Applying time-dependent algebraic systems for describing situations

Peeter Lorents

Professor. Cooperative Cyber Defence Centre of Excellence. Research and Development Branch Chief.

Erika Matsak

Associated Professor. Tallinn University. Institute of Informatics.

Abstract. Situation management requires means for describing various things, both situation states and developments. For this reason we turn to algebraic systems. We define time as a system, as well as systems that are time-dependent. We consider the application of the DST dialogue system as one possible way to reliably represent descriptions in the form of formulas.

Keywords. Situation management. Algebraic systems. Time. Time-dependent systems. Formulas. Dialogue system DST for transformation of texts from natural language to formulas.

Introduction

The prerequisites for successful situation management are correct descriptions of situations. This immediately raises two problems: (1) what could be the suitable solution for assembling and representing these descriptions and (2) how can we be sure that the descriptions are correct?

Our aim is to consider the algebraic systems based way to describe things. Specifically, we consider algebraic systems with so-called multiple main sets. These systems are a suitable means for describing many such situations that include elements belonging to certain sets, the relevant properties of these elements and the relationships between the elements - both “inside”, as well as “between“ sets.

We also handle time as a system by defining (several possible) times as algebraic systems of a specific type. Based on this notion of time(s) we define time-dependent systems, which are suitable means for describing the situations (for example, a natural disaster, a battle, national economic effects, etc.) that a system may encounter at some point in time. We describe various principles that “frame” the time-dependence of systems – for example, the principle of multiple courses of development, the principle of no predestination, or the principle of the correlation between developments and deduction.

To correctly describe the situation states and possible situation developments of systems, we study the formulas of systems and how the truth values are assigned to them. In order to extract formulas from natural language texts we can use E. Matsak’s DST dialogue system, which transforms arguments found in natural language texts into formulas.

The main part of this work is divided into three parts: (1) algebraic systems related to situation states, (2) time as a system and time-dependent systems related to development of systems, and (3) the options for using the dialogue system to represent arguments in situation descriptions with formulas that have a truth value.

1. Situation states and algebraic systems

One of the primary tasks in crisis management is to create an overview of the situation. For example, in order to neutralize a terrorist cell at the attack site, we need to identify which people present are members of the cell, what type of people they are, how are they connected to each other, what type of weapons, explosives, detonation devices and communication solutions they possess, etc. This example list included five sets: the perpetrators of the terrorist act (Ter), the weapons (Wea), the explosives (Exp), the detonating devices (Det) and the communication solutions (Com) at their disposal. In addition to the sets it is usually important to know the properties of the elements present in these sets. For example, whether the terrorists are fanatics (Fan) who cannot be reasoned with, or if they are people that one can talk to (Tal); which of the weapons pose a more serious threat (Ser^{+}), or a less serious threat (Ser^{-}); if the communication devices are citizens' band radios (CB), a bit more "complex" solutions (PRM) or professional devices (VHF); if detonation of explosives is triggered by radio (Rad), cable (Cab) or timer (Tim). It is also important to know the relations in and between sets, such as the subordination (Sub) and coordination (Coo) relations between terrorists; who has access to which arms (Arm); who deals with explosive devices (Fir). Obviously, this list is not exhaustive.

We see that there are two collections of interest in this situation state:

(1) collection H, which includes some *sets* with the notations H_1, H_2, … , H_m and

(2) collection S, which includes *predicates* with the notations P_1, P_2, … , P_n or the properties of the elements or the relations between the elements (including, potentially, both relations in and between sets).

Definition 1. An ordered pair, represented as $\langle H;S \rangle$, where the first component is the collection of sets H and the second component is the collection of predicates S, is called and *algebraic system* (see and compare, for example, Grätzer 2008, Lorents 2006, Maltsev 1970, Cohn 1965). The collection of sets H is often referred to as the *collection of main sets of a system*, or a bit more loosely, as the *systems' main sets*, and the collection of predicates S is called the *systems' signature*.

