# Adapting Spatial and Temporal Cases

Valmi Dufour-Lussier, Florence Le Ber, Jean Lieber, Laura Martin

HAL Id: hal-00735231
https://inria.hal.science/hal-00735231

Submitted on 25 Sep 2012

# Adapting Spatial and Temporal Cases

Valmi Dufour-Lussier[1,2,3], Florence Le Ber[4,1,2],
Jean Lieber[1,2,3], and Laura Martin[5]

[1] Université de Lorraine, LORIA, UMR 7503 — 54506 Vandœuvre-lès-Nancy, France
[2] CNRS — 54506 Vandœuvre-lès-Nancy, France
[3] Inria — 54602 Villers-lès-Nancy, France
[4] ENGEES, LHYGES, UMR 7517 — 67000 Strasbourg, France
[5] INRA, UR055 ASTER–Mirecourt — 88500 Mirecourt, France
*first name.surname*@loria.fr, laura.martin@mirecourt.inra.fr

**Abstract.** Qualitative algebras form a family of languages mainly used
to represent knowledge depending on space or time. This paper proposes
an approach to adapt cases represented in such an algebra. A spatial
example in agronomy and a temporal example in cooking are given. The
idea behind this adaptation approach is to apply a substitution and then
repair potential inconsistencies, thanks to belief revision on qualitative
algebras.

**Keywords:** adaptation, case-based reasoning, qualitative algebra, spatial reasoning, temporal reasoning

## 1 Introduction

Qualitative spatial and temporal reasoning (QSTR) as a research domain has
been active since the beginning of the 1980s. The paradigm has been exploited to
help solve planning and constraint satisfaction problems, but rarely within case-
based reasoning (CBR). Nevertheless, many domains in which QSTR is used
could be addressed with CBR because the knowledge involved is usually con-
textual and incompletely formalised. This is the case in the domain of landscape
agronomy, in which knowledge is acquired from farm surveys and from schematic
descriptions of the spatial organisation of farming territories. Another example
is the cooking domain, in which some knowledge is of a temporal nature.

This paper proposes an approach for the adaptation of spatial and temporal
cases, which is based on a process integrating substitution and revision-based
adaptation. One may, for instance, want to replace a plot of maize with a plot of
a different crop in a farm, or to replace mushrooms with carrots in a recipe. Mere
substitution is insufficient, because not all crops, nor ingredients, are used in the
same way. It is therefore supplemented with a belief revision process through
which the spatial representation of the farm or the temporal representation of
the recipe are changed to be consistent with domain knowledge.

Section 2 justifies the paper's approach by introducing two examples from
the farming and cooking domains. Section 3 then introduces the formal notions

required for the approach, namely in terms of CBR, revision-based adaptation, and QSTR. The approach is then defined in details in section 4, and an algorithm is described in section 5. Section 6 illustrates those formal notions and the results of the algorithm using the two examples introduced in section 2. Related work is discussed in section 7. Section 8 concludes and presents some future work.

## 2   Introduction of the running examples

This section describes two examples in which a mere substitution would not yield an acceptable adaptation of a retrieved case with respect to the target problem. In the first example, the spatial adaptation of a farm must take into account agronomic knowledge about the location of fields with respect to the environment and cropping constraints. In the second example, the temporal adaptation of a risotto recipe must take into account cooking knowledge about the length required to cook different vegetables.

### 2.1   A spatial, agronomic example

Spatial adaptation is illustrated using the example of Miscanthus allocation practices in agriculture. Miscanthus is a perennial grass currently promoted as a renewable source of energy in Europe to produce high yield of biomass with low input [1]. Its potential to reduce greenhouse gas emission is dependent to its spatial allocation into farmlands [2], therefore modelling spatial land use changes into farmlands is of great interest.

CBR can be used to model Miscanthus spatial allocation. The problem is the a crop production requirements and the farm description, and the solution is a crops spatial allocations. A farm description is defined by a cropping plan (the crop proportions/allocations into farmland) and by the spatial farmland features (e.g. the spatial relations of plots with buildings, woodland and rivers).

In this example, illustrated by figure 1, a farmer who wants to cultivate Miscanthus is considered. A case corresponding to a maize farm could be retrieved (figure 1a), based on real cases surveyed in France [3] and expert knowledge which identify similarities in Miscanthus and maize allocation requirements regarding temperature and soil moisture [4].

Replacing maize with Miscanthus (which is usually harvested from February to March in France) comes with a spatial constraint in the agronomic domain knowledge. Because access to plots by harvesting machinery is impaired by excess soil water in winter, Miscanthus must not be allocated near a river, in a flood-risk area, whereas maize can be planted up to a legal 5 metres from rivers.

Therefore, it is expected that the adaptation process would not only replace maize by Miscanthus, but also reduce the size of the plot so that it doesn't overlap with the flood plain of any nearby river (figure 1b). A better adaptation may break the plot into two parts and retain maize cultivation in the flood plain to ensure that this part of the farmland remains productive (figure 1c).
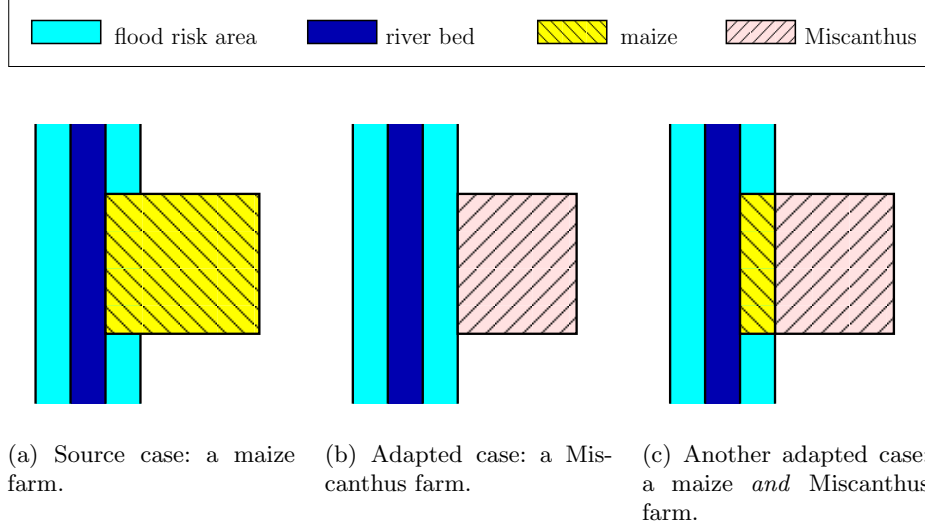
(a) Source case: a maize farm.

