm) from the set of transactions. A transaction $T = \Phi((\texttt{srce-case}_1, \texttt{srce-case}_2))$ encodes a specific adaptation $((\texttt{srce}_1, \texttt{Sol}(\texttt{srce}_1)), \texttt{srce}_2) \mapsto \texttt{Sol}(\texttt{srce}_2)$. For example, consider the following FCI:

$$I = \left\{a^{-}_{\texttt{pb}}, c^{=}_{\texttt{pb}}, d^{+}_{\texttt{pb}}, A^{-}_{\texttt{sol}}, B^{=}_{\texttt{sol}}, C^{+}_{\texttt{sol}}\right\} \qquad (3)$$

$I$ can be considered as a generalization of a subset of the transactions including the transaction $T$ of equation (2): $I \subseteq T$. The interpretation of this FCI as an adaptation rule is explained below.

**Interpretation.** The interpretation step is supervised by the analyst. The CABAMAKA system provides the analyst with the extracted FCIs and facilities for navigating among them. The analyst may select an FCI, say $I$, and interpret $I$ as an adaptation rule. For example, the FCI in equation (3) may be interpreted in the following terms:

**if** $a$ is a property of `srce` but is not a property of `tgt`,
$c$ is a property of both `srce` and `tgt`,
$d$ is not a property of `srce` but is a property of `tgt`,
$A$ and $B$ are properties of `Sol(srce)`, and
$C$ is not a property of `Sol(srce)`
**then** the properties of `Sol(tgt)` are
$\Phi(\texttt{Sol(tgt)}) = (\Phi(\texttt{Sol(srce)}) \setminus \{A\}) \cup \{C\}$.

This rule has to be translated from the formalism $2^{\mathcal{P}}$ (sets of boolean properties) to the formalism of the adaptation rules of the CBR system. The result is an *adaptation rule*, i.e. a rule whose left part represents conditions on `srce`, `Sol(srce)` and `tgt` and whose right part represents a way to compute `Sol(tgt)`. The role of the analyst is to correct and to validate this adaptation rule and to associate an explanation with it. The analyst is helped in this task by the domain ontology $\mathcal{O}$ that is useful for organizing the FCIs and by the already available adaptation knowledge that is useful for pruning from the FCIs the ones that are already known adaptation knowledge.

## 4.2 Implementation

The CABAMAKA discovery process relies on the steps described in the previous section: $(s_1)$ input the case base, $(s_2)$ select a subset of it (or take the whole case base): first filtering step, $(s_3)$ first formatting substep, $(s_4)$ second filtering step, $(s_5)$ second formatting substep, $(s_6)$ third filtering step, $(s_7)$ data-mining (CHARM), $(s_8)$ last filtering step and $(s_9)$ interpretation. This process is interactive and iterative: the analyst runs each of the $(s_i)$ (and can interrupt it), and can go back to a previous step at each moment.

Among these steps, only the first ones ($(s_1)$ to $(s_3)$) and the last one are dependent on the representation formalism. In the following, the step $(s_3)$ is illustrated in the context of an application. First, some elements on the application itself and the associated knowledge representation formalism are introduced.

**Application domain.** The application domain of the CBR system we are developing is breast cancer treatment: in this application, a problem `pb` describes a class of patients with a set of attributes and associated constraints (holding on the age of the patient, the size and the localization of the tumor, etc.). A solution `Sol(pb)` of `pb` is a set of therapeutic decisions (in surgery, chemotherapy, radiotherapy, etc.).

Two features of this application must be pointed out. First, the source cases are *general cases* (or *ossified cases* according to the terminology of [Riesbeck and Schank, 1989]): a source case corresponds to a class of patients and not to a single one. These source cases are obtained from statistical studies in the cancer domain. Second, the requested behavior of the CBR system is to provide a treatment and explanations on this treatment proposal. This is why the analyst is required to associate an explanation to a discovered adaptation rule.

**Representation of cases and of the domain ontology $\mathcal{O}$.** The problems, the solutions, and the domain ontology of the application are represented in a light extension of OWL DL (the Web Ontology Language recommended by the W3C [Staab and Studer, 2004]). The parts of the underlying description logic that are useful for this paper are presented below (other elements on description logics, DLs, may be found in [Staab and Studer, 2004]).