Note 1. An *ordered pair* is a two-element set with so-called position sensitive elements. In addition to ordered pairs, we may also need ordered triplets, ordered quadruplets etc. Such sets with position sensitive elements (or elements, where we can refer to the first, second, third, etc. element) are called *corteges* in set theory and algebra. The following "standard technology" can be used to define them:

- a 2-member cortege consisting of sets h_1 and h_2 is the set $\{\{h_1\},\{h_1,h_2\}\}$, which we refer to with the notation $\langle h_1,h_2\rangle$ or $\langle h_1;h_2\rangle$
- a k+1 member cortege consisting of sets $h_1, \dots, h_k, h_{k+1}$ is the set $\langle\langle h_1, \dots, h_k\rangle, h_{k+1}\rangle$, which we refer to with the notation $\langle h_1, \dots, h_k, h_{k+1}\rangle$.

Note 2. In case there are several sets in the collection of algebraic system sets, we talk about an *algebraic system with multiple main sets*. We can also encounter a situation, where we only have *one main set*, for example K. Indeed, if we wish to use only one main set K instead of many main sets and (NB!) preserve all the elements of all the sets $H_1, H_2, \dots, H_m$, then we use the union:

$K= H_1 \cup H_2 \cup \dots \cup H_m$

Note 3. If the signature does not contain functional relations, then we could be talking about a *model*. If, however, all the predicates in the signature turn out to be functional relations, then we could be talking about *algebra* (see, for example, Maltsev 1970).

The example above (the situation state of a terrorist attack) was neither a model or algebra, but a general case of an algebraic system, which can be represented as:

⟨{Ter, Wea, Exp, Det, Com}**;** {Fan, Tal, Ser+, Ser-, CB, PMR, VHF, Rad, Cab, Tim, Sub, Coo, Arm, Fir}⟩ or, if there is ***really*** no possibility for error, a simpler representation:

⟨Ter, Wea, Exp, Det, Com**;** Fan, Tal, Ser+, Ser-, CB, PMR, VHF, Rad, Cab, Tim, Sub, Coo, Arm, Fir⟩

The main sets in this system are Ter, Wea, Exp, Det, Com and the signature is Fan, Tal, Ser+, Ser-, CB, PMR, VHF, Rad, Cab, Tim, Sub, Coo, Arm, Fir.

Note 4. The presence of a property in the framework of algebra and set theory is represented by belonging to a certain subset. If the elements of a set X have the property P, then accordingly, $P \subseteq X$. If some element $y \in X$ has the property P, then we can write P(y) or $y \in P$. Thus, *property is equivalent to a subset* (the property P of the elements of a set X is nothing more than a subset of set X).

Example. Fan⊆Ter (some terrorists are fanatics, or in a special case, all the terrorists may be fanatics)

Note 5. A relationship is represented by belonging to a certain subset of a Cartesian product. If some elements in the sets $X_1, X_2, \dots, X_k$ have a relationship R, then accordingly $R \subseteq X_1 \times X_2 \times \dots \times X_k$. If it turns out that some elements $y_1 \in X_1, y_2 \in X_2, \dots, y_k \in X_k$ have the relationship R, then we can write $R(y_1, y_2, \dots, y_k)$ or $\langle y_1, y_2, \dots, y_k\rangle \in R$. We see that the *relationship is considered equivalent to the subset of the Cartesian product* (the relationship between elements of sets $X_1, X_2, \dots, X_k$ is some subset of the Cartesian product $X_1 \times X_2 \times \dots \times X_k$). It should be noted, however, that it is ***not required*** that the sets $X_1, X_2, \dots, X_k$ are all different from each other.

Examples.

Sub⊆Ter×Ter (in this situation, some terrorists are subordinates of other terrorists and must therefore take orders from them)

Arm⊆Ter×Wea (some terrorists are armed with a weapon of a certain type)

Definition 2. Unary, or *single member predicates* are properties. *Multi-member predicates* are relations. Depending on the number of elements that can have a certain relationship, we can speak of *binary* (two elements), *ternary* (three elements), … , *k-member relations*. The notations (symbols) of predicates are called the *predicate symbols* of the corresponding system.

Note 1. The predicate (property or relation) must not be confused with the notation of this predicate or the procedure that identifies objects belonging to a predicate.

Example. Let us consider the set of "points" on the sides of a common dice {1,2,3,4,5,6} and a binary relation between its elements with the *notation* Div (divisibility). The actual *relation* is the following collection of ordered pairs {⟨1,1⟩,⟨2,1⟩,⟨2,2⟩,⟨3,1⟩,⟨3,3⟩,⟨4,1⟩,⟨4,2⟩,⟨4,4⟩,⟨5,1⟩,⟨5,5⟩,⟨6,1⟩,⟨6,2⟩,⟨6,3⟩,⟨6,6⟩}. The *procedure for verifying the divisibility of* two numbers, however, consists of division and then checking the remainder.

