A model of fuzzy spatio-temporal knowledge representation and reasoning based on high-level Petri nets

Slobodan Ribaric, Tomislav Hrkac*

University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

ARTICLE INFO

Article history:
Received 11 May 2011
Accepted 20 September 2011
Recommended by: D. Shasha
Available online 17 November 2011

Keywords:
Knowledge representation
Fuzziness
Spatio-temporal reasoning
Multi-agent systems
High-level Petri nets

ABSTRACT

In many application areas there is a need to represent human-like knowledge related to spatio-temporal relations among multiple moving objects. This type of knowledge is usually imprecise, vague and fuzzy, while the reasoning about spatio-temporal relations is intuitive. In this paper we present a model of fuzzy spatio-temporal knowledge representation and reasoning based on high-level Petri nets. The model should be suitable for the design of a knowledge base for real-time, multi-agent-based intelligent systems that include expert or user human-like knowledge. The central part of the model is the knowledge representation scheme called FuSpaT, which supports the representation and reasoning for domains that include imprecise and fuzzy spatial, temporal and spatio-temporal relationships. The scheme is based on the high-level Petri nets called Petri nets with fuzzy spatio-temporal tokens (PeNeFuST). The FuSpaT scheme integrates the theory of the PeNeFuST and 117 spatio-temporal relations.

The reasoning in the proposed model is a spatio-temporal data-driven process based on the dynamical properties of the scheme, i.e., the execution of the Petri nets with fuzzy spatio-temporal tokens. An illustrative example of the spatio-temporal reasoning for two agents in a simplified robot-soccer scene is given.

1. Introduction

In the past 10 years, one of the central problems in intelligent system design has been the development of appropriate knowledge-representation schemes that support spatio-temporal representation and reasoning [1–5]. These schemes have been used in different application areas such as computer vision and robot navigation [6–14], multimedia [15–18], geographical information systems (GIS) [2,19–23], natural language processing and engineering design [24], etc.

Recently, in many application domains there is a need for the development of knowledge-representation schemes that support the human-like knowledge representation of spatial, temporal and spatio-temporal information and human ways of reasoning. Most human knowledge, however, is typically expressed in vague and imprecisely defined concepts and the inference is mostly supported by common-sense and intuitive reasoning. One of the approaches to enable the representation and handling of such a type of knowledge is to introduce the concept of fuzziness [25]. Although some successful formalisms have been proposed for the separate representation of fuzzy temporal [26] or fuzzy spatial [27] data, relatively little work has been done in the field of integrated fuzzy spatio-temporal knowledge representation and reasoning.

The motivation for our research was the development of a model that allows the human-like representation of spatial, temporal and spatio-temporal information and reasoning that is suitable for the knowledge-base design used in computer-vision and robot-navigation intelligent systems based on the concepts of a multi-agent system (MAS) [28–30].
The main goals that have to be achieved are as follows:

(i) The model has to support the design of a knowledge base that includes fuzzy and imprecise temporal, spatial, and spatio-temporal relations among moving agents and/or objects. The fuzziness and imprecision related to temporal, spatial and spatio-temporal relationships have to be expressed in a form that is appropriate for human experts and users.

(ii) The model has to be appropriate for multi-agent-based systems in such a way that it enables the independent design of a knowledge base for each of the agents, and the modeling of the interactions among them.

(iii) The model has to allow a hierarchical representation and modeling of the system at different abstraction levels.

(iv) The model should be based on a well-defined formalism that allows a formal analysis of different spatial, temporal and spatio-temporal relationships among the objects (according to Ferber [28], the agents are specific objects, representing the active entities in the system), by changing the initial conditions, temporal, spatial, spatio-temporal relationships or the final goals of a modeled system.

(v) The aim of the model is the development of a knowledge base for real-time applications, meaning that the model has to support an efficacious short-time-consuming reasoning process.

(vi) The proposed model has to be suitable for the design of a program simulator based on an object-oriented programming environment.

(vii) For small or moderate-sized modeled systems the model should offer a graphical representation of the knowledge base for each of the agents.

In this paper we propose a model, the main component of which is an original fuzzy spatio-temporal knowledge representation and reasoning scheme called FuSpaT, which fulfills the above-mentioned main goals. The proposed scheme is based on an original high-level Petri net, called the Petri net with fuzzy spatio-temporal tokens (PeNeFuST). The model also includes an object-oriented simulator that contains tools for the analysis of a modeled system.

The rest of the paper is organized as follows. The related work concerning the proposed scheme is introduced in Section 2. In Section 3, we briefly describe the theoretical basis of the high-level Petri net with fuzzy spatio-temporal tokens. Section 4 presents the fuzzy spatio-temporal knowledge representation scheme called FuSpaT. In Section 5, an example of using the FuSpaT for modeling the details of a robot-soccer scene is given. The conclusion and future work are discussed in Section 6.