(b) Adapted case: a Miscanthus farm.

(c) Another adapted case: a maize *and* Miscanthus farm.

Fig. 1: Crop spatial allocation example.

## 2.2 A temporal, cooking example

Temporal adaptation is illustrated through recipe adaptation, in the spirit of TAAABLE [5], a CBR application for cooking. If a user asks for a carrot risotto recipe and none is found in the case base, TAAABLE may retrieve a mushroom risotto recipe and suggest reusing it with mushrooms replaced with carrots.

Say the mushrooms are added to the rice 2 minutes before the end, and the cooking domain knowledge indicates that carrots must be cooked for 25 minutes in order to be done, whereas the rice must be cooked for 18 minutes. A proper adaptation would require not only the lengthening of the cooking time of the vegetables, but also a reordering of the actions in the recipe.

## 3 Background

### 3.1 Case adaptation

In this paper, `Source`, `Target` and DK respectively denote the case to be adapted, the target case and the domain knowledge. `Source` and `Target` are required to be consistent with DK.[6] Given `Source` and `Target`, the adaptation aims at building

---

[6] If a case `Source` from the case base is inconsistent with the domain knowledge, a consistency maintenance process should be triggered to restore this consistency. If a query to the CBR system is inconsistent with the domain knowledge, it either should be rejected (e.g. "I want a recipe with apples but without fruit" is rejected by TAAABLE since DK entails that apples are fruits), or DK should be modified.

a new case, `AdaptedCase`. This case is built by adding some information to the target case (intuitively, `Target` specifies only the "problem part" of the query), and it has to be consistent with `DK`.

It is assumed that a matching step precedes the adaptation process, providing links between `Source` and `Target`. It is represented by a substitution $\sigma$, mapping descriptors of `Source` to descriptors of `Target`. As an example, in the system TAAABLE, matching is performed during retrieval [5]. This process, applied to the cooking example of the previous section, would have returned $\sigma = $ `mushroom` $\rightsquigarrow$ `carrot`. In the following, this preprocessing step of adaptation is considered to be given and, thus, $\sigma$ is an input of the adaptation process described in section 4.

## 3.2 Belief revision and revision-based adaptation

**Belief revision.** In a given representation formalism, a revision operator $\dotplus$ maps two knowledge bases $\psi$ and $\mu$ to knowledge base $\psi \dotplus \mu$, the revision of $\psi$ by $\mu$. Intuitively, $\psi \dotplus \mu$ is obtained by making a minimal change of $\psi$ into $\psi'$, so that the conjunction of $\psi'$ and $\mu$, $\psi' \wedge \mu$, is consistent. Then, $\psi \dotplus \mu$ is this conjunction. The notion of minimal change can be modelled in various ways, so there are various revision operators. However, postulates have been proposed for such an operator, such as the AGM postulates [6]. These postulates have been applied to propositional logic [7] and well studied in this formalism. Given a distance `dist` on the set $\mathcal{U}$ of the interpretations, an operator $\dotplus_{\texttt{dist}}$ can be uniquely defined (up to logical equivalence) as: the set of models of $\psi \dotplus_{\texttt{dist}} \mu$ is the set of models of $\mu$ that have a minimal distance to the set of models of $\psi$.

**Revision-based adaptation.** Given a revision operator $\dotplus$, $\dotplus$-adaptation consists simply in using this revision operator to perform adaptation, taking into account the domain knowledge:

$$\texttt{AdaptedCase} = (\texttt{DK} \wedge \texttt{Source}) \dotplus (\texttt{DK} \wedge \texttt{Target}) \tag{1}$$

The intuition behind revision-based adaptation is to reduce adaptation to an inconsistency repair. It has been studied in propositional logic [8] and, more generally, in metric spaces [9]. It has been generalised to multiple case adaptation using integrity constraint belief merging in [10]. Its principle has also been applied to adaptation in an expressive description logic [11]. This paper applies it, with some differences, to qualitative constraint networks.

## 3.3 Qualitative representation of spatial and temporal knowledge

**Definitions.** A qualitative algebra is a relation algebra that defines a set $\mathfrak{B}$ of binary relations applicable between two variables, usually representing points, intervals or regions. Allen interval algebra [12], for instance, introduces 13 basic relations between intervals, corresponding to the 13 possible arrangements of their lower and upper bounds. 7 relations are illustrated in figure 2a. The 6 others are the inverse of the first 6 (eq is symmetric).

$\mathcal{INDU}$ [13] extends the set of Allen relations by combining them with relations over the interval durations. For 7 Allen relations, there is only one possible duration relation (e.g. $i$ {d} $j$ implies that the duration of $i$ is shorter than the duration of $j$). For the other 6, all three duration relations $<$, $=$ and $>$ are possible. This yields a total of 25 basic relations. They are written as $r^s$, where $r$ is an Allen relation and $s$ is a duration relation.

Region connection calculi [14] are well-known spatial algebras. The most usual, RCC8, introduces 8 relations between regions, as shown in figure 2b.
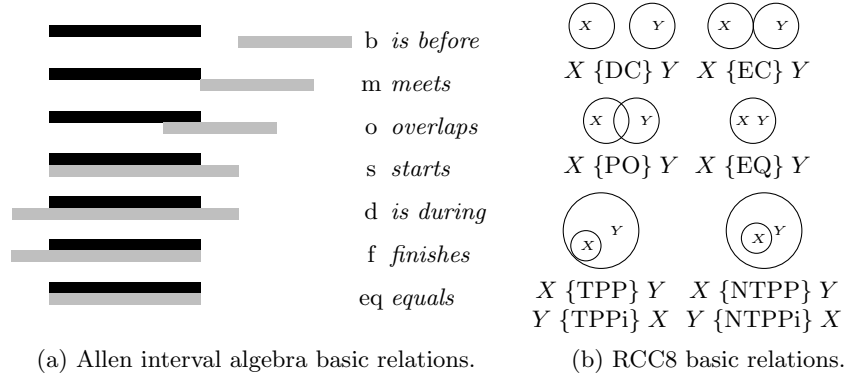


(a) Allen interval algebra basic relations.   (b) RCC8 basic relations.

Fig. 2: Two common qualitative algebras.

