On Specifying and Analysing Domain Ontologies for Workflows in "Binary Model of Knowledge"

Gerald S. Plesniewicz[1] and Valery B. Tarasov[2]
[1]Applied Mathematics Department, National Research University MPEI, Krasnokazarmennaya 14, Moscow, Russia
[2]CIM Department, Bauman Moscow State Technical University, 2nd Baumanskaya 5, Moscow, Russia

Keywords: Knowledge Bases, Ontologies, Workflows, Concept Languages, Logical Inference, Analytic Tableaux.

Abstract: The main purpose of the present paper is to show how concept languages of the system "Binary Model of Knowledge" can be used for specifying workflow ontologies. The system is under development in the Applied Mathematics and Artificial Intelligence Department of National Research University MPEI (Moscow). In particular, the system includes the language LTS of temporal specification. The language includes the sentences matching the sentences of the Boolean and metric extensions of Allen's interval logic. For the extended logics we present the complete systems of inference rules (in style of analytic tableaux).

1 INTRODUCTION

Workflow is a representation of a process whose participants (agents which are humans or programs), perform, having a common goal, some set of tasks in accordance with certain rules and constraints (Aalst et al., 2002).

The concept of workflow appeared in business informatics. But at present, the workflow technique is used in many other areas such as medical informatics, bioinformatics (in particular, genomics), scientifique process automation et al.An important application of workflows is the design of web services.

An ontology is based on a conceptualization. A conceptualization is an abstract, simplified view of the subject world that we wish to represent. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization. An ontology is an explicit specification of a conceptualization (Gruber, 1993).

There are logical approaches to modeling and analysis of workflows. In such cases a workflow is considered as an instance of a workflow scheme, and the scheme is written as a set of sentences in appropriate logic. Then we get the opportunity to express properties of workflows and to verify them using logical procedures. In particular, the co-called Kifer's transaction logic was applied (Davulcu, 1989), (Mukherjee et al., 2002).Also, temporal logics was used for analyzing workflows (Bettini, 2002).

A conceptualization ***Cz***of a workflow scheme ***S***for a real application contains many concepts and relations between them.It is natural to define in concept languages an ontology ***O***that specifies the conceptualization ***Cz***.

The main purpose of the present paper is to show how concept languages of the system "Binary Model of Knowledge" can be used for specifying workflow ontologies. The system is under development in the Applied Mathematics Department of National Research University MPEI (Moscow). In particular, the system includes the language **LTS** of temporal specification. The language includes the sentences matching the sentences of the Boolean and metric extensions of Allen's interval logic. For the extended logics we present the complete systems of inference rules (in style of analytic tableaux (Agostino et al., 2001), (Fitting, 1996)). We show (by examples) how to use the inference systems for recognizing inconsistency of ontologies and for query answering.

2 ABOUT THE SYSTEM "BINARY MODEL OF KNOWLEDGE"

"BinaryModel of Knowledge" is the system of concept languages and tools for their interpretations

Plesniewicz, G. and Tarasov, V.
On Specifying and Analysing Domain Ontologies for Workflows in âĂIJBinary Model of Knowledge".
DOI: 10.5220/0010134902050212
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 2: KEOD*, pages 205-212
ISBN: 978-989-758-474-9

(Plesniewicz, 2014). These languages have semantics based on formal concepts.

A *formal concept* has the following components:

- The *name C* of the concept;
- The *unverse* U^C of the concept – a countable set of names denoting possble instances of the concept. The universe also contains so-called *surrogates* {object-oriented identifyiers): `Surr` = {#1, #2,…} ⊆ U^C.
- The *extension* E^C of the concept, i.e., the set of names that denote*instances* of the concept, $E^C \subseteq U^C$;
- The *coreferentiality* relation $\sim^C \subseteq E^C \times E^C$. If$a \sim^C b$ then the names *a* and *b* denote the same object of the application modelled.

Example 1. Define a formal concept as follows:

- U^{Person} =`String` ∪ {[`Surr`:*x*,`Name`:*y*,`SSN`:*z*,`works_in`:*u*]| *y* ∈`String`, *z* ∈`typeSSN`, *x*, *u* ∈`Surr`};

  Here typeSSN is the attribute domain (data type) for social security numbers (i.e., strings of the format XXX-XX-XXXX, where X are decimal digits);
- E^{Person}= {…,#105,[`Surr:#110, Name:john,SSN:078-05-1120, works_in:#27`],…};
- {…,#105$\sim^{Person}$[`Surr:#105, Name:john,SSN:078-05-1120, works_in:#27`],…}.

The concept `Person` has three attributes: `Name`,`SSN` and `works_in`. The first two attributes take values in the standard data type `String` and in the specified data type type`SSN`.The third attribute takes the value `#27` which is the surrogate referred to some organization where John works.