Let us consider the following example:

$$\begin{aligned}\texttt{srce} \equiv \texttt{Patient} \sqcap \exists \texttt{age}.{\geq_{45}} \sqcap \exists \texttt{age}.{<_{70}} \\ \sqcap\ \exists \texttt{tumor}.(\exists \texttt{size}.{\geq_4} \\ \sqcap\ \exists \texttt{localization}.\texttt{Left-Breast})\end{aligned} \tag{4}$$

`srce` represents the class of patients with an age $a \in [45; 70[$, and a tumor of size $S \geq 4$ centimeters localized in the left breast.

The DL representation entities used here are atomic and defined concepts (e.g. `srce`, `Patient` and $\exists \texttt{age}.{\geq_{45}}$), roles (e.g. `tumor` and `localization`) concrete roles (e.g. `age` and `size`) and constraints (e.g. $\geq_{45}$ and $<_{70}$). A *concept* `C` is an expression representing a class of objects. A *role* `r` is a name representing a binary relation between objects. A *concrete role* `g` is a name representing a function associating a real number to an object (for this simplified presentation, the only concrete domain that is considered is $(\mathbb{R}, \leq)$, the ordered set of real numbers). A constraint `c` represents a subset of $\mathbb{R}$ denoted by $\texttt{c}^{\mathsf{R}}$. For example, intervals such as $\geq_{45}^{\mathsf{R}} = [45; +\infty[$ and $<_{70}^{\mathsf{R}} = ]-\infty; 70[$ introduce constraints that are used in the application.

A concept is either atomic (a concept name) or defined. A defined concept is an expression of the following form: $\texttt{C} \sqcap \texttt{D}$, $\exists \texttt{r}.\texttt{C}$ or $\exists \texttt{g}.\texttt{c}$, where `C` and `D` are concepts, `r` is a role, `g` is a concrete role and `c` is a constraint (many other constructions exist in the DL, but only these three constructions are used here). Following classical DL presentations [Staab and Studer, 2004], an *ontology* $\mathcal{O}$ is a set of axioms, where an axiom is a formula of the form $\texttt{C} \sqsubseteq \texttt{D}$ (general concept inclusion) or of the form $\texttt{C} \equiv \texttt{D}$, where `C` and `D` are two concepts.

The semantics of the DL expressions used hereafter can be read as follows. An interpretation is a pair $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta_{\mathcal{I}}$ is a non empty set (the *interpretation domain*) and $\cdot^{\mathcal{I}}$ is the *interpretation function*, which maps a concept `C` to a set $\texttt{C}^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}}$, a role `r` to a binary relation $\texttt{r}^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$, and a concrete role `g` to a function $\texttt{g}^{\mathcal{I}} : \Delta_{\mathcal{I}} \longrightarrow \mathbb{R}$. In the following, all roles `r` are assumed to be *functional*: $\cdot^{\mathcal{I}}$ maps `r` to a function $\texttt{r}^{\mathcal{I}} : \Delta_{\mathcal{I}} \longrightarrow \Delta_{\mathcal{I}}$. The interpretation of the defined concepts, for an interpretation $\mathcal{I}$, is as follows: $(\texttt{C} \sqcap \texttt{D})^{\mathcal{I}} = \texttt{C}^{\mathcal{I}} \cap \texttt{D}^{\mathcal{I}}$, $(\exists \texttt{r}.\texttt{C})^{\mathcal{I}}$ is the set of objects $x \in \Delta_{\mathcal{I}}$ such that $\texttt{r}^{\mathcal{I}}(x) \in \texttt{C}^{\mathcal{I}}$ and $(\exists \texttt{g}.\texttt{c})^{\mathcal{I}}$ is the set of objects $x \in \Delta_{\mathcal{I}}$ such that $\texttt{g}^{\mathcal{I}}(x) \in \texttt{c}^{\mathsf{R}}$. An interpretation $\mathcal{I}$ is a *model* of an axiom $\texttt{C} \sqsubseteq \texttt{D}$ (resp. $\texttt{C} \equiv \texttt{D}$) if $\texttt{C}^{\mathcal{I}} \subseteq \texttt{D}^{\mathcal{I}}$ (resp. $\texttt{C}^{\mathcal{I}} =$

$\mathtt{D}^{\mathcal{I}}$). $\mathcal{I}$ is a model of an ontology $\mathcal{O}$ if it is a model of each axiom of $\mathcal{O}$. The inference associated with this representation formalism that is used below is the subsumption test: given an ontology $\mathcal{O}$, a concept $\mathtt{C}$ is subsumed by a concept $\mathtt{D}$, denoted by $\models_{\mathcal{O}} \mathtt{C} \sqsubseteq \mathtt{D}$, if for every model $\mathcal{I}$ of $\mathcal{O}$, $\mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$.