Qualitative knowledge can be represented as qualitative constraint networks (QCNs). A QCN is a pair $(V, C)$, where $V$ is a set of variables, and $C$ is a set of constraints of the form $V_i \ C_{ij} \ V_j$ with $V_i, V_j \in V$, and $C_{ij}$ a set of the basic relations defined by the algebra ($C_{ij}$ is a relation that is a disjunction of the basic relations, i.e. $i \ \{r_1, r_2\} \ j$ means that $i$ is related to $j$ with either $r_1$ or $r_2$). In $\mathcal{INDU}$, shortcut notations $r^?$ and $?^s$ respectively represent the Cartesian product of $r$ and all possible duration relations and the product of $s$ and all possible Allen relations (e.g., $\{m\}^? = \{m^<, m^=, m^>\}$; $\{d\}^? = \{d^<\}$; $\{?\}^= = \{b^=, m^=, o^=, eq^=, oi^=, mi^=, bi^=\}$.

A *scenario* is a QCN $\mathcal{S} = (V_\mathcal{S}, C_\mathcal{S})$ such that for each $V_i, V_j \in V_\mathcal{S}$, there exists one constraint $V_i \ \{r\} \ V_j \in C_\mathcal{S}$. $\mathcal{S}$ satisfies the QCN $\mathcal{N} = (V_\mathcal{N}, C_\mathcal{N})$ if $\mathcal{S}$ and $\mathcal{N}$ have the same set of variables and each constraint relation in $\mathcal{S}$ is a subset of the corresponding constraint relation in $\mathcal{N}$. A scenario is consistent if a valuation can be provided for the variables such that all constraints are observed, and a QCN is consistent if it has a consistent scenario. Two QCNs are said to be equivalent if every scenario of the former is a scenario of the latter and vice-versa.

**Revision of QCNs.** A QCN is a knowledge base and thus, the issue of revising a QCN $\psi$ by a QCN $\mu$ can be addressed. In [16], such a revision operator is

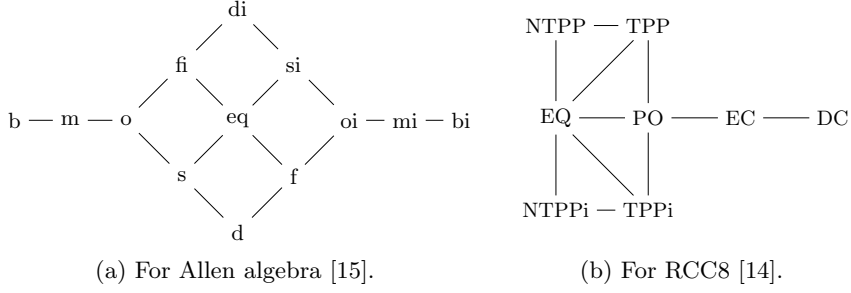(a) For Allen algebra [15].　　　　　(b) For RCC8 [14].

Fig. 3: The relation neighbourhood graphs of two common qualitative algebras.

defined,[7] following the idea of an operator $\dot{+}_{\texttt{dist}}$ (cf. section 3.2), where an interpretation is a scenario, a model of a QCN is a scenario that satisfies it, and a distance $\texttt{dist}$ between scenarios/interpretations is defined as follows.

First, a distance $d$ between basic relations of the considered algebra is defined. Formally, a *neighbourhood graph* whose vertices are the relations of the algebra is given, and $d(r, s)$ is the distance between $r$ and $s$ in the graph. It represents closeness between relations. For instance, b and m are close ($d(\text{b}, \text{m}) = 1$) since they express similar conditions on the boundaries of the intervals (for the lower bounds: = for both; for the upper bounds: < for b and = for m). Figure 3 presents such graphs, respectively for Allen algebra and RCC8. $d$ makes it possible to define $\texttt{dist}$, a distance between two scenarios $\mathcal{S} = (V, C_\mathcal{S})$ and $\mathcal{T} = (V, C_\mathcal{T})$ based on the same set of variables $V$, as:

$$\texttt{dist}(\mathcal{S}, \mathcal{T}) = \sum_{V_i, V_j \in V, i \neq j} d(r_\mathcal{S}(V_i, V_j), r_\mathcal{T}(V_i, V_j)) \qquad (2)$$

where $r_\mathcal{S}(V_i, V_j)$ is the relation $r$ such that $V_i \{r\} V_j \in C_\mathcal{S}$.

Given two QCNs $\psi$ and $\mu$, the revision of $\psi$ by $\mu$ returns the set $R$ of scenarios satisfying $\mu$ that are the closest ones to the set of scenarios satisfying $\psi$.[8]

## 4　Formalisation

### 4.1　Representation of the adaptation problem

**Parametrised QCNs.** It is assumed that the variables of the considered QCNs can be parametrised by elements of a given set $\mathcal{P}$. A parameter $p \in \mathcal{P}$ is either a *concrete parameter*, $p \in \mathcal{CP}$, or an *abstract parameter*, $p \in \mathcal{AP}$: $\mathcal{P} = \mathcal{CP} \cup \mathcal{AP}$,

---

[7] Technically, the authors of [16] define a merge operator taking a coercive QCN as a parameter. A revision operator can be defined using this merge operator.

[8] This slightly differs from the definition of revision given in section 3.2 where $\psi \dot{+} \mu$ is a knowledge base, not a set of models. This is due to the fact that there may be no QCN that has $R$ as set of scenarios satisfying it.

$\mathcal{CP} \cap \mathcal{AP} = \emptyset$. A concrete parameter denotes a concept of the application domain, e.g. $\texttt{mushroom} \in \mathcal{CP}$ for the cooking example. In this example, the formal interval $\texttt{cooking}(\texttt{mushroom})$ represents the temporal interval of the mushroom cooking. The domain knowledge $\texttt{DK} = (V_{\texttt{DK}}, C_{\texttt{DK}})$ is a set of constraints, for example:

$$C_{\texttt{DK}} = \begin{cases} \texttt{cooking}(\texttt{rice}) \ ?^{=} \ \texttt{18\_min} & \textit{(rice requires 18 min of cooking)} \\ \texttt{cooking}(x) \ \{\texttt{m}\}^{?} \ \texttt{cooked}(x) & \textit{(when the action of cooking x is} \\ & \textit{finished, x is cooked)} \\ \texttt{18\_min} \ ?^{<} \ \texttt{25\_min} & \textit{(18 min is shorter than 25 min)} \end{cases} \quad (3)$$

where $\texttt{rice} \in \mathcal{CP}$ and $x \in \mathcal{AP}$. An abstract parameter must be understood with a universal quantification over the concrete parameters; e.g. $\texttt{cooking}(x) \ \{\texttt{m}\}^{?}$ $\texttt{cooked}(x)$ entails $\texttt{cooking}(\texttt{mushroom}) \ \{\texttt{m}\}^{?} \ \texttt{cooked}(\texttt{mushroom})$.

Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two QCNs. $\mathcal{N}_1 \wedge \mathcal{N}_2$ is the QCN $\mathcal{N} = (V, C)$ such that $V = V_1 \cup V_2$ and $C$ contains the constraints of $C_1$, the constraints of $C_2$, and the constraints that are deduced by instantiation of the abstract parameters by concrete parameters appearing in $\mathcal{N}_1$ and $\mathcal{N}_2$ (technically, this instanciation of $x \in \mathcal{AP}$ to $p \in \mathcal{CP}$ is a unification [17]). For example, if $\mathcal{N}_1 = C_{\texttt{DK}}$ defined by equation (3) and $\mathcal{N}_2 = (\{\texttt{cooking}(\texttt{tomato}), \texttt{cooked}(\texttt{tomato})\}, \emptyset)$, then $\mathcal{N}_1 \wedge \mathcal{N}_2 = (V, C)$ with $C = C_{\texttt{DK}} \cup \{\texttt{cooking}(\texttt{tomato}) \ \{\texttt{m}\}^{?} \ \texttt{cooked}(\texttt{tomato})\}$.

**Substitutions.** The *atomic substitution* $\sigma = p \rightsquigarrow q$, where $p, q \in \mathcal{P}$, is the function from $\mathcal{P}$ to $\mathcal{P}$ defined by $\sigma(a) = \begin{cases} q & \text{if } a = p \\ a & \text{otherwise} \end{cases}$. A *substitution* is a composition $\sigma_1 ; \ldots ; \sigma_n$ of atomic substitutions $\sigma_i$.[9]

Let $\sigma = p \rightsquigarrow q$ be an atomic substitution. $\sigma$ is *concrete* if $p, q \in \mathcal{CP}$. $\sigma$ is an *atomic abstraction* if $p \in \mathcal{CP}$ and $q \in \mathcal{AP}$. $\sigma$ is an *atomic refinement* if $p \in \mathcal{AP}$ and $q \in \mathcal{CP}$. A *concrete substitution* (resp., an abstraction, a refinement) is a composition of concrete atomic substitutions (resp., of atomic abstractions, of atomic refinements). Any concrete substitution $\sigma$ can be written $\sigma = \alpha ; \varrho$ where $\alpha$ is an abstraction and $\varrho$ is a refinement, as the following equation illustrates:

$$\texttt{mushroom} \rightsquigarrow \texttt{carrot} = \texttt{mushroom} \rightsquigarrow x ; x \rightsquigarrow \texttt{carrot}$$

where $\texttt{mushroom}, \texttt{carrot} \in \mathcal{CP}$ and $x \in \mathcal{AP}$. This can be shown as follows. First, $\sigma$ can be written $p_1 \rightsquigarrow q_1 ; \ldots ; p_n \rightsquigarrow q_n$ with $p_i, q_i \in \mathcal{CP}$ and $p_i \neq p_j$ if $i \neq j$.[10] Let $x_1, \ldots, x_n$ be $n$ abstract parameters, let $\alpha_i = p_i \rightsquigarrow x_i$, let $\varrho_i = x_i \rightsquigarrow q_i$, let $\alpha = \alpha_1 ; \ldots ; \alpha_n$, and let $\varrho = \varrho_1 ; \ldots ; \varrho_n$. $\alpha$ is an abstraction, $\varrho$ is a refinement and $\sigma = \alpha ; \varrho$.

Let $\sigma$ be a substitution. $\sigma$ is extended on qualitative variables by applying it to their parameters. For example, if $\sigma = \texttt{mushroom} \rightsquigarrow \texttt{carrot}$ then

---

[9] The composition of $\sigma$ and $\sigma'$, denoted by $\sigma ; \sigma'$, is the function that associates to $p \in \mathcal{P}$, $\sigma ; \sigma' \ (p) = \sigma'(\sigma(p)) \in \mathcal{P}$.

[10] Which can be shown inductively thanks to the following lemmas: (1) if $p \neq q$ then $p \rightsquigarrow q ; p \rightsquigarrow q' = p \rightsquigarrow q$, (2) $p \rightsquigarrow q ; q \rightsquigarrow r = p \rightsquigarrow r$, and (3) if $p \neq p'$, $q \neq p'$ and $q' \neq p$, then $p \rightsquigarrow q ; p' \rightsquigarrow q' = p' \rightsquigarrow q' ; p \rightsquigarrow q$.

$\sigma(\texttt{cooking}(\texttt{mushroom})) = \texttt{cooking}(\texttt{carrot})$. Then, $\sigma$ is extended to a constraint $c = (V_i \ C_{ij} \ V_j)$ by $\sigma(c) = (\sigma(V_i) \ C_{ij} \ \sigma(V_j))$. Finally, $\sigma$ is extended on a QCN by applying it to its variables and constraints: $\sigma((V, C)) = (\sigma(V), \sigma(C))$ where $\sigma(V) = \{\sigma(V_i) \mid V_i \in V\}$ and $\sigma(C) = \{\sigma(c) \mid c \in V\}$.

**Adaptation problem.** An adaptation problem is given by a tuple $(\texttt{Source}, \texttt{Target}, \texttt{DK}, \sigma)$. $\texttt{Source}$ and $\texttt{Target}$ are the representations of the source and target cases by QCNs with concrete variables (i.e. not parametrised by any abstract parameter). $\texttt{DK}$ is a QCN representing the domain knowledge. $\sigma = p_1 \rightsquigarrow q_1 \ ; \ \ldots \ ; \ p_n \rightsquigarrow q_n$ is a concrete substitution such that each $p_i$ (resp., $q_i$) parametrises a variable of $\texttt{Source}$ (resp., $\texttt{Target}$). $\texttt{DK} \wedge \texttt{Source}$ and $\texttt{DK} \wedge \texttt{Target}$ are assumed to be consistent (cf. section 3.1). The goal of adaptation is to build a consistent QCN $\texttt{AdaptedCase}$ that entails $\texttt{DK} \wedge \texttt{Target}$, whose qualitative variables are obtained by applying $\sigma$ on the qualitative variables of $\texttt{Source}$, and that is obtained thanks to minimal modification of $\texttt{DK} \wedge \texttt{Source}$.