The concept`Person`from Example 1 is static in the sense that their extensions do not depend on time. In general, the extension of a concept is variable. It is natural to introduce a special attribute `Por`(point of reference) whose values refer to this variability.The attribute `Por` may have such components as time (point or interval), position in space, state of affairs, context, truth degree et al.

For any point of reference γ, we denote by E^C_γ the extension of the concept *C* at the point of reference γ. Let Γ be the set of all possible points of reference that are considered under a given conceptualization. Then we say that the family of sets $\{E^C_\gamma \mid \gamma \in \Gamma\}$ is the*total extension* of the concept *C*,

So, formally conceptualization of a given application can be represented by a (finite) set ***S*** of formal concepts with the same set Γ of points of reference. An *ontology* ***O***that specifies the set S of formal concepts is written in the concept languages of the system BMK.

The sentences of the ontology ***O*** differ in what components of concepts they specify. The sentences that specify concept universes U^C(*C* ∈ ***S***), define the structure of members of U^C, and therefore, we call them *structural* sentences. We call *logical* the sentences that specify the extensions E^C (*C* ∈ ***S***). We also call *transitory* the sentences that specify the changes $(E^C_\gamma - E^C_\delta) \cup (E^C_\delta - E^C_\gamma)$ in the transition from the point γ to the point δ.

In the system "Binary Model of Knowledge", there are the languages for structural, logical and transitory specification of ontologies.

2.1 Language LSS of Structural Specification

In the language **LLS** two type of concepts are distinguished: *classes* and *binary relations*. **LLS** sentences are composed of primitive sentences that have the following forms:

C[*D*], *C*[*A*:*D*], *C*[*A*:*T*], (*CLD*), (*CLD*)[*E*], (*CLD*)[*A*:*E*], (*CLD*)[*A*:*T*].

Here *C*, *D*, *E*are names of classes, *L* is a name of binary relation, *A* is an attribute, and *T* is a data type specification. (There are some means for defining data types in LSS.)

An arbitrary structural sentence is obtained by joining primitive sentences. For example, the sentence *C*[*D*, *A*: (String, Integer), *E*(*)]arises from the primitive sentences*C*[*D*], *C*[*A*: (String, Integer)] and *C*[*D*, *A*:*E*(*)].

Here are some examples of structural sentences.

```
1) Car[Brand:String,Engine,
      Dimensions:
   (Length/m/:Integer,
   Width/mm/:Integer,
   Height/mm/:Integer,
Wheelbase/mm/:Integer)
Gearbox:String].
2) Engine[Type:Integer,
Power/hp/:Integer,
Max_speed/km/h/: Integer].
3)(Person owns Car}
[RegisterDate:Date,
DocsReg:String].
```

The assertion $e \in E^C_\gamma$ corresponds to the fact "*e* is an instance of the concept *C* at the point of reference γ", and $e \notin E^C_\gamma$ corresponds to the fact "*e* is a counter-instance of the concept *C* at the point of reference γ".

In the system BMK facts are represented as tuples of the tables whose headers are determines by **LSS** sentences. For example, the second sentence determines the following table header.

Engine				
Surr	Por	Type	Power	Max_speed

Note that the language **LSS** is essentially a data description language for some object-oriented language model. In the system BMK there is an appropriate query language.

2.2 Language LLS of Logical Specification

There are several types of LLS sentences. Here are some examples of LLS sentences:

```
1) Car ISA Car(Engine.
Max_speed/km/h/ =< 300).
```

(Every car has a maximum speed of not more than 300 kilometers per hour.)

```
2) Minivan == Car(Dimension.
        Length/mm/ =< 3600;
        Width/mm/ =< 1600).
3) NOT EXIST Person THAT own SOME
   Car THAT has a SOME Defect.
```

(There is no person who owns a car with a defect.)

```
4) EACH Product(Brand = AAA)
   Transported_bySOME Minivan.
```

(Each brand AAA product is transported by a minivan owned by the company "TransVan".)

```
5) EACH Product(Brand = BBB) NOT
   Transported_byANYCar THAT
   Belong_to "RoadTrans".
```

(No BBB products are transported by cars of the firm RoadTrans.)

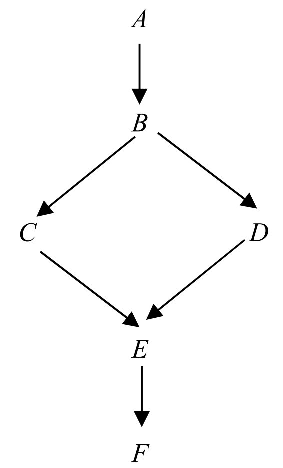
2.3 Language LTS of Temporal Specification

In workflow ontologies, the main role is played by *events*, i.e. concepts whose instances exist in temporal intervals.

If the concept *E* is an event then it has two special attributes Beg(begin) and End. Thus, When we use language **LTS** for specifying such ontologies, we chose events for modelling workflow tasks (works).