Note 2. The notation must not be confused with the letter symbol! A notation **can be** a so-called single symbol, including a single letter. However, the notation can also be some acronym, word, phrase, geometric shape, sound, color, movement, change, etc.

Examples. According to the above, Fan (or "is fanatic") is a unary predicate symbol, which consists of three letters. Arm (or "wields a weapon of type…") is a binary predicate symbol, even though it also consists of three letters. A single symbol notation + represents a ternary relation, which is formed by the following triplets {…,⟨1,1,2⟩,⟨1,2,3⟩,…,⟨2,1,3⟩,⟨2,2,4⟩,…,⟨3,1,4⟩,…}.

Let us consider a system M=⟨H;S⟩, where the main sets are sets with the notations H_1, H_2, … , H_m and the signature consists of predicates P_1, P_2, … , P_n, representing properties or relations of the elements of the main sets. We now define the systems' atomic formulas, (non-atomic) formulas and the methods for finding the truth values of these formulas.

Definition 3. Let us define the atomic formulas of the given system M:

- if y is the notation of the element from one of the main sets of M and P is the notation of a unary predicate from that systems' signature, then P(y) is an *atomic formula* of system M
- if y_1, y_2, … , y_k are the notations of the elements from the main sets of M and R is the notation of a k-member predicate from the systems' signature , then R(y_1, y_2, … , y_k) is the atomic formula of system M.

Example. Formulas Ter(Jim), Ter(Joe), Sub(Joe, Jim), Wep(AK47), Ser^+(AK47), Arm(Joe, AK47) are the systems'

⟨Ter, Wea, Exp, Det, Com**;** Fan, Tal, Ser+, Ser-, CB, PMR, VHF, Rad, Cab, Tim, Sub, Coo, Arm, Fir⟩

atomic formulas. The first two - Ter(Jim), Ter(Joe) – represent the fact that *Jim and Joe are terrorists*. The third, Sub(Joe, Jim), means that *Joe is a subordinate of Jim*. The fourth, fifth and sixth formula - Wep(AK47), Ser^{+}(AK47), Arm(Joe, AK47) – represent the facts that *AK47 is a weapon*, a *serious weapon at that* and that *Joe currently has that weapon*.

Definition 4. *The truth values of system* M=⟨H;S⟩, where H={H_1, … , H_m} and S={P_1, … , P_n} *in the classical case*:

- an atomic formula P(y) *is true*, if the element with the notation y is from such a subset of some main set of the system M, that has the notation P, while this subset is a unary predicate of system M (meaning, $P\in S$).
- an atomic formula R(y_1, y_2, … , y_k) *is true*, if
  (1) the symbol R is annotation for the subset of a Cartesian product of sets with the notations H_{j1}, H_{j2}, … , H_{jk} (or shorter, $R\subseteq H_{j1}\times H_{j2}\times \ldots \times H_{jk}$), while the sets in question are from the collection of main sets of the system M (therefore, $1\leq j_1\leq m$, $1\leq j_2\leq m$, … , $1\leq j_k\leq m$) and the subset of the Cartesian product, with the notation R, is some k-member predicate of system M (meaning, $R\in S$)
  (2) in sets with notations H_{j1}, H_{j2}, … , H_{jk} there are such elements, with notations y_1, y_2, … , y_k, where a cortege with the notation $\langle y_1, y_2, \ldots, y_k\rangle$ is an element of exactly that subset of the Cartesian product $H_{j1}\times H_{j2}\times\ldots\times H_{jk}$, which has the notation R (or $y_1\in H_{j1}$, $y_2\in H_{j2}$, …., $y_k\in H_{jk}$, $\langle y_1, y_2, \ldots, y_k\rangle\in R$, $R\subseteq H_{j1}\times H_{j2}\times \ldots \times H_{jk}$, $R\in S$).

In other cases we say that the aforementioned atomic formulas are *false* in the classical case.

Note. The fact that there exists a relation Q between elements of the main sets H_{j1}, H_{j2}, … , H_{jk} of the system M=⟨H;S⟩ does not automatically infer that Q(y_1, y_2, … , y_k) is an atomic formula of the system M! Only if $Q\in S$ is Q(y_1, y_2, … , y_k) an atomic formula of the system M.