### 4.2 Principles of revision-based adaptation of a QCN

A first idea to perform the adaptation, given a tuple $(\texttt{Source}, \texttt{Target}, \texttt{DK}, \sigma)$, is to apply $\sigma$ on $\texttt{Source}$, thus obtaining a QCN $\texttt{DK} \wedge \sigma(\texttt{Source})$ that may be inconsistent, and then restoring consistency. Although this gives a good intuition of the revision-based adaptation of a QCN, it is not consistent with the irrelevance of syntax principle.[11] Indeed, any two inconsistent knowledge bases (e.g. two inconsistent QCNs) are equivalent: their sets of models are both empty. Thus, at a semantic level, repairing an inconsistent knowledge base is meaningless. By contrast, revision aims at modifying a *consistent* knowledge base with another *consistent* one, the conjunction of which may be inconsistent. This is one reason why we do not follow straightforwardly this first idea. Another reason is more practical: using revision of QCNs makes it possible to exploit the work of [16].

The revision-based adaptation consists first in decomposing $\sigma$ in an abstraction $\alpha$ and a refinement $\varrho$: $\sigma = \alpha \ ; \ \varrho$ (cf. previous section). Then, $\alpha$ is applied to $\texttt{Source}$: a QCN $\texttt{DK} \wedge \alpha(\texttt{Source})$ is built that is necessarily consistent since $\texttt{DK} \wedge \texttt{Source}$ is consistent and every constraint of $\texttt{DK} \wedge \alpha(\texttt{Source})$ corresponds to a constraint of $\texttt{DK} \wedge \texttt{Source}$. In other words, $\texttt{DK} \wedge \texttt{Source}$ is consistent and is more or equally constrained as $\texttt{DK} \wedge \alpha(\texttt{Source})$, so $\texttt{DK} \wedge \alpha(\texttt{Source})$ is consistent.

The third step involves revision. The idea is to make a revision of $\psi$ by $\mu$ where $\psi = \texttt{DK} \wedge \alpha(\texttt{Source})$ and $\mu = \texttt{DK} \wedge \texttt{Target} \wedge \mathcal{N}_\varrho$ where $\mathcal{N}_\varrho$ represents the following statement: "Each qualitative variable $V_i$ of $\alpha(\texttt{Source})$ is constrained to be equal to its refinement $\varrho(V_i)$." For this purpose, the relation *eq* for equality is used:[12] $V_i \ eq \ \varrho(V_i)$. Therefore, $\mathcal{N}_\varrho = (V_\varrho, C_\varrho)$ where

$$V_\varrho = \alpha(V) \cup \sigma(V) \qquad C_\varrho = \{V_i \ eq \ \varrho(V_i) \mid V_i \in \alpha(V)\}$$

---

[11] This states that an inference remains valid when replacing formulas by logically equivalent formulas. This principle is usually observed in belief revision [7], i.e. if $\psi \equiv \psi'$ and $\mu \equiv \mu'$, then $\psi \dotplus \mu \equiv \psi' \dotplus \mu'$.

[12] *eq* is eq$^=$ for $\mathcal{INDU}$ and EQ for RCC8.

$\mu$ is consistent since DK $\wedge$ Target is and since each constraint $V_i$ *eq* $\varrho(V_i)$ of $\mathcal{N}_\varrho$ either is a tautology (when $V_i$ does not contain any abstract parameter refined by $\varrho$) or links a variable $V_i$ that does not appear in DK $\wedge$ Target with $\varrho(V_i)$.

Then, $\psi \dotplus \mu$ gives a set of scenarios and AdaptedCase is chosen among them.

## 5  Algorithm and implementation

**Input and Output.** The revision algorithm takes as input $\psi =$ DK$\wedge\alpha($Source$)$, $\mu =$ DK$\wedge$Target$\wedge C_\varrho$, as well as a relation neighbourhood graph and a transitivity table for the algebra used. The neighbourhood graph enables to define a distance $d$ between relations and the transitivity table defines a relation composition function $\circ : \mathfrak{B} \times \mathfrak{B} \rightarrow 2^{\mathfrak{B}}$, for example, m $\circ$ mi = {eq, f, fi} in Allen algebra. The revision algorithm returns a set of scenarios of $\mu$ and their distance to $\psi$.

**Algorithm.** First, it is necessary to ensure that all variables in either QCN are present in the other QCN as well. All pairs of variables that have no relation associated to them are given the relation $\mathfrak{B}$–the unspecified relation.

The algorithm must then generate all the scenarios of $\mu$ and of $\psi$ and measure their distance pair-wise. The amount of scenarios for a given QCN is of the order of $O\left(|\mathfrak{B}|^{\frac{|V|\cdot(|V|-1)}{2}}\right)$. For each pair of scenarios, the distance is calculated using equation (2). The distance between a scenario $\mathcal{S}$ of $\mu$ and $\psi$ is the smallest distance between $\mathcal{S}$ and any scenario of $\psi$. The distance between the QCNs $\mu$ and $\psi$ is the smallest distance between a scenario of $\mu$ and the QCN $\psi$. Once all scenarios of $\mu$ and of $\psi$ have been compared pair-wise, the distance between $\mu$ and $\psi$ is known and the scenarios of $\mu$ equal to this distance are returned.

Only consistent scenarios are to be considered. Path-consistency of a scenario[13] is a necessary and sufficient condition for consistency.[14] This condition is verified in time $O(|V|^3)$ if, for each $V_i, V_j, V_k \in V$, $r(V_i, V_k) \in r(V_i, V_j) \circ r(V_j, V_k)$. All inconsistent scenarios are discarded.

**Optimisations.** Because of the complexity, limiting the search space is essential. Considering that the minimum of sums is never less than the sum of minimums, a lower bound on the distance between two QCNs can be obtained in time $O(|V|^2 \cdot |\mathfrak{B}|^2)$ by computing the pair-wise minimal distance for each constraint and summing those. Empirically, it appears that the actual distance is usually closer to this lower bound than to the maximal bound, which is a function of the length of the longest path in the neighbourhood graph.

Therefore, it is often profitable to set an initial upper bound on the distance between $\psi$ and $\mu$ which is equal to the lower bound, and search incrementally.

---

[13] A scenario is path-consistent if, in each 3-tuples $(V_i, V_j, V_k)$ of variables, for each consistent valuation of $V_i$ and $V_j$, there exists a consistent valuation of $V_k$. In RCQs, this is checked with transivity tables indicating the possible results of the composition of any two relations.

[14] The same is not true for QCNs in general, as shown in [12].