Consider a simple example of a workflow.

Example 2. The workflow represents a business process that aims to transport goods by trucksof two companies "TransVan" and "RoadTrans". Figure 2 shows a diagram of tasks and relations between them that determine their possible sequencing.



Legend:
A: order processing
B: invoice registration
C: goods transportation by"TransVan"
D: goods transportation by "RoadTrans"
E: goods unloading
F: payment registration

Figure 1: Example of workflow scheme.

The business process of goods transportation starts with an order processing (task *A*). Then, the invoice is registered (task *D*) and the goods transportation is carried out (tasks *C* and *D*).

Suppose, there is a condition *p* affecting how the transportation is carried out. If *p* is satisfied then all goods are transported by the company "TransVan". Otherwise, ittransports only part of the goods, and the rest is transported by the firm "RoadTrans"; in this case "RoadTrans" starts loading the goods a little later and brings the goods later than "TransVan". After delivery, the goods are unloaded(task *E*). Finely, the payment is registered(task *F*).

In the language **LTS**, the temporal relations between tasks can be write as following workflow ontology:

O = {

```
OrderProc BEFOREInvoiceRegist.
InvoiceRegist BEFORE Transp1.
Transp1 ISA Transp.
Transp2 ISA Transp.
IF CondP THEN
Transp1 OVERLAP Transp2.
Transp BEFORE Unload.
Unload BEFORE PaymentRegist}.
```

This ontology is written in the language **LTL** of temporal specification from the system BMK, more exactly, in the fragment of **LTL** consisted in the sentences that correspond to Boolean extension of Allen's temporalinterval logic **BAL**. With the logic **BAL** we can determine relations between temporal intervals during which works are performed.

For example, the LTL sentence

```
IF W1 BEFORE W2 THEN (W2 START W3)
      OR (W3 START W2).
```

states that if it turns out that work W1 was performed before work W2, then work W2 should be started simultaneously with work W3.

In the ontology ***O***the names`BEFORE`and `OVERLAP` denote are the relations between events that correspond to Allen's relations ***b*** and ***o***between temporal intervals (Allen, 1983). There are 7 Allen's relations and 6 inverse relations:

`BEFORE`(***b***), `MEET`(***m***),`DURING`(***d***),`START`(***s***), `OVERLAP` (***o***), `FINISH` (***f***),`EQUAL` (***e***), `AFTER`($\boldsymbol{b}^{-1}$),`MET-BY`($\boldsymbol{m}^{-1}$),`CONTAIN` ($\boldsymbol{d}^{-1}$), `STARTED-BY`($\boldsymbol{s}^{-1}$),`OVERLAPPED-BY`($\boldsymbol{o}^{-1}$), `FINISHED-BY`($\boldsymbol{b}^{-1}$).

($\boldsymbol{v}^{-1}$ denotes the reverse relation: $A\ \boldsymbol{v}^{-1}B \Leftrightarrow B\boldsymbol{v}A$.)

The language **LTS** contains also the temporal quantifiers `ANYTIME` and `SOMETIME`.

The following sentence specifies the concept "former car owner":

```
'Former car owner' ==Person THAT
Own (SOMETIME X) SOME Car;X
BEFORENow.
```

The following term defines those persons who at the current moment (expressed by the time interval `NOW`) have changed their Audi 200 car to a Toyota Land Cruiser:

```
Person THAT Owns (SOMETIME X)
SOME Car (Brand='Audi 200') AND
Owns (SOMETIME Y)SOME
Car(Make='Toyota land cruiser');
X START NOW; NOW FINISH Y;
X MEET Y.
```

3 BOLEAN AND METRIC EXTENSIONS OF ALLEN' S INTERVAL LOGIC

The above mention workflow ontology ***O*** can be rewrite in Allen's notation as

$\boldsymbol{O}^A$= {A ***b*** BB***b***C, B ***b***D, C ***b*** E, D ***b*** E, E***b*** F, $p \rightarrow D$***o*** F},

where the names of the intervals in ***O*** are renamed accordingly. In general, let $\boldsymbol{O}^A$ denote the result of such renaming for any**LTS** ontology***O***.

It is clear, if the ontology $\boldsymbol{O}^A$ is inconsistent then the ontology ***O*** is also inconsistent. Since the problem of logical consequence is reduced to the problem of inconsistency, then for any sentence φ, ***O***|= φ takes place if $\boldsymbol{O}^A \models \varphi^A$.

3.1 Boolean Extension of Allen's Logic

Let Ω = {***b***, ***m***, ***d***,***s***, ***o***, ***f***, ***e***, $\boldsymbol{b}^{-1}$, $\boldsymbol{m}^{-1}$, $\boldsymbol{d}^{-1}$, $\boldsymbol{s}^{-1}$, $\boldsymbol{o}^{-1}$, $\boldsymbol{b}^{-1}$} . A sentence of Allen's logic **AL** has the form $A\ \omega\ B$ where A, B are temporal interval and ω is a subset of Ω written as a word. (For example, A $\boldsymbol{bdm}^{-1}$$B$ is a **AL** sentence.This sentence is equivalent to the disjunction A ***b*** B∨A ***d***B∨ B***m***A.)

