The Algorithm to Retrieve Temporal Cases for Temporal Case-Based Reasoning

Kurilenko I.E.
National Research University
Moscow Power Engineering Institute
Moscow, Russia
Email: ivan@appmat.ru

Guliakina N.A.
Belorussian State University of
Informatics and Radioelectronics
Minsk, Republic of Belarus
Email: guliakina@bsuir.by

Abstract—**In this paper the problem of the application of temporal reasoning and case-based reasoning (CBR) in intelligent decision support systems (IDSS) is considered. The way to expand the existing case-based reasoning mechanisms to temporal case-based reasoning mechanism explained.**

***Keywords*—Intelligent decision support systems, temporal reasoning, analogous reasoning, case-based reasoning, temporal logic, temporal constraint satisfaction problem.**

I. ABSTRACT

An important task in the development of advanced intelligent systems such as intelligent decision support systems of real-time (IDSS RT) is the problem of modelling common sense reasoning [1-2].

Along with widely used in the field of artificial intelligence (AI) techniques such as inductive reasoning, abduction, argumentation and analogy are actively developed methods of case-based reasoning (CBR-methods) [3-5]. Temporal reasoning and case-based reasoning (CBR) can be used in various applications of AI and for solving various problems, e.g., for diagnostics and forecasting or for machine learning [6-8].

Case-based reasoning, like analogous reasoning, is based on analogy, however, there are certain differences in the implementation. A precedent is defined as a case that took place earlier and is an example or justification for subsequent events of this kind. Case-based reasoning allows to make decisions in new, unknown situations, using or adapting the decision taken earlier in the already known situation, (i.e. using already acquired early decision-making experience) [4].

CBR-methods are well developed and widely used in practice.

Typically, classical CBR-methods allows to extract cases based on the values of the parameters controlled by the system at the current time, but without taking into account the dynamics of the process (i.e. the situation described as the "snapshots" of the control parameters of the observed object or system with no account of their history changes).

Given the fact that the nature of the physical processes is that for the same values of parameters of controlled process often can go with different ways, which will largely determine the future dynamics of their changes, sharply raises the question of improving the CBR-methods by giving them the possibility of taking into account the temporal and causal relationships.

Thus, the need for a new way of presenting the precedents allows to take into account parameters change history - a method based on temporal precedents, as well as adapted to take into account the time factor extraction algorithms precedents.

This work is proposed extension of CBR-methods that allows taking into account the behaviour of the controlled process or object over time.

When taking into account the time factor it is possible to consider the problem situation in the dynamics, that is, the current situation is not compared with fixed parameters snapshot from the case. We can take into account changes of this parameters over time, their behaviour.

II. TEMPORAL CASE

There are several methods to build the temporal case-based reasoning mechanism on the basis of the existing snapshot-based approaches: we can replace snapshot with historical set of the snapshots or we can introduce the time explicitly.

The first method corresponding to the changes modelling. Each snapshot in the history is the system state. These states are regarded as momentary pictures of the object, which don't have any time duration. Time itself is regarded implicitly, via modelling of the system changes within time.

Method of the replacing snapshot with historical set of the snapshots is the easiest, but it often require storing a huge volume of information. Also this approach have constraints when presenting complex time dependences (events, which have duration, continuance of processes, causal relations etc.).

Some different ways to eliminate this can be found, however in the most cases they are reduced to introduction of an explicit time model. Explicit time modelling provides the possibility to make flexible formalized languages, which help to do reasoning on the basis of expressions, truth values of which are timed to the definite moment or time interval, and they can change in the course of time [9]. Time is presented explicitly, taking into consideration its properties. Time can be presented both syntactically (via explicit temporary structures) and semantically (modal logics are typical representatives of this approach). Also we gain ability to use temporal structure to speed up reasoning mechanism.

So, more promising in this light looks methods of temporal case construction based on temporal logic.