Example. Let us consider the system W, which only has one main set. The elements of that set are the integers -1, 0 and 1. This systems' signature consists of a unary predicate with the notation Pos, binary predicates with the notations $\neq$ and $\leq$, and ternary predicates with the notations + and $\cdot$. Therefore, we can write

$T=\langle\{-1,0,1\};\ Pos, \neq, \leq, +, \cdot\rangle$, where

Pos={1}

$\neq = \{\langle -1,0\rangle, \langle -1,1\rangle, \langle 0,-1\rangle, \langle 0,1\rangle, \langle 1,-1\rangle, \langle 1,0\rangle\}$, $\leq = \{\langle -1,-1\rangle, \langle -1,0\rangle, \langle -1,1\rangle, \langle 0,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle\}$

$+ = \{\langle -1,0,-1\rangle, \langle -1,1,0\rangle, \langle 0,-1,-1\rangle, \langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,-1,0\rangle, \langle 1,0,1\rangle\}$

$\cdot = \{\langle -1,-1,1\rangle, \langle -1,0,0\rangle, \langle -1,1,-1\rangle, \langle 0,-1,0\rangle, \langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,-1,-1\rangle, \langle 1,0,0\rangle, \langle 1,1,1\rangle\}$

The atomic formula Pos(1) is true in the given system, but Pos(0) is false.

The atomic formula $\leq(0,0)$ is true in the given system, but $\leq(1,-1)$ is false.

The atomic formula $\neq(0,-1)$ is true in the given system, but $\neq(-1,-1)$) is false.

The atomic formula $+(-1,1,0)$ is true in the given system, but $+(1,1,1)$) is false.

The atomic formula $\cdot(1,-1,-1)$ is true in the given system, but $\cdot(0,-1,1)$) is false.

The atomic formulas $\geq(1,-1)$ and exp(-1,-1,-1), however, are not formulas in the given system, because the relations $\geq$ or exp were not part of the systems' signature.

Definition 5. *The formulas of the system* are all the atomic formulas of the system, as well as all the formulas that can be constructed with the help of logic operations and quantifiers ($\neg$, $\vee$, &, $\supset$, $\Leftrightarrow$, $\forall$, $\exists$) from already existing formulas in the system.

Example 1. The formula $\neg(\forall y)[(\neg Pos(y))\supset(\exists z)(Pos(z)\&\cdot(y,y,z))]$ is a correct formula of the system W. (The natural language explanation of this formula is: it is not true that multiplying every non-positive element with itself gives a positive result. This is true, since 0 is not positive and multiplying it with itself does not yield a positive result.)

Example 2. Let us consider the formula Fan(z)$\supset\neg$Tal(z), or in other words: *if the terrorist z is a fanatic, then he cannot be reasoned with by talking.* This formula is a correct formula for the system

⟨Ter, Wea, Exp, Det, Com**;** Fan, Tal, Ser+, Ser-, CB, PMR, VHF, Rad, Cab, Tim, Sub, Coo, Arm, Fir⟩ **.**

The computation of the truth values of the systems' formulas follows the rules of first order predicate calculus. For this **we can use the truth value determination process of atomic formulas**, as described in Definition 4.

The correct formulas of the system give the "correct picture" of the situation of the system. If we base the description of the system on (correct) formulas, then we can use formal methods to verify the conclusions made about the system, including formal methods realized in computer applications. This is important in cases where the amount of data and the available time are overwhelming the human capabilities.

2. System developments and time-dependent algebraic systems

In order to predict the behavior of a system it is very important to know the state of the system. Unfortunately, we live in a world where the ***time-dependence of real (existing) systems is unavoidable***. We can refer to this as the *principle of unavoidable development* (see Lorents 2006, Ch 1, §5.3.). We use the notion of a time-dependent algebraic system to explain the time-dependence of systems or the possible developments of a system.

Definition 6. Let us consider the system Time that has two main sets: T and D. We call the elements of set T the *(time) moments of time T*, the elements in set D the *(time) interval lengths of time T*. The systems' signature contains binary relations Bef, Aft, Sim (where $Bef \subseteq T \times T$, $Aft \subseteq T \times T$, $Sim \subseteq T \times T$) for comparing time moments and a ternary relation Dur (where $Dur \subseteq T \times T \times D$) to evaluate the length of the time interval between time moments. Therefore, we can write Time=⟨T, D; Bef, Aft, Sim, Dur⟩.