Table 2: Inference rules for propositional connectives.

No	Antecedent	Consequents
1	$+\sim p$	$-p$
2	$-\sim p$	$+p$
3	$+\ p\wedge q$	$+p, +q$
4	$-\ p\wedge \boldsymbol{q}$	$-p\ \|-q$
5	$+\ p\vee q$	$+p\ \|+q$
6	$-\ p\vee \boldsymbol{q}$	$-p, -q$
7	$+\ p\rightarrow q$	$-p\ \|+q$
8	$-\ p\rightarrow q$	$+p, -q$

Table 3: Inference rules for Allen's connectives.

No	Antecedent	Consequent
1	$+A$ ***b*** B	$B^- - A^+ \geq 1$
2	$-A$ ***b*** B	$A^+ - B^- \geq 0$
3	$+A$ ***m*** B	$A^+ = B^-$
4	$-A$ ***m*** B	$A^+ - B^- \geq 1\ \|B^- - A^+ \geq 1$
5	$+A$ ***o***B	$B^- - A^- \geq 1, A^+ - B^- \geq 1, B^+ - A^+ \geq 1$
6	$-A$ ***o***B	$A^- - B^- \geq 0\ \|B^- - A^+ \geq 0\ \|A^+ - B^+ \geq 0$
7	$+A$ ***f***B	$A^- - B^- \geq 1, A^+ = B^+$
8	$-A$ ***f***B	$B^- - A^- \geq 0\|A^+ - B^+ \geq 1\|\ B^+ - A^+ \geq 1$
9	$+A$ ***s***B	$A^- = B^-, B^+ - A^+ \geq 1$
10	$-A$ ***s***B	$A^- - B^- \geq 1\ \|B^- - A^- \geq 1\ \|A^+ - B^+ \geq 0$
11	$+A$ ***d***B	$A^- - B^- \geq 1, B^+ - A^+ \geq 1$
12	$-A$ ***d***B	$B^- - A^- \geq 0\|\ A^+ - B^+ \geq 0$
13	$+A$ ***e***B	$A^- = B^-, A^+ = B^+$
14	$-A$ ***e***B	$B^- - A^- \geq 1\ \|A^- - B^- \geq 1$ $B^+ - A^- \geq 1\ \|A^- - B^- \geq 1$
15	$+A\ \theta^{-1}B$	$+B\theta\ A$
16	$-A\ \theta^{-1}B$	$-B\ \theta A$
17	$+A\ \theta\omega B$	$+A\ \theta\ B\|+A\ \omega B$
18	$-A\ \theta\omega B$	$-A\ \theta\ B, -A\ \omega B$
$\theta \in \Omega,\ \ \omega \subseteq \Omega$		

The Boolean extension **BAL** of Allen's logic **AL** has Boolean combinations of **AL** sentences and propositional variables as its sentences. (For example,

$(p \wedge {\sim}q \rightarrow {\sim}A\ \boldsymbol{b}\ B \wedge B\ \boldsymbol{sdo} C) \vee {\sim}A\ \boldsymbol{f}\ C$

is a **BAL** sentence.)

The tables Table 1 and Table 2 contain the inference rules by analytic tableaux method for the logic **BAL**. (Note that the rules of Table 1 is usual tableaux inference rules for signified propositional formulas (Fitting, 1996).)

Consider an example ofan inference tree for proving logical consequences in the logic **BAL**.

Example 3. Take three sentences

$p \rightarrow A\ \boldsymbol{m}\ B$, ${\sim}B\ \boldsymbol{bm}\ C \rightarrow q$, $A\ \boldsymbol{o}\ D \wedge D\ \boldsymbol{o}\ C$

as an ontology ***O***in the logic **BAL**.

For the sentence $p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D$, let us put the question $\boldsymbol{O} \models p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D$? ("Is it true or not that ***O*** logically implies$p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D$?")

Due the known relation between the problems of inconsistency and logical consequence, we have

$\boldsymbol{O} \models p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D$

if and only if the set $\boldsymbol{O} \cup \{{\sim}(p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D)\}$ is inconsistent, i.e. the set

$E = \{p \rightarrow A\ \boldsymbol{m}\ B, {\sim}B\ \boldsymbol{bm}\ C \rightarrow q,\ A\ \boldsymbol{o}\ D \wedge D\ \boldsymbol{o}\ C, {\sim}(p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D)\}$

is inconsistent.