In this case, the information about time is separated from the information about parameters. This reduces the amount of stored information, and have a positive effect on the inference algorithms that should process less volume of data. In addition, we can get the advantages of possibility to determine similarity in two levels - on the level of the temporal structure and the level of a parametric snapshot comparison at the specific time moments.

In this paper, we consider the models, based on the presentation of information about time as constraints (dependences) between time primitives. In temporal logics using the concept of constraint satisfaction, information about time is presented as dependences between temporal primitives (moments, intervals or their combinations). Dependences between primitives are interpreted as constraints to real time of their appearance. Usually sets of temporal primitives and relations among them are presented as the Temporal Constraint Satisfaction Problem (TCSP), which is detailing of a more general Constraint Satisfaction Problem (CSP), what permits to use CSP methods to solve the TCSP.

Lets see how temporal case based reasoning can be build on the base of the metric temporal constraint satisfaction problem (Metric TCSP) [9].

Metric TCSP defined as $Z = (V, D, C)$, where:

- V - a finite set of temporal variables, corresponding to the time points;
- D - range of values of the temporal variables (the set of integers);
- C - a finite number of binary temporal constraints $C_{ij} = [a_1, b_1], \ldots, [a_k, b_k]$, where the intervals are disjoint.

Each constraint C_ij defines for the temporal variables V_i and V_j allowable distance between them. Intervals in the constraint C_ij are interpreted as a disjunctive [3].

To define the model of temporal case the situation definition introduced as $S =< V, P, D, C >$, where

- $V = V_1, \ldots V_m$ - finite set of temporal variables (interpreted as the time points);
- D - range of values of the temporal variables (the set of integers);
- C - a finite number of binary temporal constraints $C_{ij} = [a_1, b_1], \ldots, [a_k, b_k]$, where the intervals are disjoint; $P = P_1, \ldots P_k$ – the set of the parameters of the controlled object.

Temporal case defined as the situation supplemented with a diagnosis and recommendations to the decision maker (DM) – $Tc =< V, P, D, C, Q >$, where Q - the DM recommendations.

When looking for a case applicable to the situation observed an algorithm that takes into account the temporal constraints are used.

An algorithm for constructing temporal case on the history of parameters changes contains several steps.

The first stage uses algorithm 1 to compress the history of parameters changes to the series of events $S = e_1, \ldots e_r$, where:

- $e_i = (t_i, P_i)$ – event description,
- $t_i \in Z$ - time of the event,
- $P_i = (p_1, \ldots p_k)$ – the parameters set, described object state at time t_i.

On the second stage, we extracting metric TCSP using algorithm 2.

Algorithm 1 Temporal situation construction based on the history of the parameters sets

Input: $H = H_i$ - history, where:

- $H_i = (p_1, ..p_k)$ – object parameters set on the i-th tact,
- τ – the count of the recorded tact's.

Output: $S = e_i$ – situation, where:

- $e_i = (t_i, P_i)$ – event,
- $t_i \in Z$ – time point of the observation of the e_i,
- $Pi = (p_1, ..p_k)$ – object parameters set at the t_i

Code:

```
$k \leftarrow 0$
$S \leftarrow S \cup \{(k, H_k)\}$
for ($i \leftarrow 1$ to $\tau - 1$) do
    if ($H_i \neq H_k$) then
        $S \leftarrow S \cup \{(i, H_k)\}$
        $k \leftarrow i$
    end if
end for
return $S$
```

This algorithm produces metric TCSP with non-disjunctive constraints from STP subclass (Simple Temporal Problems).

If we need update existing case with new information, we can use algorithm 3 to merge cases.

Note that we can use different strategies to combine constraints at the line 8 of the algorithm 3.

If we simple combine them with union operation we can increase fragmentation level (the disjunctive constraints can appear and resulted TSP will leave STP subclass). To avoid this and at the same time soft the resulted constraint we can apply the method of upper-lower tightening [12].

In this case, will be allowed any value of the time of occurrence of the event in between the lower and upper boundary (Fig. 1).

Note that the algorithm 3 assumes that the state of the controlled object is identical in the merged cases at corresponding moments of time.