Note. Depending on which sets T, D and predicates Bef, Aft, Sim, Dur we choose, we can get different things. Therefore, we are not dealing with a single, unique and "absolute time", but a collection of *different times*, where we may need to find ways of crossing over from one time to another.

Example. A suitable time for the observations and experiments conducted in school physics lessons is $\langle Q , Q^{0+} ; <, >, =, |\,| \rangle$, where Q is the set of all rational numbers, Q^{0+} is the set of all non-negative rational numbers; <, > and = are so called regular comparison operators, and | | is the notation of the absolute value of the difference between two numbers.

Next we fix some time Time=⟨T,D; Bef, Aft, Sim, Dur⟩ and classes CSets and CPred. The elements of class CSets are sets of sets. The elements of class CPred are sets, where the elements are predicates (recall that unary predicates are properties or subsets of some sets, and k-member predicates (where $k > 1$) are k-member relations or subsets of the Cartesian product of some sets).

Then we fix a binary relation with the notation Set, which relates time moments (elements of set T) to collections of sets (meaning, $Set \subseteq T \times CSets$). Next we fix a binary relation Sig, which relates time moments to collections of predicates (meaning, $Sig \subseteq T \times CPred$).

Definition 7. An ordered triplet ⟨Time, Set, Sig⟩ is called a time-dependent system, if the following condition is met:

If at some time moment $t \in T$, with collection of sets $s \in CSets$ and collection of predicates $p \in CPred$

(1) Set(t,s) and
(2) Sig(t,p),

then, without exceptions, all predicates from collection p must be the properties of or relations between the elements of the sets in collection s.

Let us agree that *the main sets of the system* ⟨Time, Set, Sig⟩ *at time moment t can be sets* that are elements of collection s. Let us also agree that predicates from collection p *can be the predicates of system* ⟨Time, Set, Sig⟩ *at time moment t*. Finally, let us agree that the *possible state of system* ⟨Time, Set, Sig⟩ *at time t* is the ordered pair ⟨s;p⟩ or a system, where the collection of main sets is s (where Set(t,s)) and signature (or collection of predicates) is p (where Sig(t,p) and where all predicates from collection p must be the properties of or relations between the elements of the sets in collection s).The collection of all possible states of the system is called the *development space of the system in time Time*.

Theorem. With time Time=⟨T,D; Bef, Aft, Sim, Dur⟩, collection of sets class CSets and collection of predicates class CPred, there is at least one time-dependent system, if the Cartesian product T×CSets×CPred has at least one subset W, where each element τ satisfies the following condition:

If $\tau=\langle t,s,p\rangle$ (where $\tau\in W$, $t\in T, s\in CSets$, $p\in CPreds$), then

$$(\forall\beta)[(\beta\in p)\supset[(\exists\alpha\in s)(\beta\in 2^{\alpha})\vee(\exists\alpha_1\in s)(\exists\alpha_2\in s)(\beta\in 2^{\alpha1\times\alpha2})\vee(\exists\alpha_1\in s)(\exists\alpha_2\in s)(\exists\alpha_3\in s)(\beta\in 2^{\alpha1\times\alpha2\times\alpha3})\vee\ldots]]$$

Proof. Let us take Set={⟨t,s⟩| ⟨t,s,p⟩∈W} and Sig={⟨t,p⟩| ⟨t,s,p⟩∈W}.

There are numerous examples of time-dependent systems. One good example is a family, which is initially formed by two people with the marriage relation. Some time interval later, when the children are born, the contents of the main set change. The signature also changes by adding the parent-child relation and possibly a sibling relation, etc.

Definition 8. The pair of relations ⟨Set, Sig⟩ is the *development shaper in time T* of the corresponding system dependent on the time Time, or the *fate of the system.* If the relations Set and Sig are functional relations, then we say that the development of system M in time Time is *predetermined* or *(fully) determined.* In other words, the fate of this system is (completely) set in time Time, or M(t)=⟨s(t);p(t)⟩. In the opposite case, we say that the system M is undetermined in time Time.

Conclusion. There exists a function f in case of time Time=⟨T,D; Bef, Aft, Sim, Dur⟩ and the development space of a determined system M=⟨Time, Set, Sig⟩, where the domain dom f is the set T of all time moments of the time Time and the value range rng f is the development space of the observed system. In essence, the system M is the abovementioned functional relation f (between the set T and the set CSets×CPred), which assigns exactly one possible state ⟨s(t);p(t)⟩ for each time moment t in time Time.