Figure 2 shows the inference tree built for proving the inconsistency of the set E. We started by writing formulas from E with "+" signs as the initial branch of the inference tree. Then inference rules are applied step by step to the **BAL** sentences assigned to the vertices of the tree under construction.

So, at step 1, the rule 3 from Table 2 is applied to the sentence $+A\ \boldsymbol{o}\ D \wedge D\ \boldsymbol{o}\ C$. As a result of applying the rule, two sentences $+A\ \boldsymbol{o}\ D$and$+D\ \boldsymbol{o}\ C$ are obtained that are attached sequentially to the initial branch. At step 8, the rule 7 from Table 2 is applied to the sentence $+p \rightarrow A\ \boldsymbol{m}\ B$.As a result, two sentences $-p$and $+A\boldsymbol{m}B$are obtained, and the "fork" of these sentences is attached to the current branch of the tree.

Here we followed the standard tactics for choosing the sentence to which an inference rule should be applied and choosing the branches to which the resulting consequents should be attached (Fitting, 1996).

The sign "X" attached to the branch at step 9 signalized that it is closed in the sense that the sentences and inequalitiesfrom the branch form an inconsistent set (in this case, due the presence of $+p$ and $-p$). So, in the inference tree there are two brunches marked by the sign "X".

$+p \rightarrow A\ \boldsymbol{m}\ B$ [8]
$+{\sim}B\ \boldsymbol{bm}\ C \rightarrow q$ [11]
$+A\ \boldsymbol{o}\ D \wedge D\ \boldsymbol{o}\ C$ [1]
$+ {\sim}p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D$[2]
1: $+A\ \boldsymbol{o}\ D$ [6]
1: $+ D\ \boldsymbol{o}\ C$ [7]
2: $- {\sim}p \wedge {\sim}q \rightarrow B\ \boldsymbol{d}\ D$[3]
3: $+p \wedge {\sim}q$[4]
3: $- B\ \boldsymbol{d}\ D$[14]
4: $+ p$ (9)
4: $+ {\sim}q$ [5]
5: $- q$ (12)
6: $D^{-}-A^{-} \geq 1$
6: $A^{+}-D^{-} \geq 1$
6: $D^{+}-A^{+} \geq 1$
7: $C^{-}-D^{-} \geq 1$
7: $D^{+}-C^{-} \geq 1$
7: $C^{+}-D^{+} \geq 1$

8: $-p$(9)	8: $+A\boldsymbol{m}B$[10]
9:X	10: $A^{+} = B^{-}$

11 $-{\sim}B\boldsymbol{bm}C$[13]	11: $+q$(12)
13: $+B\boldsymbol{bm}C$[15]	12: X

14:$D^{-}-B^{-} \geq 0$	14: $B^{+}-D^{+} \geq 0$

15:$+B\boldsymbol{b}C$ [16]		15:$+B\boldsymbol{b}C$[18]	
16:$C^{-}-B^{+} \geq 1$		18:$C^{-}-B^{+} \geq 1$	15:$+B\boldsymbol{b}C$[19]
	15:$+B\boldsymbol{m}C$[17]		19:$C^{-}-B^{+} \geq 1$
	17: $B^{+-}= C^{-}$		

Figure 2: Inference tree for the set E.

Let us write out inequalities from other branches:

$E_1 = \{D^{-}-A^{-} \geq 1,\ A^{+}-D^{-} \geq 1, D^{+}-A^{+} \geq 1, C^{-}-D^{-} \geq 1, D^{+}-C^{-} \geq 1, C^{+}-D^{+} \geq 1, A^{+} = B^{-}, D^{-}-B^{-} \geq 0, C^{-}-B^{+} \geq 1\}$,

$E_2 = \{D^{-}-A^{-} \geq 1,\ A^{+}-D^{-} \geq 1, D^{+}-A^{+} \geq 1, C^{-}-D^{-} \geq 1, D^{+}-C^{-} \geq 1, C^{+}-D^{+} \geq 1, A^{+} = B^{-}, D^{-}-B^{-} \geq 0, B^{+-} = C^{-}\}$,

$E_3 = \{D^{-}-A^{-} \geq 1,\ A^{+}-D^{-} \geq 1, D^{+}-A^{+} \geq 1, C^{-}-D^{-} \geq 1, D^{+}-C^{-} \geq 1, C^{+}-D^{+} \geq 1, A^{+} = B^{-}, B^{+}-D^{+} \geq 0, C^{-}-B^{+} \geq 1\}$,

$E_4 = \{D^{-}-A^{-} \geq 1,\ A^{+}-D^{-} \geq 1, D^{+}-A^{+} \geq 1, C^{-}-D^{-} \geq 1, D^{+}-C^{-} \geq 1, C^{+}-D^{+} \geq 1, A^{+} = B^{-}, B^{+}-D^{+} \geq 0, B^{+-} = C^{-}\}$.