In practice, however, such a condition is sufficiently rigid.

So it makes sense to implement the algorithm that averages parameters sets for the combined time point (for example such as done in the algorithm 4) or we can implement the transition from the exact value of the parameter to an acceptable range.

Algorithm 2 Temporal case construction

Input: $S = e_i$, i=0..n, where:

- $e_i = (t_i, P_i)$ – event,
- $t_i \in Z$ – time of the event observation,
- $P_i = (p_1, ..p_k)$ – object parameters set at the time t_i,
- Q – diagnosis.

Output: $U = (V, P', D, C, W)$ – temporal case, where:

- $V = V_i, i = 0..n$ – finite set of the temporal variables (time points), $V_i \in R$,
- $P' = P'_i$ – object parameters set at the time moment $V_i, P'_i = (p_1, ..p_k)$,
- D - domain of the temporal variables;
- C – set of temporal constraints $C_{ij} = [a_1, b_1], \ldots, [a_k, b_k]$,
- Q – diagnosis.

Code:

```
$e_i \leftarrow e_i = (t_i, P_i) \in E : \forall j \neq i t_j > t_i, where e_j = (t_j, P_j)$
$S \leftarrow S \cup \{(k, H_k)\}$
for all $(e_k \in S)$ do
    $e_k \leftarrow (t_k - t_i, P_k)$
    $V \leftarrow V \cup \{V_k\}$
    $P' \leftarrow P' \cup \{P_k\}$
end for
for $(i \leftarrow 0\ to\ n+1)$ do
    for $(j \leftarrow 0\ to\ n+1)$ do
        if $(i \neq j)$ then
            $C_{ij} \leftarrow \{[t_j - t_i, t_j - t_i]\}$    ▷ $(where\ t_i\ and\ t_j\ defined\ from\ the\ e_i = (t_i, P_i)\ and\ e_j = (t_j, P_j))$
        else
            $C_{ij} \leftarrow$
        end if
    end for
end for
return $U = (V, P', D, C, Q)$
```

Algorithm 3 Merging temporal cases

Input: U^1, U^2 - cases to merge, where:

- $U^l = (V^l, P^l, D^l, C^l, Q^l)$ – temporal case, in which $V^l = V^l_i$, $i = 0..n$, – finite set of temporal variables, $V_i \in Z$, $P^l = \{P^l_i\}$ – object parameters set at the moment V^l_i, $P^l_i = (p_1, ..p_k)$, D^l – domain of the temporal variables; ^{l} – finite set of temporal constraints $C^l_{ij} = \{[a_1, b_1], \ldots, [a_k, b_k]\}$, Q – diagnosis.
- Assumed that $|V^1| = |V^2|$ and $P^1 = P^2$.

Output: $M = (V, P', D, C, Q)$ – temporal case, where:

- V=Vi, i=0..n, – finite set of temporal variables,
- $Vi \in Z$,
- $P = \{P_i\}$ – object parameters set at the moment V_i, $P'_i = (p_1, ..p_k)$,
- D - domain of the temporal variables;
- C – finite set of temporal constraints $C_{ij} = \{[a_1, b_1], \ldots, [a_k, b_k]\}$. t_i

Code:

```
if then$(|V^1| \neq |V^2|)$
    return                                   ▷ Couldn't merge cases
end if
                                             ▷ Softening temporal constraints
for $(i \leftarrow 0\ to\ n+1)$ do
    for $(j \leftarrow 0\ to\ n+1)$ do
        if $(i \neq j)$ then
            $C*_{ij} \leftarrow C^1_{ij} \cup C^2_{ij}$
        end if
    end for
end for
return $M = (V^1, P', D, C^*, Q)$
```