More practically, the problems of the CBR application are represented by concepts (as `srce` in (4)). A therapeutic decision `dec` is also represented by a concept. A solution is a finite set $\{\mathtt{dec}_1, \mathtt{dec}_2, \ldots \mathtt{dec}_k\}$ of decisions. The decisions of the system are represented by atomic concepts. The knowledge associated with atomic concepts (and hence, with therapeutic decisions) is given by axioms of the domain ontology $\mathcal{O}$. For example, the decision in surgery $\mathtt{dec} = \mathtt{Partial\text{-}Mastectomy}$ represents a partial ablation of the breast:

$$\begin{aligned} \mathtt{Partial\text{-}Mastectomy} &\sqsubseteq \mathtt{Mastectomy} \\ \mathtt{Mastectomy} &\sqsubseteq \mathtt{Surgery} \\ \mathtt{Surgery} &\sqsubseteq \mathtt{Therapeutic\text{-}Decision} \end{aligned} \quad (5)$$

**Implementation of the first formatting substep** $(s_3)$**.** Both problems and decisions constituting solutions are represented by concepts. Thus, computing $\Phi(\mathtt{srce})$ and $\Phi(\mathtt{Sol}(\mathtt{srce}))$ amounts to the computation of $\Phi(\mathtt{C})$, $\mathtt{C}$ being a concept. A property $p$ is an element of the finite set $\mathcal{P}$ (see section 4.1). In the DL formalism, $p$ is represented by a concept $\mathtt{P}$. A concept $\mathtt{C}$ has the property $p$ if $\models_{\mathcal{O}} \mathtt{C} \sqsubseteq \mathtt{P}$. The set of boolean properties and the set of the corresponding concepts are both denoted by $\mathcal{P}$ in the following. Given $\mathcal{P}$, $\Phi(\mathtt{C})$ is simply defined as the set of properties $\mathtt{P} \in \mathcal{P}$ that $\mathtt{C}$ has:

$$\Phi(\mathtt{C}) = \{\mathtt{P} \in \mathcal{P} \mid \models_{\mathcal{O}} \mathtt{C} \sqsubseteq \mathtt{P}\} \quad (6)$$

As a consequence, if $\mathtt{P} \in \Phi(\mathtt{C})$, $\mathtt{Q} \in \mathcal{P}$ and $\models_{\mathcal{O}} \mathtt{P} \sqsubseteq \mathtt{Q}$ then $\mathtt{Q} \in \Phi(\mathtt{C})$. Thus, the implication (1) is satisfied.

The algorithm of the first formatting substep that has been implemented first computes the $\Phi(\mathtt{C})$'s for $\mathtt{C}$: the source problems and the decisions occurring in their solutions, and then computes $\mathcal{P}$ as the union of the $\Phi(\mathtt{C})$'s. This algorithm relies on the following set of equations[1]:

$$\begin{aligned} \Phi(\mathtt{A}) &= \left\{ \mathtt{B} \;\middle|\; \begin{array}{l} \mathtt{B} \text{ is an atomic concept} \\ \text{occurring in } \mathtt{KB} \text{ and } \models_{\mathcal{O}} \mathtt{A} \sqsubseteq \mathtt{B} \end{array} \right\} \\ \Phi(\mathtt{C} \sqcap \mathtt{D}) &= \Phi(\mathtt{C}) \cup \Phi(\mathtt{D}) \\ \Phi(\exists \mathtt{r}.\mathtt{C}) &= \{\exists \mathtt{r}.\mathtt{P} \mid \mathtt{P} \in \Phi(\mathtt{C})\} \\ \Phi(\exists \mathtt{g}.\mathtt{c}) &= \left\{ \exists \mathtt{g}.\mathtt{d} \;\middle|\; \mathtt{d} \in \mathtt{Cstraints}_{\mathtt{g}} \text{ and } \mathtt{c}^{\mathtt{R}} \subseteq \mathtt{d}^{\mathtt{R}} \right\} \\ \mathtt{Cstraints}_{\mathtt{g}} &= \{\mathtt{c} \mid \text{the expression } \exists \mathtt{g}.\mathtt{c} \text{ occurs in } \mathtt{KB}\} \end{aligned}$$