Let us add toevery $E_i$the standard inequalities $A^{+}-A^{-} \geq 1$, $B^{+}-B^{-} \geq 1$,$C^{+}-C^{-} \geq 1$,$D^{+}-D^{-} \geq 1$,and denote E_i* the resulting set.

It turns out that the sets E_i* are inconsistent. In fact, consider, for example, the set E_1*. It contains the inequalities $A^+ = B^-, D^- – B^- \geq 0$, $A^+ – D^- \geq 1$. From here we have

$B^- – A^+ \geq 0, D^- – B^- \geq 0, A^+ – D^- \geq 1.$

Adding up these inequalities, we get

$(B^- – A^+). + (D^- – B^-) + (A^+ – D^-) \geq 0 + 0 + 1,$

i.e. the contradictory $0 \geq 1$. Thus, the set E_1* is inconsistent.

Let us associate with any set S of inequalities of the form $X_i – X_j \geq r$ ($r \in \{0, 1\}$) the following graph $\Gamma(S)$:

- The set of $\Gamma(S)$ vertices makes up of X_i;
- The set of $\Gamma(S)$ edges with labels makes up of (X_i, X_j, r) such that $X_i – X_j \geq r$.

Figure 3 shows the graph $\Gamma(E_1$*).

It easy to prove that the set of inequalities S is inconsistent if and only if the graph $\Gamma(S)$ contains a positive cycle (i.e., the cycle having at least one edge with the label 1).

For example, the graph $\Gamma(E_1$*) in Figure 3 has the positive cycle

$(B^-, D^-, 0), (D^-, A^+, 1), (A^+, B^-, 0).$

Hence, the set E_1* is inconsistent.

Thus, we can apply an algorithm for detecting positive cycles in the graph $\Gamma(S)$ to recognize the inconsistency of the set S of inequalities.Hence, the set E_1* is inconsistent.

3.2 Metric Extension of Allen's Logic

This logic **MAL** is an extension of the logic **BAL** by inserting durations of temporal intervals and their fragments into **AL** sentences.

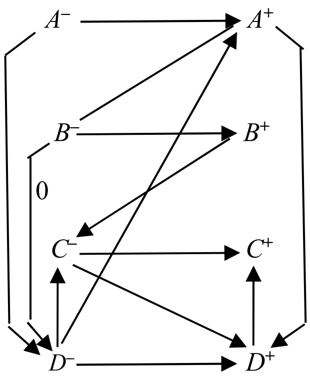


Figure 3: Graph $\Gamma(E_1$*).

The fragments of intervals entering sentences are denoted by I, J and K. Figure 4 shows how they are represented by the ends of temporal intervals in **AL** sentences. For example, for the sentence *A* ***b*** *B*we have $I = A^+ – A^-$, $J = B^- – A^+$ and $K = B^+ – B^-$.

A ***b****B* A^-========== A^+ B^-======B^+ |-------- *I* ---------|------ *J* ------|---- *K* ------| $I = A^+ – A^- J = B^- – A^+ K = B^+ – B^-$
A ***m****B* A^-========== A^+=B^-=====B^+ |---------- *I* --------|-------*K* -------| $I = A^+ – A^- K = B^+ – B^-$
A ***d*** *B* A^-======= A^+ B^-=========================B^+ |------ *I* ------|------ *J* -------|----- *K* ------| $I = A^- – B^- J = A^+ – A^- K = B^+ – A^+$
A ***s*** *B* A^-========== A^+ B^-=====================B^+ |-------- *J* -------|-------*K* --------| $J = A^+ – A^- K = B^+ – B^-$
A ***o*** *B* A^-=================== A^+ B^-=====================B^+ |------ *I* ------|--------- *J* -------|------- *K* -------| $I = B^- – A^- J = A^+ – B^- K = B^+ – A^-$
A ***f****B* A^-========== A^+ B^-=====================B^+ |--------- *I* --------|--------*J*-------| $I = A^- – B^- J = A^+ – A^-$

Figure 4: Intervals and their fragments for **AL** sentences.

sentences. For example, for the sentence *A* ***b*** *B*we have $I = A^+ – A^-$, $J = B^- – A^+$ and $K = B^+ – B^-$.

The expressions of the form

$I \geq r, J \geq r, K \geq r, I \leq r, J \leq r, J \leq r, K \leq r$

where r is an integer, are called α-*estimates*. Also, α-*terms*are conjunctions of α-estimates where semicolons are used as conjunction signs. For example, the expression

$I \leq 2; I \geq 5; J \leq –1; K \leq 4; K \geq 2$

an α-term.Another type estimates is β-*estimates*:

$X – Y \geq r$, $X – Y \geq r$ ($X, Y \in \{A^+, B^+, A^-, B^-\}$, $X \neq Y$). Also, β-*terms* are conjunctions β-estimates.