Note. Note, that it is ***not required*** that the relations Set and Sig in Definition 7 are functional relations. Therefore, if Set and Sig are not functional relations, then one cannot exclude the possibility that for some time moment t there exist multiple different states for the system M. This raises the requirement for a few more relevant principles.

The principle of multiple courses of development. Time-dependent systems may have several different possible states.

Example. One or more children may "join" the family next year.

The principle of being in a single state. Each time-dependent system is at each time moment t in exactly one of its possible states. This means that if some system M has multiple possible states for time moment t (for example, ⟨s,p⟩ and ⟨s',p'⟩), then it can be in exactly one of these states at time moment t (for example, if at time moment t, the system is in state ⟨s',p'⟩, then the system cannot be at the same time in state ⟨s,p⟩, as long as ⟨s,p⟩≠⟨s',p'⟩).

Example. A family can only have one concrete state at any time moment t. It is not possible to have only one child in the family, and at the same time have three children in the same family. The situation that options – one, two, three, … children - are possible does not mean that they can be true at the same time.

Note. Transitioning from one state to another there, the collections of main sets of the system may change in the "shallow" or "deep" level. In the "shallow" level, a main set or predicate can appear or disappear in the system. In the "deep" level, elements "in" existing main sets or predicates may appear or disappear.

Example. If a child is born into a family that had no children before, then a new set is added to the collection of main sets – the set of children. The signature also gets a new relation – the parent-child relation. In a few years, a second child is born. No new main sets are needed, just the children set gets a new element. However, the situation is a bit different in the signature: first, a new "shallow" change by adding a new relation – the sibling relation; at the same time, two new ordered pairs are added to the parent-child relation: ⟨father, second child⟩ and ⟨mother, second child⟩.

The principle of no predestination. The system arriving to one specific possible state at time moment t is not predestined. In other words, it happens with a certain probability.

Note. In principle it is possible to even require even stricter conditions. For example, we can require that at no time moment will there be a single state where the system transitions with the probability of 1 (this would eliminate determined systems). The total of the probabilities of all possible states at time moment t must equal 1, etc.

When describing time-dependent systems, we can use formulas and procedures for determining their truth values in a way analogous to the description in the previous part.

Example. Let us consider a time-dependent system M, which at the time moment t is in state

$\langle Ter_t, Wea_t, Exp_t, Det_t, Com_t; Fan_t, Tal_t, Ser+_t, Ser-_t, CB_t, PMR_t, VHF_t, Rad_t, Cab_t, Tim_t, Sub_t, Coo_t, Arm_t, Fir_t\rangle$.

The formula $(\exists t)(\forall t')[\ [(t\in T)\&(t'\in T)\&(Aft_{t'}(t',t)\&\neg Sub_{t'}(Joe, Jim)] \supset Tal_{t'}(Joe)]$ is a formula of system M, which represents that *if from some moment Joe is no longer a subordinate of Jim, then Joe is the one we can "talk" to (reason with).*

This, however, presents an interesting **problem**: how are the formulas representing the system M at (NB! comparable) times t and t' (for example, Bef(t,t')) related to the formulas representing the system M at time moment t'?

One possible answer is in the following principle, which includes the modalities *less probable* and *more probable.*

The principle of correlation between developments and deduction. If for two observed time moments t and t', the time moment t precedes the time moment t' and if it can be *proven* that the formulas describing M(t') follow logically from the formulas describing M(t), then it is *more probable* that the system M transitions from M(t) to M(t'). However, if it *cannot be proven* that the formulas describing

M(t') follow logically from the formulas describing M(t), then it is *less probable* that the system M transitions from M(t) to M(t').

Several different approaches have been developed and implemented for describing situations, in addition to the one demonstrated here. In many cases it is not difficult to see the road from a concrete approach to suitable (time-dependent) algebraic systems. Let us review, for example, the situation description distribution developed by Shijan (2007):

- Describing individual objects
- Describing the relations between objects
- Describing the distribution, structure etc. of the collection of objects
- Describing the states
- Describing the processes

The transition from these components to suitable algebraic systems could take place as follows:

- When describing individual objects we use formulas that are assembled based on unary predicates
- When describing the relations between objects we allow the "basic collection" of formulas to be extended with suitable k-member predicates (where k>1)
- When describing the structure of the collection of objects we rely on a suitable algebraic system
- When describing states we use the formulas from systems that are formed in the way demonstrated in the previous part
- When describing processes we use the formulas of time-dependent systems demonstrated in this part.