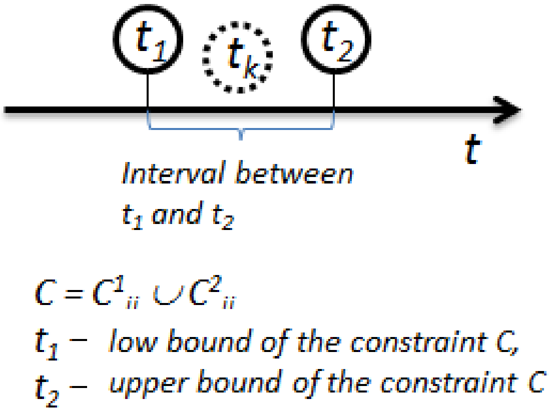

Figure 1. Illustration of the strategy to reduce the fragmentation of the metric temporal constraints on temporal case construction

III. CONCLUSIONS

The case-based temporal reasoning allows taking into account the sequence of the events and their durations.

If the metric information didn't needed on the level of the description of the temporal structure of the temporal case, we can use another base model to build it. For example we can use qualitative TCSP, based on the point algebra for example, for which exists effective reasoning algorithms [7, 11, 13].

Described models and algorithms can be used to extend the capabilities of modern CBR-systems, allowing to implement temporal case-based reasoning, taking into account the course of the controlled process, the history of its transition to the observed situation.

For example, described methods and algorithms was widely used to build intelligent features of payable parking access control system sPARK.

Modern parking solutions is the complicated complexes, which are equipped with an automatic barriers, the video cameras, fire and access alarm, etc.. The major target of the car access control system is passage control of the cars, registration of the visitors and the car owners, stealing prevention. The object of access in this system is the car. The execution units are the barriers and the gates, which system should open before the passage and close after car entrance completed. So, the system should control that the car successfully entered to the parking territory. The necessity to control the driving process leads to take into account the temporal dependencies [14]. The ability of analysis of the sequences of observed by the system actions permits to implement more reliable and intelligent

Algorithm 4 Merging temporal cases

Input: U^1, U^2 - cases to merge, where:

- $U^l = (V^l, P^l, D^l, C^l, Q^l)$ – temporal case, in which $V^l = V_i^l$, $i = 0..n$, – finite set of temporal variables, $V_i \in Z$, $P^l = \{P_i^l\}$ – object parameters set at the moment V_i^l, $P_i^l = (p_1, ..p_k)$, D^l – domain of the temporal variables; ^{l} – finite set of temporal constraints $C_{ij}^l = \{[a_1, b_1], \ldots, [a_k, b_k]\}$, Q – diagnosis.
- Assumed that $|V^1| = |V^2|$ and $P^1 = P^2$.

Output: $M = (V, P', D, C, Q)$ – temporal case, where:

- V=Vi, i=0..n, – finite set of temporal variables,
- $Vi \in Z$,
- $P = \{P_i\}$ – object parameters set at the moment V_i, $P_i' = (p_1, ..p_k)$,
- D - domain of the temporal variables;
- C – finite set of temporal constraints $C_{ij} = \{[a_1, b_1], \ldots, [a_k, b_k]\}$. t_i

Code:

```
if then($|V^1| \neq |V^2|$)
    return                                  ▷ Couldn't merge cases
end if
                                            ▷ Averaging parameters values
for ($i \leftarrow 0\ to\ n+1$) do
    $P_i \leftarrow \frac{P_i^1 + P_i^2}{2}$
end for
                                            ▷ Softening temporal constraints
for ($i \leftarrow 0\ to\ n+1$) do
    for ($j \leftarrow 0\ to\ n+1$) do
        if ($i \neq j$) then
            $C*_{ij} \leftarrow C_{ij}^1 \cup C_{ij}^2$
        end if
    end for
end for
return $M = (V^1, P', D, C^*, Q)$
```

solution.

This work was supported by BRFBR-RFBR, grant 16-51-00058, "Temporal reasoning formalisation in intelligent systems".

REFERENCES

[1] D. A. Pospelov and L. V. Litvintseva, The Representation of Knowledge about Time and Space in Intelligent Systems, Ed. by D. A. Pospelov (Nauka, Moscow, 1986) [in Russian].