Sentences of the logic **MAL** are obtained by inserting α-estimates and α into **BAL** sentences For example, from the **BAL** sentence

A ***bs*** *B* ∧*B* ***d*** *C* → *A* ***fo****C*

we can obtain the **MAL** sentence

A ***b***($J \geq 2$)***s*** *B* ∧*B* ***d*** *C*→*A* ***f***($I \geq 3$; $K \leq 5$)***o*** *C*.

The inference system for the logic **MAL** consists the rules entering the tables Table 1 – Table 4.

Table 4: Inference rules for the logic **MAL.**

No	Antecedent	Consequents
1	$+A\ \theta(\tau)B$	$+A\ \theta B$, $+\theta(\tau)$
2	$+A\ \theta^{-1}(\tau)B$	$+B\ \theta A$, $+\theta(\tau)$
3	$-A\ \theta(\tau)B$	$-A\ \theta B \| -\theta(\tau)$
4	$-A\ \theta^{-1}(\tau)B$	$-B\ \theta A \| -\theta(\tau)$
5	$+\boldsymbol{b}(\tau)$	$+\tau\{I:=A^+-A^-, J:=B^--A^+, K:=B^+-B^-\}$
7	$+\boldsymbol{m}(\tau)$	$+\tau\{I:=A^+-A^-, K:=B^+-B^-\}$
9	$+\boldsymbol{d}(\tau)$	$+\tau\{I:=A^--B^-, J:=A^+-A, K:=B^+-A^+\}$
11	$+\boldsymbol{s}(\tau)$	$\tau\{I:=A^+-A^-, K:=B^+-B^-\}$
9	$+\boldsymbol{o}(\tau)$	$\tau\{I:=A^--B^-, J:=A^+-B^-; K:=B^+-B^-\}$
11	$+\boldsymbol{f}(\tau)$	$+\tau\{I:=A^--B^-, J:=A^+-A^-\}$
12	$+\sigma;\tau$	$+\sigma$, $+\tau$
13	$-\sigma;\tau$	$-\sigma \mid -\tau$
Plus the rules which the same as the rules 5 – 11 but with the replacement the signs '+' with '–'..		

Consider an example of an inference in the logic**MAL**.

Example 4. Let there are three works W*a*,W*b* and W*c* with temporal intervals *A*, *B*, *C*, the lengths of which are 4, 8, and 5, respectively. In addition, there are conditions *p* and *q* for which the following statements are true:

(1) if *p* is true, then W*a* is performed during W*b*, and W*a*finishes 2-4 time units before the end of W*b*;

(2) if*q*is true, then action W*c*finishes with action W*b*.

Put the question: "Does action *a*overlap in time with action *c*, under the assumption that both conditions *p* and *q* are satisfied? If so, then find the best estimate for the overlap time."

This knowledge can be represent in the language **MAL** an ontology:

$\boldsymbol{O} = \{|A| = 4, |B| = 8, |C| = 5,$

$p \rightarrow A\boldsymbol{d}(2 \le K \le 4)\ B,\ q \rightarrow C\boldsymbol{f}B\}.$

The question can be written as the query to the knowledge base ***O***:

Q: ? max *x*, min *y*: $p \wedge q \rightarrow A\ \boldsymbol{o}(x \le J \le y)C$.

Fig.5 shows the inference tree for the set of signed sentences $+Kb \cup \{-p \wedge q \rightarrow A\ \boldsymbol{o}(x \le J \le y)\ C\}$. The fourth branch of the graph contains the following inequalities (β-estimates):

$A^+-A^- \ge 4$, $A^--A^+ \ge -4$, $B^+-B^- \ge 8$, $B^--B^+ \ge -8$, $C^+-C^- \ge 5$, $C^--C^+ \ge -5$, $A^--B^- \ge 1$, $B^+-A^+ \ge 2$,

$+|A| = 4$ [1]
$+|B| = 8$ [2]
$+|C| = 5$ [3]
$+p \rightarrow A\ \boldsymbol{d}(2 \le K \le 4)\ B$ [6]
$+q \rightarrow C\,\boldsymbol{f}\,B$ [11]
$-p \wedge q \rightarrow A\ \boldsymbol{o}(x \le J \le y)\ C$ [4]
1: $A^+-A^- \ge 4$
1: $A^--A^+ \ge -4$
2: $B^+-B^- \ge 8$
2: $B^--B^+ \ge -8$
3: $C^+-C^- \ge 5$
3: $C^--C^+ \ge -5$
4: $+p \wedge q$ [5]
4: $-A\ \boldsymbol{o}(x \le J \le y)\ C$ [14]
5: $+p$ [7]
5: $+q$ [12]

6: $-p$ [7] | 6: $+A\ \boldsymbol{d}(2 \le K \le 4)\ B$ [8]
7: X | 8: $+A\ \boldsymbol{d}B$ [9]
8: $2 \le K \le 4$ [10]
9: $A^--B^- \ge 1$
9: $B^+-A^+ \ge 1$
10: $B^+-A^+ \ge 2$
10: $A^+-B^+ \ge -3$

11: $-q$ [12] | 11: $+C\boldsymbol{f}B$ [13]
12: X | 13: $C^--B^- \ge 1$
13: $C^+=B^+$

14: $-A\boldsymbol{o}C$ | 14: $-x \le J \le y$

$A^--C^- \ge 1$ | $C^--A^+ \ge 1$ | $A^+-C^+ \ge 1$ | $C^--A^+ \ge 1-x$ | $A^--C^+ \ge y-1$

Figure 5: Inference tree for knowledge base in Example 4.