where $\mathtt{A}$ is an atomic concept, $\mathtt{C}$ and $\mathtt{D}$ are (either atomic or defined) concepts, $\mathtt{r}$ is a role, $\mathtt{g}$ is a concrete role, $\mathtt{c}$ is a constraint and $\mathtt{KB}$, the knowledge base, is the union of the case base and of the domain ontology.

[1] This set of equations itself can be seen as a recursive algorithm, but is not very efficient since it computes several times the same things. The implemented algorithm avoids these recalculations by the use of a cache.

It can be proven that the algorithm for the first formatting substep (computing the $\Phi(\mathtt{C})$'s and the set of properties $\mathcal{P}$) respects (6) under the following hypotheses. First, the constructions used in the DL are the ones that have been introduced above ($\mathtt{C} \sqcap \mathtt{D}$, $\exists \mathtt{r}.\mathtt{C}$ and $\exists \mathtt{g}.\mathtt{c}$, where $\mathtt{r}$ is functional). Second, no defined concept may strictly subsume an atomic concept (for every atomic concept $\mathtt{A}$, there is no defined concept $\mathtt{C}$ such that $\models_{\mathcal{O}} \mathtt{A} \sqsubseteq \mathtt{C}$ and $\not\models_{\mathcal{O}} \mathtt{A} \equiv \mathtt{C}$). Under these hypotheses, (6) can be proven by a recursion on the size of $\mathtt{C}$ (this size is the number of constructions that $\mathtt{C}$ contains). These hypotheses hold for our application. However, an ongoing study aims at finding an algorithm for computing the $\Phi(\mathtt{C})$'s and $\mathcal{P}$ in a more expressive DL, including in particular negation and disjunction of concepts.

For example, let `srce` be the problem introduced by the axiom (4). It is assumed that the constraints associated with the concrete role `age` in `KB` are $<_{30}, \geq_{30}, <_{45}, \geq_{45}, <_{70}$ and $\geq_{70}$, that the constraints associated with the concrete role `size` in `KB` are $<_4$ and $\geq_4$, that there is no concept $\mathtt{A} \neq \mathtt{Patient}$ in `KB` such that $\models_{\mathcal{O}} \mathtt{Patient} \sqsubseteq \mathtt{A}$, and that the only concept $\mathtt{A} \neq \mathtt{Left\text{-}Breast}$ of `KB` such that $\models_{\mathcal{O}} \mathtt{Left\text{-}Breast} \sqsubseteq \mathtt{A}$ is $\mathtt{A} = \mathtt{Breast}$. Then, the implemented algorithm returns:

$$\begin{aligned} \Phi(\mathtt{srce}) = \{&\mathtt{Patient}, \exists \mathtt{age}.\geq_{30}, \exists \mathtt{age}.\geq_{45}, \exists \mathtt{age}.<_{70}, \\ &\exists \mathtt{tumor}.\exists \mathtt{size}.\geq_4, \\ &\exists \mathtt{tumor}.\exists \mathtt{localization}.\mathtt{Left\text{-}Breast} \\ &\exists \mathtt{tumor}.\exists \mathtt{localization}.\mathtt{Breast}\} \end{aligned}$$

And the 7 elements of $\Phi(\mathtt{srce})$ are added to $\mathcal{P}$.