[2] V. N. Vagin and A. P. Eremeev, “Certain Basic Principles of Designing Intelligent Real-Time Systems for Decision Support,” Izv. Ross. Akad. Nauk, Teor. Sist. Upr., No. 6, (2001) [Comp. Syst. Sci. 40 (6), 707 (2001)]

[3] Eremeev A.P., Kurilenko I.E. Extending temporal dependencies modelling ability for intelligent systems using temporal cases / Intelligent systems. Collective monograph. Number 6. / Under editing by V.M. Kureichick. – M.: FizMatLit, 2013, 300 p., pp. 89-118.

[4] Varshavsky P.R., Kurilenko I.E., Makashova M.B. Implementing temporal case-based reasoner based on neural network approach for intelligent decision support system // In proc. XIII national conference on Artificial intelligence CAI-2012. .3. – Belgorod:BGTU, 2012. - pp.80-87.

[5] David C. Wilson, David B. Leake Maintaining Case-Based Reasoners: Dimensions and Directions // Computational Intelligence, Volume 17, Number 2, 2001, pp. 196-213.

[6] Varshavskii P.R., Eremeev A.P. Modeling of case-based reasoning in intelligent decision support systems // Scientific and Technical Information Processing, 37 (5), 2010, pp. 336-345.

[7] Eremeev A.P., Kurilenko I.E. Temporal reasoning component for real-time intelligent decision-support systems // Scientific and Technical Information Processing, 38 (5), 2011, pp. 332-343.

[8] Karpov U. G. MODEL CHECKING. Verifikaciya parallelnyh i raspredelennyh programmnyh sistem/ Uriy Glebovich Karpov – SPb.: BHV - Peterburg, 2010 – 560 p.

[9] Kurilenko I.E. Constructing temporal case-base reasoning mechanism // In proc. XX conference Information tools and technologies – M.:MPEI Publishing, 2012. - pp.129-136.

[10] Kurilenko I.E., Smirnova A.V. The implementation of typical temporal case analyser for intelligent system //In proc. XX conference Information tools and technologies – M.:MPEI Publishing, 2012. - pp.152-158.

[11] Kurilenko I.E. Metric temporal logic application to build temporal case-based reasoning mechanizm // In proc. XIII national conference on Artificial intelligence CAI-2012. .1. – Belgorod:BGTU, 2012. - pp.25-33.

[12] Eddie Schwalb and Rina Dechter. Coping with disjunctions in Temporal Constraint Satisfaction Problems. In AAAI 1993: Proceedings of the 11th National Conference on Artificial Intelligence, pages 127-132, 1993.

[13] Alfonso Gerevini, Lenhart Schubert, Efficient algorithms for qualitative reasoning about time, Artificial Intelligence, Volume 74, Issue 2, 1995, Pages 207-248, ISSN 0004-3702, http://dx.doi.org/10.1016/0004-3702(94)00016-T.

[14] Ivan E. Kurilenko, Pavel R. Varshavskiy, Alexander P. Eremeev Temporal Case-Based Reasoning System for Automatic Parking Complex / International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol 2., 5, 2015.

АЛГОРИТМ ИЗВЛЕЧЕНИЯ ТЕМПОРАЛЬНЫХ ПРЕЦЕДЕНТОВ ДЛЯ МЕХАНИЗМА РАССУЖДЕНИЙ НА ОСНОВЕ ТЕМПОРАЛЬНЫХ ПРЕЦЕДЕНТОВ

Куриленко И.Е., Гулякина Н.А.

В статье рассматривается проблема применения механизма рассуждений на основе темпоральных прецедентов в составе современных интеллектуальных систем типа интеллектуальных систем поддержки принятия решений реального времени. Рассматриваются способы перехода к темпоральным прецедентам через явное и неявное введение фактора времени. Предлагается модель представления прецедента на основе метрической точечной временной логики. Объясняется механизм извлечения темпоральных прецедентов и приводятся алгоритмы формирования прецедента и слияния (обобщения) прецедентов.