A relatively similar way allows us to get from so called fuzzy relations describing the temporal and spatial aspects of situations (see Jakobson 2010) to descriptions that are represented by the formulas of time-dependent systems.

This brings us to the next problem: the formula based descriptions of systems are reliable input for automated processing, but how can we extract them if the situations or processes are described in so called natural language texts? Today and in the foreseeable future, the descriptions of the state and development of various events are largely based on human observations, represented in natural language, or possibly in a bit more constrained "specialty" language. Therefore, we need means that allow us to translate natural language texts into the so called formula language.

3. The language aspects of describing situations

We now turn to the options that the DST dialogue system developed by E. Matsak (see Matsak 2005-2010) provides for finding the necessary formulas for describing situations. One of the foundations of this work is the text transformation procedure developed by P. Lorents (Lorents 2000; Lorents, Matsak 2010).

The DST system is capable of transforming texts based on morphologic-syntactical characteristics, where the natural language constructs represent *individual objects, predicates, logic operators and quantifiers, as well as various modalities* (Matsak 2005). Various morphologic analyzers have been developed to conduct the corresponding morphologic-syntactic analysis in many languages, including Estonian. Some of them are also available online (www.filosoft.ee).

The DST system also allows the transformation of texts that include so called *higher order predicates* (for example, properties of properties or relations, assessments of properties and relations, etc.) (Matsak 2006). Other means allow the transformation of *events taking place in time, activities* and other descriptions in natural language (Matsak 2006). Therefore, the necessary means exist for transitioning from natural language descriptions of system states or developments in time to formula based descriptions.

However, it should be noted that the DST system is *principally* a dialogue system, which will ask for human assistance if the need arises. Although, the more the DST system is used, the more it will "learn" and is able to act more independently. The source of this *"principal"* peculiarity is the requirement in the Lorents' procedure, which mandates that the transformed text conveys the same "idea" as the original text. This is particularly evident if missing parts need to be added during the transformation process. A somewhat lesser "problem" is the opposite act – removing so called unnecessary words and symbols. There are several possible reasons for this. For example, the Estonian language morphologic analyzer is able to identify 17 different types of words. Three of these types are unnecessary for identifying the logic roles of parts of the text: interjection (for example, "Hey"), adposition (for example, "through") and punctuation (for example, "?"). In the transformation process we can also exclude sentences that have no logic truth value and parts of text that represent direct or indirect speech (for example, *the lieutenant said that nobody will fire without orders;* and *"Nobody fires without orders," said the lieutenant*). The transformed text need not consist of simple sentences. It is enough if the borders between the clauses are automatically identifiable (see Müürsepp, Puolakainen 2007). Before the final step of the transformation process where the formula is assembled, we must ensure that there are no multiple predicates within the same clause. In order to get rid of unnecessary predicates, we must "teach" the DST system to perform the necessary replacement and relocation operations. For example, let us observe the sentence: *short white man carries a bag*, the structure of which is represented by:

adjective 1 adjective 2 noun ((subject) for designation) verb(s) noun (object).

In case of a sentence with this structure the system must be "taught" to transition to the next structure:

noun (subject) is adjective 1 & noun (subject) is adjective 2 & noun (subject) verb(s) noun (object),

which corresponds to the following sentence: *man is short and man is white and man carries a bag.*

This can easily be transformed to the formula: Sh(m)&Wh(m)&We(m,b).

In order to teach the DST system to perform such replacements, we must study and present the following regular expressions:

- Several adjectives in the same clause
- Verb (or several verbs as a single predicate) and one or more adjectives
- Using superlatives as values to assessments
- Joining multiple nouns into a single individual (for example, if one noun is a complement to another, then this situation can be identified syntactically)
- Categorizing nouns into predicates (Let us consider two texts in Estonian language: *Eriku arvuti* [Erik's computer] and *Erika arvuti* [Erika's computer]. In the latter case, according to a **peculiarity in the Estonian language**, we do not just have two nouns (*Erika* and *arvuti*), but one of these (*Erika*) must also be in the role of a predicate).