Another example, based on the set of axioms (5) is:

$$\begin{aligned} \Phi(\mathtt{Partial\text{-}Mastectomy}) = \{&\mathtt{Partial\text{-}Mastectomy}, \\ &\mathtt{Mastectomy}, \mathtt{Surgery}, \mathtt{Therapeutic\text{-}Decision}\} \end{aligned}$$

### 4.3 Results

The CABAMAKA process piloted by the analyst produces a set of FCIs. With $n = 647$ cases and $\sigma = 10\%$, CABAMAKA has given 2344 FCIs in about 2 minutes (on a current PC). Only the FCIs with at least a `+` or a `-` in both problem properties and solution properties were kept, which corresponds to 208 FCIs. Each of these FCIs $I$ is presented for interpretation to the analyst under a simplified form by removing some of the items that can be deduced from the ontology. In particular if $\mathtt{P}^{=}_{\mathtt{pb}} \in I$, $\mathtt{Q}^{=}_{\mathtt{pb}} \in I$ and $\models_{\mathcal{O}} \mathtt{P} \sqsubseteq \mathtt{Q}$ then $\mathtt{Q}^{=}_{\mathtt{pb}}$ is removed from $I$. For example, if $\mathtt{P} = (\exists \mathtt{age} \geq_{45}) \in \mathcal{P}$, $\mathtt{Q} = (\exists \mathtt{age} \geq_{30}) \in \mathcal{P}$ and $(\exists \mathtt{age} \geq_{45})^{=}_{\mathtt{pb}} \in I$, then, necessarily, $(\exists \mathtt{age} \geq_{30})^{=}_{\mathtt{pb}} \in I$, which is a redundant piece of information.

The following FCI has been extracted from CABAMAKA:

$$\begin{aligned} I = \{&(\exists \mathtt{age}. <_{70})^{=}_{\mathtt{pb}}, \\ &(\exists \mathtt{tumor}.\exists \mathtt{size}. <_4)^{-}_{\mathtt{pb}}, (\exists \mathtt{tumor}.\exists \mathtt{size}. \geq_4)^{+}_{\mathtt{pb}}, \\ &\mathtt{Curettage}^{=}_{\mathtt{sol}}, \mathtt{Mastectomy}^{=}_{\mathtt{sol}}, \\ &\mathtt{Partial\text{-}Mastectomy}^{-}_{\mathtt{sol}}, \mathtt{Radical\text{-}Mastectomy}^{+}_{\mathtt{sol}}\} \end{aligned}$$

It has been interpreted in the following way: if `srce` and `tgt` both represent classes of patients of less than 70 years old, if the difference between `srce` and `tgt` lies in the tumor size of

the patients—less than $4$ cm for the ones of $\texttt{srce}$ and more than $4$ cm for the ones of $\texttt{tgt}$—and if a partial mastectomy and a curettage of the lymph nodes are proposed for the $\texttt{srce}$, then $\texttt{Sol}(\texttt{tgt})$ is obtained by substituting in $\texttt{Sol}(\texttt{srce})$ the partial mastectomy by a radical one.

It must be noticed that this example has been chosen for its simplicity: other adaptation rules have been extracted that are less easy to understand. More substantial experiments have to be carried out for an effective evaluation.

The choice of considering *every* pairs of distinct source cases can be discussed. Another version of CABAMAKA has been tested that considers only similar source cases, as in [Hanney and Keane, 1996]: only the pairs of source cases such that $|\Phi(\texttt{srce}_1) \cap \Phi(\texttt{srce}_2)| \geq k$ were considered (experimented with $k = 1$ to $k = 10$). The first experiments have not shown yet any improvements in the results, compared to the version without this constraint ($k = +\infty$), and involves the necessity to have the threshold $k$ fixed.

## 5 Conclusion and Future Work

The CABAMAKA system presented in this paper is inspired by the research of Kathleen Hanney and Mark T. Keane [Hanney and Keane, 1996] and by the principles of KDD for the purpose of semi-automatic adaptation knowledge acquisition. It reuses an FCI extraction tool developed in our team and based on a CHARM-like algorithm. Although implemented for a specific application to breast cancer treatment decision support, it has been designed to be reusable for other CBR applications: only a few modules of CABAMAKA are dependent on the formalism of the cases and of the domain ontology, and this formalism, OWL DL, is a well-known standard.

One element of future work consists in searching for ways of simplifying the presentation of the numerous extracted FCIs to the analyst. This involves an organization of these FCIs for the purpose of navigation among them. Such an organization can be a hierarchy of FCIs according to their specificities or a clustering of the FCIs in themes.