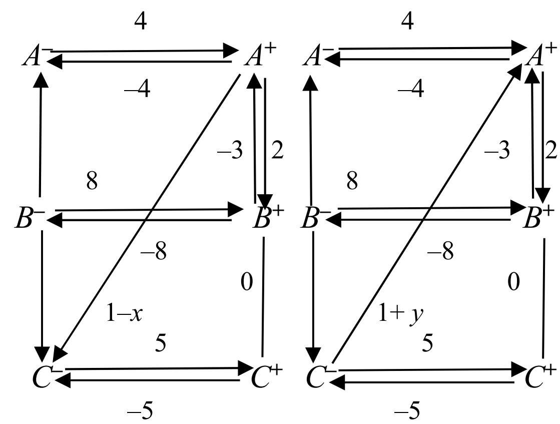


Figure 6: Graphs for the fourth and the fifth branches of the inference tree.

Fig.6 shows the graph constructed from this inequalities. It easy to see that the graph contain cycle

A^+, C^-,C^+, A^+, which corresponds to the inequalities

$C^- - A^+ \geq 1 - x$, $C^+ - C^- \geq 5$, $B^- - C^+ \geq 0$, $A^+ - B^- \geq -3$.

Adding up these inequalities, we obtain the inequality $0 \geq 1 - x + 5 + 0 - 3$, i.e., $0 \geq 3 - x$. Therefore, this inequality is contradictory if and only if $x \leq 2$. Thus, 2 is the maximum of x when the fourth branch is closed.

Similarly, in the fifth graph there is the cycle A^+, B^+, C^+, C^-,A^+ with the corresponding inequalities

$B^+ - A^+ \geq 2$, $C^+ - B^+ \geq 0$, $C^+ - C^- \geq -5$, $A^+ - C^- \geq y - 1$.

Adding up these inequalities, we obtain the inequality $0 \geq 2 + 0 - 5 + 1 + y - 1$, i.e., $0 \geq y - 3$. Therefore, this inequality is contradictory if and only if $y \geq 4$. Thus,4 is the minimum of y when the fifth branch is closed.

It is easy to verify that the first 3 branches are closed. Thus, $x = 2$, $y = 4$ is the answer to the query Q addressed the knowledge base ***O***.

4 CONCLUSION

We examined the possibility of using the languagesof the system “Binary Model of Knowledge” for describing domain workflow ontologies. The languages have users-friendly syntax and semantics which is based on formal concepts. It is important for workflows to model temporal properties. In “Binary Model of Knowledge”, there is the language **LTS** of temporal specification. We have introduced the logic that extends Allen’s interval logic by inserting durations of temporal intervals and their fragments.

ACKNOWLEDGEMENTS

This work was supported by Russian Foundation for Basic Research (projects 20-07-00615, 18-29-03088, 20-57-00015 and 20-07-00770).

REFERENCES

Aalst, W.H.P van der, Hee K.M. van, 2002. *Workflow Management: Models, Methods and Systems,* MIT Press, Cambridge.

Agostino, M., Gabbay, D.,Hahnle, R., Possega, 2002. *Handbook of tableaux methods*, Springer (2001).

Allen, J.F., 1983. Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26(11).

Bettini, X. Wang, and S. Jajodia. Temporal Reasoning in Workflow Systems. Distributed and Parallel Databases, 11(3):269–306, 2002.

Davulcu, H., M. Kifer, M., 1998, Ramakrishnan, C., Ramakrishnan, I. Logic based modelling and analysis of workflows. In ACM International Symposium on Principles of Database Systems (PODS),

Fitting, M., 1996.*First-order logic and automated theorem proving*, Springer.

Gruber, T.R., 1993. Translation approach to portable ontology specifications. Knowledge Acquisition, 5(2).

Mukherjee, S., Davulcu, H., Kifer, M., Senkul, S., Yang, G., 2003. Logic based approaches to workflow modeling and verification. In J. Chomicki, R. van der Meyden, and G. Saake, editors, Logics for Emerging Applications of Databases. Springer Verlag.

Plesniewicz, G. S., Karabekov, B. S., 2014. Ontologies in the “Binary Model of Knowledge”. ProgrammnyjeProduktyiSistemy, 1 (105).