There is another important aspect to consider here. In situation management, in addition to reliable descriptions (that can be used by computers) such as formulas, we also need means that support the decision making process. In other words, we need hardware or software solutions that are capable of creating new correct formulas (that represent correctly the description of following situations) based on existing correct formulas (that represent correctly the current situation). Natural language text analysis is becoming one feasible approach for this task. Specifically, the analysis of texts that contain the steps that describe how reliable experts have arrived at one decision or another. The (improved version of) DST system can be used to solve such tasks (see Matsak, Lorents 2010; Lorents, Matsak 2010).

Summary

Humans live in an environment where success depends on having the correct overview of situations and making correct decisions based on it. Unfortunately, we also face situations where the amount of data or the need for very fast decision making outstrips human capability, for example, in the framework of cyber space (see for example, Ottis, Lorents 2010, Lorents, Ottis 2010). This raises the need for technological support means – artificial intelligence systems – that perform and produce results reliably. The best way to ensure reliability are sound mathematical foundations: a sound terminology, means for writing down arguments, and means for getting new correct arguments from existing correct arguments. Here we focused primarily on finding a suitable terminology and means for formalizing arguments. For this we used algebraic systems and the natural language text transformation dialogue system DST.

We would like to sincerely thank Gabriel Jakobson from Altusys Corp, who motivated us to research this issue, especially with a view towards situation management aspects, as well as Rain Ottis for his contribution to the preparation of this work.

References

Cohn P. M. (1965). *Universal Algebra.* Evanston: Harper&Row.

Grätzer G. (2008). *Universal Algebra*. Second Edition. Springer.

Jakobson G. 2010. *Incorporating time and spatial-temporal reasoning into situation management*. Proceedings of SPIE. Cyber Security, Situation Management, and Impact Assesment II; and Visual Analytics for Homeland Defence and Security II. 5 and 8-9 April 2010 orlando, Florida, united States. Volume 7709.

Lorents, P. (2000) *Keel ja loogika* (*Language and Logic*). EBS Print. Tallinn.

Lorents, P. (2006). *Süsteemide maailm* (The World of Systems). Tartu: Tartu Ülikooli Kirjastus.

Lorents P., Matsak E. (2010). *System mining inference rules from natural language texts*. The 3rd International Multi-Conference on Engeneering and Technological Innovation. International Institute of Informatics and Systemics, 2010, Orlando, Florida, USA, p. 309 - 314.

Lorents P., Ottis R. 2010. *Knowledge based frmework for cyber weapons and conflict*. Conference on Cyber Conflict. Proceedings. June 16th – 18. 2010. Tallinn, Estonia. CCD COE Publications. p. 129 – 142.

Maltsev, A. I. (Мальцев А. И.) (1970). *Алгебраические системы* (Algebraic systems). Moscow: Наука.

Matsak E. (2005). *Dialogue System for Extracting Logic Constructions in Natural Language Texts*. The 2005 International Conference on Artificial Intelligence, Las Vegas, USA. , 2005.

Matsak, Erika (2006). *System DST for Transforming Natural Language Texts, Representing Estimates and Higher Order Predicates and Functionals*. In: Proceedings of The 3rd International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2006: The 3rd International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2006. July 20-23, 2006. Orlando, Florida, USA, p. 79 - 84.

Matsak E. (2010). *Discovering Logical Constructs from Estonian Children Language*. Germany: Lambert Academic Publishing.

Matsak E., Lorents P. (2010). *Digital solutions for inference rules in decision-supporting systems*. Fifth International Conference on Digital Information Management. Thundar Bay, Canada: IEEE, 2010.

Müürsepp K., Puolakainen T. (2007). *Eesti keele formaalne grammatika: mudelist rakenduseni*. Aastaraamat. 2002–2003 / Eesti Matemaatika Selts. Tallinn. Lk. 16–29.

Ottis R., Lorents P. 2010. *Cyberspace: Definition and Implications.* Proceedings of the 5th International Conference on Information Warfare and Technology (ICIW 2010). Air Force Institute of Technology. 8 – 9. April 2010. Published by Academic Publishing Limited Reading. p. 267 – 270.

Shijan A. A. (2006, 2007) Шиян А. А. *Економічна кібернетика : вступ до моделювання соціальних і економічних систем* : навч. посіб. : гриф МОН України Львів : Магнолія 2006, 2007. – 228 с.