A second piece of future work, still for the purpose of helping the analyst, is to study the algebraic structure of all the possible adaptation rules associated with the operation of composition: $r$ is a composition of $r_1$ and $r_2$ if adapting $(\texttt{srce}, \texttt{Sol}(\texttt{srce}))$ to solve $\texttt{tgt}$ thanks to $r$ gives the same solution $\texttt{Sol}(\texttt{tgt})$ as (1) solving a problem $\texttt{pb}$ by adaptation of $(\texttt{srce}, \texttt{Sol}(\texttt{srce}))$ thanks to $r_1$ and (2) solving $\texttt{tgt}$ by adaptation of $(\texttt{pb}, \texttt{Sol}(\texttt{pb}))$ thanks to $r_2$. The idea is to find a smallest family of adaptation rules, $F$, such that the closure of $F$ under composition contains the sets of the extracted adaptation rules expressed in the form of FCIs. It is hoped that $F$ is much smaller than $S$ and so requires less effort from the analyst while corresponding to the same adaptation knowledge.

Another study on AKA for our CBR system was AKA from experts (based on the analysis of the adaptations performed by the experts). This AKA has led to a few adaptation rules and also to *adaptation patterns*, i.e. general strategies for case-based decision support that are associated with explanations but that need to be instantiated to become operational. A third future work is *mixed* AKA, that is a combined use of the adaptation patterns and of the adaptation rules extracted from CABAMAKA: the idea is to try to instantiate the former by the latter in order to obtain a set of human-understandable and operational adaptation rules.

## References


[Aamodt, 1990] A. Aamodt. Knowledge-Intensive Case-Based Reasoning and Sustained Learning. In L. C. Aiello, editor, *Proc. of the 9th European Conference on Artificial Intelligence (ECAI'90)*, August 1990.

[Carbonell, 1983] J. G. Carbonell. Learning by analogy: Formulating and generalizing plans from past experience. In R. S. Michalski and J. G. Carbonell and T. M. Mitchell, editor, *Machine Learning, An Artificial Intelligence Approach*, chapter 5, pages 137–161. Morgan Kaufmann, Inc., 1983.

[Hand *et al.*, 2001] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, Cambridge (MA), 2001.

[Hanney and Keane, 1996] K. Hanney and M. T. Keane. Learning Adaptation Rules From a Case-Base. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning – Third European Workshop, EWCBR'96*, LNAI 1168, pages 179–192. Springer Verlag, Berlin, 1996.

[Hanney, 1997] K. Hanney. Learning Adaptation Rules from Cases. Master's thesis, Trinity College, Dublin, 1997.

[Jarmulak *et al.*, 2001] J. Jarmulak, S. Craw, and R. Rowe. Using Case-Base Data to Learn Adaptation Knowledge for Design. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 1011–1016. Morgan Kaufmann, Inc., 2001.

[Leake *et al.*, 1996] D. B. Leake, A. Kinley, and D. C. Wilson. Acquiring Case Adaptation Knowledge: A Hybrid Approach. In *AAAI/IAAI*, volume 1, pages 684–689, 1996.

[Riesbeck and Schank, 1989] C. K. Riesbeck and R. C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989.

[Shiu *et al.*, 2001] S. C. K. Shiu, Daniel S. Yeung, C. Hung Sun, and X. Wang. Transferring Case Knowledge to Adaptation Knowledge: An Approach for Case-Base Maintenance. *Computational Intelligence*, 17(2):295–314, 2001.

[Smyth and Keane, 1996] B. Smyth and M. T. Keane. Using adaptation knowledge to retrieve and adapt design cases. *Knowledge-Based Systems*, 9(2):127–135, 1996.

[Staab and Studer, 2004] S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, Berlin, 2004.

[Wiratunga *et al.*, 2002] N. Wiratunga, S. Craw, and R. Rowe. Learning to Adapt for Case-Based Design. In S. Craw and A. Preece, editors, *Proceedings of the 6th European Conference on in Case-Based Reasoning (ECCBR-02)*, LNAI 2416, pages 421–435, 2002.

[Zaki and Hsiao, 2002] M. J. Zaki and C.-J. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining. In *SIAM International Conference on Data Mining SDM'02*, pages 33–43, Apr 2002.