This search can be further optimised by computing the lower bound on the distance between a scenario $\mathcal{S}$ of $\mu$ and $\psi$, which can be done in time $O(|V|^2)$. This bound makes it possible to discard altogether certain scenarios of $\mu$ and thus avoid having to generate all the scenarios of $\psi$ over again. Of course, whenever a scenario of $\mu$ is found to be at an acceptable distance to a scenario of $\psi$, incremental search means it is not necessary to examine other scenarios of $\psi$.

Another worthwhile optimisation in problems larger than a few variables is computing the algebraic closure of the QCNs, which is obtained by enforcing path-consistency. That is, for each $V_i, V_j, V_k \in V$, $C_{ik}$ is replaced with $C'_{ik} = C_{ik} \cap (C_{ij} \circ C_{jk})$ where $\circ$ is extended on $2^{\mathfrak{B}} \times 2^{\mathfrak{B}} \to 2^{\mathfrak{B}}$ by $R \circ S = \bigcup_{r \in R, s \in S} r \circ s$. This is repeated until stability, i.e. no relation is changed after considering all 3-tuples of variables.

The optimisations proposed herein maintain the completeness and correctness of the algorithm, but they may prove insufficient to obtain a usable system. We think that approximation algorithms may give satisfactory results while running significantly faster. This will be the subject of future work.

## 6 Application on the running examples

This section revisits the examples from section 2. First, the agronomic example is taken in its simple form (where only Miscanthus is cultivated) to illustrate the algorithm. Then, the formalisation and the results are shown and discussed for the more complex form of the agronomic example and for the cooking example.

### 6.1 Simple agronomic problem

Consider a farm with one maize plot being adjacent to a river. To address the fact that there is a difference in possible agricultural uses between the bed of the river and the zone with flood risks, it is broken in two regions, `low_water_channel` and `flood_plain`, such that the former is a proper part of the latter and that their boundaries don't touch. This is expressed in RCC8 as `low_water_channel` {NTPP} `flood_plain`. The fact that a maize plot is adjacent to a river is represented as `plot(maize)` {EC} `low_water_channel`.

A farmer wishes to cultivate Miscanthus in a similar setting, prompting the retrieval of the farm case just described. A substitution must be applied: $\sigma =$ `maize` $\rightsquigarrow$ `Miscanthus` = `maize` $\rightsquigarrow x$ ; $x \rightsquigarrow$ `Miscanthus`. An important knowledge about Miscanthus is that it must not be cultivated in a zone susceptible to flooding, which can be expressed as `plot(Miscanthus)` {DC, EC} `flood_plain`.

In this example, $\psi$ contains the constraints

$$C_{\text{DK}} = \{\texttt{plot(Miscanthus)} \ \{\text{DC}, \text{EC}\} \ \texttt{flood\_plain}\}$$
$$C_{\alpha(\texttt{Source})} = \left\{ \begin{array}{l} \texttt{low\_water\_channel}\{\text{NTPP}\} \ \texttt{flood\_plain} \\ \texttt{plot}(x) \ \{\text{EC}\} \ \texttt{low\_water\_channel} \end{array} \right\}$$

and $\mu$ contains the constraints

$$C_{\text{DK}} = \{\texttt{plot(Miscanthus)}\ \{\text{DC, EC}\}\ \texttt{flood\_plain}\}$$
$$C_{\text{Target}} = \{\texttt{low\_water\_channel}\ \{\text{NTPP}\}\ \texttt{flood\_plain}\}$$
$$C_{\varrho} = \{\texttt{plot}(x)\ \{\text{EQ}\}\ \texttt{plot(Miscanthus)}\}$$

The first step in the algorithm is to add missing variables and constraints. In the example, all four variables are present in both QCNs, but some relations are missing, e.g. between $\texttt{plot(Miscanthus)}$ and $\texttt{low\_water\_channel}$. A constraint $\texttt{plot(Miscanthus)}$ {DC, EC, PO, TPP, NTPP, TPPi, NTPPi, EQ} $\texttt{low\_water\_channel}$ is therefore added to both $\mu$ and $\psi$.

This manipulation may complexify the QCNs, which is part of the reason why computing the algebraic closure is interesting. Here, the amount of potential scenarios is reduced from 1024 to 16 for $\psi$, and from 1024 to 4 for $\mu$.

Then, the lower bound on the distance between $\psi$ and $\mu$ is computed. Here, the lower bound is 3, which happens to be the distance between $\psi$ and $\mu$. Only one scenario $\mathcal{T}$ of $\mu$ is found at this distance: $\psi \dotplus \mu = \{\mathcal{T}\} = \{(V_{\mathcal{T}}, C_{\mathcal{T}})\}$ with

$$C_{\mathcal{T}} = \left\{ \begin{array}{l} \texttt{low\_water\_channel}\{\text{NTPP}\}\ \texttt{flood\_plain} \\ \texttt{low\_water\_channel}\{\text{DC}\}\ \texttt{plot}(x) \\ \texttt{low\_water\_channel}\{\text{DC}\}\ \texttt{plot(Miscanthus)} \\ \texttt{flood\_plain}\{\text{EC}\}\ \texttt{plot}(x) \\ \texttt{flood\_plain}\{\text{EC}\}\ \texttt{plot(Miscanthus)} \\ \texttt{plot}(x)\ \{\text{EQ}\}\ \texttt{plot(Miscanthus)} \end{array} \right\}$$

The distance is the sum of the following replacements: EC becomes DC between $\texttt{plot}(x)$ and $\texttt{low\_water\_channel}$ ($d = 1$), {TPP, NTPP, PO} becomes EC between $\texttt{flood\_plain}$ and $\texttt{plot}(x)$ ($d = 1$), and {DC, EC, NTPPi, PO} becomes EQ between $\texttt{plot}(x)$ and $\texttt{plot (Miscanthus)}$ ($d = 1$).

In this scenario, the region $\texttt{plot}(x)$ was reduced in order not to overlap with $\texttt{flood\_plain}$ as it was equated to $\texttt{plot (Miscanthus)}$. This corresponds to the allocation shown in figure 1b. It can be seen that the modification is indeed minimal, as the plot becomes externally connected to the flood plain, maximising the area used for Miscanthus cultivation. For instance, a result including $\texttt{flood\_plain}$ {DC} $\texttt{plot(Miscanthus)}$ would have been consistent with the domain knowledge but would not have constituted a minimal modification of $\psi$. Therefore, the adaptation is successful.

## 6.2 Complete agronomic problem

To obtain the more productivity-increasing adaptation described in section 2.1, the plot region is broken into $\texttt{plot1}(x)$ and $\texttt{plot2}(y)$, and $\varrho = x \rightsquigarrow \texttt{Miscanthus}$ ; $y \rightsquigarrow \texttt{maize}$. In $\psi$, the only information about both plots is that they are externally connected to the low water channel.