2. Related work

Spatio-temporal formalisms have been previously discussed in the literature. The related works concerning the proposed spatio-temporal knowledge-representation scheme can be in general divided as follows:

(i) crisp-like spatio-temporal representation models:
- approaches that temporalize the models that are based on a spatial formalism [31,32], or vice versa [1];
- approaches based on a description of possible changes of positions and relative orientations of the objects [4,5,24,33,34];
- Petri net-based models [17,35–40];
- a hybrid approach that takes into account elements from temporal logic and elements from point-set theory and point-set topology [3];

(ii) fuzzy spatio-temporal models based on the following approaches:
- fuzzy set theory and fuzzy spatio-temporal relationships [41];
- linguistic descriptions of the moving objects [42];
- fuzzy-rule-based reasoning [43–45].

The brief descriptions of the related works follow. Bennett et al. [31] introduced a temporalization of the topological RCC8 calculus. Ragni and Woelfl [32] investigated a temporalization of the cardinal directions [46] in order to define a method for encoding temporized spatial constraint satisfaction problems as deterministic planning problems.

Hornsby and Egenhofer presented an approach to spatio-temporal knowledge representation based on a description of possible changes of real-world phenomena, called identifiable objects, modeled at a high level of abstraction [2]. The foundation of the model is a set of primitives and the operations that can be performed on them. These primitives are the identity states of objects and transitions. The term object refers to the representation of a real-world phenomenon in an information system. Identity states are associated with objects, capturing the notion that although an object’s identity is enduring, the state of the identity may change, e.g., from existing to non-existing. The objects and their associated identities are linked through another primitive, the transition. The progression of an object from one state of identity to another is modeled by the transitions. The authors proposed an iconic visual language called change description language (CDL) to describe the changes to the identity states of objects. Although no explicitly spatial information has been incorporated into this model of change, it has been shown that tracking the changes to an object’s identity over periods of existence and non-existence gives useful insights into the behavior of an object over time that are relevant to many cases of spatio-temporal change. The proposed model was used in GIS applications.

Erwig and Schneider [3] described a more explicit framework for the representation of spatio-temporal data by means of so-called spatio-temporal predicates. This framework is based on a hybrid approach that takes into account elements from temporal logic and elements from point-set theory and point-set topology. Presupposing a continuous model of time, they employ temporal functions
as the basis of an algebraic model for three basic spatio-
temporal data types: moving point, evolving line and evol-
ving region. Then, starting with eight basic topological
predicates of the so-called nine-intersection model
(which has been shown to be equivalent to the RCC-8
variant of the region connection calculus) and by using
the concepts of temporal lifting and temporal aggregation,
you define eight basic spatio-temporal predicates as
temporally lifted spatial (topological) predicates with a
certain preferred or default temporal aggregation. The
relationships between two spatio-temporal objects can
then be appropriately modeled by scenarios described as
sequences of spatial and basic spatio-temporal predicates.
The authors also define an algebra of spatio-temporal
predicates and several inference rules.

Wolter and Zakharyaschev [1] proposed a family of
decidable spatio-temporal formalisms, based on a combi-
nation of the region connection calculus (RCC) as a
topological model of space with several temporal formal-
isms, including a linear temporal model based on time
points, a branching model of time and Allen's temporal
interval logic [47].

All of the above-described works are based on a purely
topological view of space and do not take into account the
relative positions of the objects with respect to each other,
which can be a drawback when representing some com-
mon scenarios. In a series of papers [4,5,33,34], Van
de Weghe et al. proposed a qualitative trajectory calculus
(QTC) as a language for representation and reasoning
about the movements of point objects in a qualitative
framework, able to differentiate between groups of dis-
connected objects. Several variations of QTC were devel-
oped. The simpler QTC-basic (QTcB) considers only the
change in the distance between the objects, while the
more general QTC-double-cross (QTcC) also takes into ac-
count the relative orientation of the object movements
in two dimensions. The relative movement of the two
objects can then be represented by a four-component
label, where the first two components describe the
tendency of the change of the distance of an object with
respect to the current position of another object, while the
other two components describe the relative orientation of
the object movements with respect to the reference line
that connects them. Each of the components can assume
one of the three possible qualitative values (+, −, or 0),
resulting in \( 3^4 = 81 \) possible QTcC relationships. More
complex scenarios can be modeled by means of the so-
called conceptual animations, i.e., sequences of the basic
QTcC relationships associated with the corresponding
time points or time intervals. Two reasoning formalisms
are proposed: a formalism based on composition tables
and a formalism based on conceptual neighborhood
diagrams.

Several Petri net-based spatio-temporal representation
formalisms have also been proposed. A spatial and tem-
poral relationship Petri net (STRPN) [17] is a Petri net-based
knowledge representation scheme able to describe the
spatio-temporal relationships of moving multimedia
objects that may refer to each other for synchronization
and computing spatial display addresses. A STRPN is, in
fact, an extension of the object composition Petri net
(OCPN), which is itself an extension of the Petri net [39] to
specify the temporal relationships of multimedia objects.

To the temporal representation capabilities of an OCPN, a
STRPN adds the capability to describe the spatial relation-
ships between objects. The objects are represented by
their minimum bounding boxes, and their relative spatial
relationships are represented by 4-tuples, that can express
169 different spatial relationships. A STRPN extends the basic Petri net model with three different
types of places and three different types of transitions
(with different firing rules). Media places (MPs) hold the
playing information of the multimedia objects. Address
places (APs) buffer the spatial information of the refer-
ential objects. This information is forwarded to targeted
media places when the targeted media places have
tokens. Delay places (DPs) delay the play of the multi-
media objects to coordinate the temporal sequences of
the presentations. As in an OCPN, all the places in the
STRPN have playing durations within which the tokens
are locked. The tokens are then unlocked after the plays
are finished. Three types of transitions are distinguished
according to the firing rule. Unlocked then fire (ULF) transi-
tions fire only when the input places are unlocked. An
enabled then firing continuously (EFC) transition fires
when a new token arrives in its input place and keeps firing
until its input token ceases to exist. An enabled then firing
once (EFO) transition fires immediately when a new token
arrives but does not fire again until the next token arrives.

Although a STRPN can describe a set of spatio-temporal
relationships between objects, its expressiveness is still
limited.

A similar, but more complex Petri net-based scheme,
called the multimedia color time Petri net (MMCTPN) has
been proposed by Gomaa et al. [35]. This scheme supports
all the capabilities of the STRPN, but adds a user-interac-
tion modeling capability. It is based on color Petri nets
[36–38] and specifies four different types of places and
different types of transitions as well as two different
types of tokens.

Ribarić and Hrkač [40] proposed a crisp spatio-temporal
representation model based on the high-level Petri
net called Petri net with spatio-temporal tokens (PNSTT),
which is used as the main building block of a knowledge-
representation scheme called SpaTem. The SpaTem
scheme integrates the theory of the PNSTT and 117
spatio-temporal relationships.

Special efforts in knowledge-base development are
made to imitate human-like expert-knowledge represen-
tation and human ways of reasoning. Köprülü et al. [41]
proposed a model for representing and querying the
spatio-temporal properties of the objects in video data.
They introduced a set of so-called fuzzy spatio-temporal
relationships between objects. In reality, however, these
relationships are purely spatial relationships that become
non-exact because of the objects’ movement during a
certain time interval.

De Runz et al. [48] proposed the use of a fuzzy set
theory to represent imprecise multi-modal archaeological
data, such as the localization and orientation of antic
streets and the estimated time periods of their existence.
The described method, however, does not provide an
explicit definition of a set of spatio-temporal relationships and it does not support any mechanism of spatio-temporal inference.

Sjahputera et al. [42] described a system capable of linguistically describing an object in motion. The system tracks a single object moving in a straight path at a constant velocity and generates a so-called dynamic linguistic description that classifies the direction of the object’s movement into one of four possible categories that can be further modified by linguistic hedges, such as “mostly” or “a little”. Although the described approach uses linguistic expressions to model the spatio-temporal knowledge, it is not based on fuzzy set theory, because no numerical membership degrees are assigned to the data. In addition, this system does not provide any reasoning mechanisms.


3. Petri net with fuzzy spatio-temporal tokens

In this section we describe a new high-level model of the Petri net, called the Petri net with fuzzy spatio-temporal tokens (PeNeFuST). The PeNeFuST is based on a p-space-timed net model (which associates information about the time duration of an action or state and the corresponding change of an object’s spatial position to each place) and the concept of fuzzy spatio-temporal tokens. A fuzzy spatio-temporal token in the proposed model has a double role: it denotes the state of the modeled system and it carries the spatio-temporal information.