The revision algorithm returns 5 scenarios. All of them address the domain knowledge and the constraint to maximise the size of the Miscanthus plot, but

they vary in their allocation of maize. One corresponds to the allocation shown in figure 1c. The other ones are similar.

This example also shows that the algorithm handles multiple substitutions.

### 6.3 Cooking example

Most temporal aspects of recipes can be represented in $\mathcal{INDU}$ by reifying cooking actions, ingredient states, and durations as intervals. For instance, the following could be included in the domain knowledge: $\texttt{cooking(carrot)}\ \{\texttt{m}\}^?$ $\texttt{cooked(carrot)}$ and $\texttt{cooking(carrot)}\ ?^=\ \texttt{25\_min}$, with the provision that, e.g. $\texttt{18\_min}\ ?^<\ \texttt{25\_min}$.

In such a simple problem, combining constraints is straightforward and makes it possible to limit the amount of variables. The problem described in section 2.2 can be compressed to just 4 variables by replacing duration intervals by duration relations between the relevant action intervals. In this representation, $\psi$ contains

$$C_{\text{DK}} = \left\{ \begin{array}{l} \texttt{cooking(rice)}\ ?^<\ \texttt{cooking(carrot)} \\ \texttt{cooking(rice)}\ \{\texttt{m}\}^?\ \texttt{serve} \\ \texttt{cooking(carrot)}\ \{\texttt{m}\}^?\ \texttt{serve} \end{array} \right\}$$

$$C_{\alpha(\text{Source})} = \left\{ \texttt{cooking}(x)\{\texttt{f}^<\}\texttt{serve} \right\}$$

In TAAABLE, there is no firm adaptation constraint from $\texttt{Target}$ ($C_{\texttt{Target}} = \emptyset$) therefore $\mu$ contains simply the constraints

$$C_{\text{DK}} = \left\{ \begin{array}{l} \texttt{cooking(rice)}\ ?^<\ \texttt{cooking(carrot)} \\ \texttt{cooking(rice)}\ \{\texttt{m}\}^?\ \texttt{serve} \\ \texttt{cooking(carrot)}\ \{\texttt{m}\}^?\ \texttt{serve} \end{array} \right\}$$

$$C_{\varrho} = \left\{ \texttt{cooking}(x)\ ?^=\ \texttt{cooking(carrot)} \right\}$$

The revision algorithm returns two scenarios which are predictably distinguished only by the duration relation between $\texttt{serve}$ and the other actions, since this relation is defined as being unimportant in the domain knowledge. One scenario $\mathcal{T} = (V_{\mathcal{T}}, C_{\mathcal{T}})$ is such that $C_{\mathcal{T}}$ is

$$\left\{ \begin{array}{ll} \texttt{cooking}(x)\{\texttt{m}^>\}\texttt{serve}, & \texttt{cooking}(x)\{\texttt{eq}^=\}\texttt{cooking(carrot)} \\ \texttt{cooking(carrot)}\{\texttt{m}^>\}\texttt{serve}, & \texttt{cooking}(x)\{\texttt{fi}^>\}\texttt{cooking(rice)} \\ \texttt{cooking(rice)}\{\texttt{m}^>\}\texttt{serve}, & \texttt{cooking(carrot)}\{\texttt{fi}^>\}\texttt{cooking(rice)} \end{array} \right\}$$

In both scenarios, the lengthening of the vegetable cooking is associated with the inversion of the relation between the vegetable and the rice, i.e. $\texttt{f}^<$ becomes $\texttt{fi}^>$, which corresponds to the expected order inversion between the start of both actions. Therefore, the adaptation is successful.

## 7 Related Work

Some recent work deals with a combination of CBR and spatial reasoning, for instance in order to improve web services for spatial information [18], or for

spatial event prediction in hostile territories [19]. Older work already underlined the interest of CBR to analyse geographical data, e.g. for soil classification [20]. We worked on a CBR system to help agronomists analysing farm surveys [21]. The model was based on conceptual graphs, with labelled vertices and edges, describing the spatial organisation of farm territories. The assumption was that similar spatial organisations correspond to similar functional organisations. The *spatio-functional cases* were represented within a description logic system, and reasoning relied on a combination of hierarchical classification (in the description logic sense), CBR and QSTR.

Several research work focused on the representation of time within the CBR framework. Most were interested in the analysis or in the prediction of temporal processes (e.g. breakdown or disease diagnosis starting from regular observations or successive events). The temporal aspect is generally taken into account from sequences of events or sometimes from relative or absolute time stamps [22–24]. Particularly, the problem of temporal adaptation has been given much attention in CBR with a workflow representation [25]. Only a few work [26, 27] adopted a qualitative representation of time, such as the Allen interval algebra. In [27], cases are represented by temporal graphs and the retrieval step is based on graph matching. In [26], cases are indexed by chronicles and temporal constraints, which are represented with a subset of Allen relations. Case-based planning (CBP, see e.g. [28]) is a research field which also deals with time. The main difference between classical CBP and our approach is that they deal with different types of knowledge: CBP deals with the achievement of goals and models actions by their applicability and effects, whereas our approach, applied to a process represented by temporal constraints between actions, deals with the known constraints between actions reified as intervals. Theoretically, these approaches could be combined, but they are generally designed for different purposes: classical CBP usually deals with efficiency and uses complete problem-solving knowledge whereas our approach deals with incompletely described knowledge (e.g. the effects of cooking actions are incompletely formalised).

## 8  Conclusion

Qualitative algebras are important to the field of knowledge representation and are especially useful for qualitative reasoning on space and on time, but their use in CBR has received very little attention so far. This paper focuses on the adaptation of cases represented in a qualitative algebra. A landscape agronomy example uses the spatial algebra RCC8, and a cooking example uses the temporal algebra $\mathcal{INDU}$. This adaptation uses the principles of revision-based adaptation and combines it with a matching between the source and target cases.

A prototype for adaptation of cases represented in a qualitative algebra has been implemented in Perl and applied to the examples of this paper, but it is very time-consuming and requires a lot of improvement in order to be integrated into an operational system like TAAABLE. Several optimisations are planned. First,

the program can be optimised thanks to certain characteristics of the revision problem that are not taken into account by the current prototype:

– The fact that the source case usually represents a specific problem-solving episode, thus the QCN `Source` is usually satisfied by only few scenarios.
– The fact that the difference between QCNs arise because of $\sigma$, which should therefore be looked upon as the possible origin of inconsistencies.

Second, the study of how a process of "repair propagation" in a QCN can be designed is planned. This is similar to the classical constraint propagation algorithm, and can also be likened to the adaptation process presented in [11].

As the landscape agronomy example shows, the QCN can appear in the query of the CBR system. This means that for such an application, the comparison of QCNs also has to be done at retrieval time. A future work will be to study how to implement such a retrieval process.

At a more abstract level, this work, as well as all the previous studies on the use of belief revision and belief merging for single and multiple case adaptation (cf. section 3.2), shares some intuitive ideas with the notion of reuse based on asymmetric and symmetric amalgams (see, e.g. [29]). A precise comparison between these two general approaches to adaptation remains to be carried out.

## References

1. Clifton-Brown, J.C., Stampfl, P.F., Jones, M.B.: Miscanthus biomass production for energy in Europe and its potential contribution to decreasing fossil fuel carbon emissions. Global Change Biology **10**(4) (April 2004) 509–518
2. Hillier, J., Whittaker, C., Dailey, G., Aylott, M., Casella, E., Richter, G.M., Riche, A., Murphy, R., Taylor, G., Smith, P.: Greenhouse gas emissions from four bioenergy crops in England and Wales: Integrating spatial estimates of yield and soil carbon balance in life cycle analyses. GCB Bioenergy **1**(4) (2009) 267–281
3. Martin, L., Wohlfahrt, J., Le Ber, F., Benoît, M.: L'insertion territoriale des cultures biomasses pérennes : le cas du miscanthus par dix agriculteurs de Côte d'Or. L'Espace Géographique (2012) In press.
4. Zub, H.W., Brancourt-Hulmel, M.: Agronomic and physiological performances of different species of Miscanthus, a major energy crop. a review. Agronomy for Sustainable Development **30**(2) (2010) 201–214
5. Cojan, J., Dufour-Lussier, V., Gaillard, E., Lieber, J., Nauer, E., Toussaint, Y.: Knowledge extraction for improving case retrieval and recipe adaptation. In: Computer Cooking Contest Workshop. (2011)
6. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the Logic of Theory Change: partial meet functions for contraction and revision. Journal of Symbolic Logic **50** (1985) 510–530
7. Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. Artificial Intelligence **52**(3) (1991) 263–294
8. Lieber, J.: Application of the Revision Theory to Adaptation in Case-Based Reasoning: the Conservative Adaptation. In: Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR-07). Springer (2007) 239–253
9. Cojan, J., Lieber, J.: Conservative adaptation in metric spaces. In: Advances in Case-Based Reasoning, Springer (2008) 135–149

10. Cojan, J., Lieber, J.: Belief merging-based case combination. In: Case-Based Reasoning Research and Development, Springer (2009) 105–119
11. Cojan, J., Lieber, J.: An Algorithm for Adapting Cases Represented in $\mathcal{ALC}$. In: 22th Internationational Joint Conference on Artificial Intelligence. (2011)
12. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM **26**(11) (November 1983) 832–843
13. Pujari, A.K., Kumari, G.V., Sattar, A.: INDU: An interval & duration network. Advanced Topics in Artificial Intelligence (1999) 291–303
14. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Knowledge Representation. (1992) 165–176
15. Ligozat, G.: On generalized interval calculi. In: Proceedings of the 9th National Conference of the American Association for Artificial Intelligence (AAAI), Anaheim, CA, AAAI Press/MIT Press (1991) 234–240
16. Condotta, J.F., Kaci, S., Schwind, N.: A framework for merging qualitative constraints networks. In Wilson, D., Lane, H.C., eds.: FLAIRS Conference, AAAI Press (May 2008) 586–591
17. Goguen, J.A.: What is unification? In: Resolution of Equations in Algebraic Structures. Academic Press (1989) 217–261
18. Osman, T., Thakker, D., Yang, Y., Claramunt, C.: Semantic spatial web services with case-based reasoning. In: Web and Wireless Geographical Information Systems, Proceedings of the 6th International Symposium, W2GIS 2006. LNCS 4295, Hong Kong, China, Springer (December 2006) 247–258
19. Li, H., Muñoz-Avila, H., Bransen, D., Hogg, C., Alonso, R.: Spatial event prediction by combining value function approximation and case-based reasoning. In McGinty, L., Wilson, D., eds.: ICCBR 2009. LNAI 5650, Springer-Verlag (2009) 465–478
20. Holt, A., Benwell, G.: Case-Based Reasoning and Spatial Analysis. Journal of the Urban and Regional Information Systems Association **8** (1996) 27–36
21. Le Ber, F., Napoli, A., Metzger, J.L., Lardon, S.: Modeling and comparing farm maps using graphs and case-based reasoning. Journal of Universal Computer Science **9**(9) (September 2003) 1073–1095
22. Dojat, M., Ramaux, N., Fontaine, D.: Scenario recognition for temporal reasoning in medical domains. Artificial Intelligence in Medicine **14** (1998) 139–155
23. Ma, J., Knight, B.: A Framework for Historical Case-Based Reasoning. In: Case-Based Reasoning Research and Development, Springer (2003) 246–260
24. Sánchez-Marré, M., Cortés, U., Martínez, M., Comas, J., Rodríguez-Roda, I.: An approach for temporal case-based reasoning: Episode-based reasoning. In: Proceedings of ICCBR 2005. Volume 3620 of LNAI. (2005) 465–476
25. Minor, M., Bergmann, R., Görg, S., Walter, K.: Towards case-based adaptation of workflows. In: Case-Based Reasoning Research and Development (ICCBR 2010). Volume 6176., Springer (July 2010) 421–435
26. Jaczynski, M., Trousse, B.: WWW assisted browsing by reusing past navigations of a group of users. In: Advances in Case-Based Reasoning. Springer (1998) 160–171
27. Dørum Jære, M., Aamodt, A., Skalle, P.: Representing temporal knowledge for case-based prediction. In: Advances in Case-Based Reasoning, Proceedings of EC-CBR 2002. Volume 2416 of LNAI., Springer (2002) 174–188
28. Munoz-Avila, H., Cox, M.: Case-Based Plan Adaptation: An Analysis and Review. Intelligent Systems, IEEE **23**(4) (july-aug. 2008) 75–81
29. Manzano, S., Ontañón, S., Plaza, E.: Amalgam-based reuse for multiagent case-based reasoning. In: Case-Based Reasoning Research and Development, Springer (2011) 122–136