3.1. Formal definition of a Petri net with fuzzy spatio-temporal tokens

The Petri net with fuzzy spatio-temporal tokens (PeNeFuST) is a high-level Petri net defined as the following n-tuple:

\[ \text{PeNeFuST} = (P, \Sigma, T, I, O, \Psi, M, \Omega, \psi, \lambda_0, \delta_0, \kappa, \beta), \]

where \( P, \Sigma, T, I, O \) and \( \Omega \) are the components of a generalized Petri net (PN), defined as follows [39]: \( P \) is a finite set of places \( \{p_1, p_2, \ldots, p_n\} \), \( n \geq 0 \), \( T \) is a finite set of transitions \( \{t_1, t_2, \ldots, t_q\} \), \( q \geq 0 \), \( I \) is an input function \( I: T \rightarrow \mathbb{P}(O) \), a mapping from transitions into bags of places, and \( O \) is an output function, \( O: \Sigma \rightarrow \mathbb{P}(P) \), a mapping from transitions into bags of places, \( P \cap T = \emptyset \).

A function \( \Psi: P \rightarrow \Sigma \times T \) associates the fuzzy spatial, temporal or spatio-temporal information to the each place. \( \Psi \) is a mapping from a set of places to a set of either fuzzy \( d \)-dimensional spatial \( (\Sigma = (\Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_d)) \), fuzzy temporal \( (T) \) or fuzzy spatio-temporal \( (\Sigma \times T) \) values, where each component of the value is a fuzzy number. In general, the above components of fuzzy values can be represented by any convex and normal fuzzy subset of \( R \) [49]. In our model the triangular and/or trapezoidal fuzzy numbers are used. In many applications, the function \( \Psi \) can be reduced to the form \( \Psi: P \rightarrow \Sigma \times T \).

A set \( M = \{m^0_1, m^0_2, \ldots, m^0_n, \ldots, m^0_l, \ldots, m^0_m\} \), where \( 1 \leq r < \infty \), is a set of fuzzy spatio-temporal (S-T) tokens. A fuzzy S-T token, like a token in the colored Petri nets [36] has individuality, i.e., it carries inherent information about the visited places and the corresponding spatial changes and the temporal durations of the corresponding activities. A fuzzy S-T token \( m^0_i \) is the successor of the fuzzy S-T token \( m^0_i \), \( k = 0, 1, 2, \ldots \), meaning that \( m^0_i \) is generated in the output place of a fired transition, after \( m^0_i \) is removed from its input place.

An injective function \( \Omega: P \rightarrow \varphi(M) \) is called the marking of the PeNeFuST. \( \varphi(M) \) denotes the power set of \( M \). With \( \Omega_0 \) we denote the initial marking, i.e., the initial distribution of fuzzy S-T tokens at the places of the PeNeFuST.

The function \( \psi \) is a mapping called a spatio-temporal token, and it is defined as follows: \( \psi: M \rightarrow \mathbb{S}_\leq 1 \{p_i, \psi(p_i)\} \), where \& denotes concatenation and \( k \) is a number of the visiting places for a token \( m^0_i \in M \). It assigns a history of visited places and the corresponding spatial changes and the temporal durations of the corresponding activities to the fuzzy spatio-temporal tokens.

\( \lambda_0: M \rightarrow \mathbb{R}^d \) is a mapping that associates the initial fuzzy spatial position of the object located in a \( d \)-dimensional world to the fuzzy S-T token, where \( \mathbb{R}^d \) denotes a set of fuzzy numbers defined in \( \mathbb{R}^d \).

\( \delta_0: M \rightarrow \mathbb{R} \), where \( \mathbb{R} \) denotes a set of fuzzy numbers defined in \( \mathbb{R} \), is a mapping from a set of fuzzy S-T tokens to a set of fuzzy numbers, and it specifies the time for which the activity of the object is postponed.

\( \kappa: M \rightarrow [0, 1] \) is a mapping that associates a degree of confidence about the information carried by each fuzzy spatio-temporal token.

The function \( \theta: T \rightarrow [0, 1] \) is a mapping that assigns a so-called firing threshold to each transition \( t_i \). In order for the transition \( t_i \) to fire, there has to be enough tokens in its input places \( I(t_i) \), and each of the tokens has to have a degree of confidence \( \kappa \) greater than or equal to the firing threshold \( \theta(t_i) \).

The \( \psi \), \( \lambda_0 \) and \( \delta_0 \) determine the structure of the fuzzy S-T tokens. Additional information can be extracted from a fuzzy S-T token: the total accumulated time \( \delta_{ac} \), i.e., the time durations of all the activities related to the object; the current position of the object \( \lambda_c \) and the degree of confidence about the spatial, temporal or spatio-temporal information \( \kappa \) associated with the S-T token. A fuzzy S-T token carries with it its entire history of visited places, the spatial changes and the time durations of the object’s activities. The complete structure of an S-T token is as follows:

\( \lambda_0(p_i, \delta_{ac}), \psi(p_i), \lambda_0(p_i), \delta_{ac}, \kappa \).

During the initial marking, the S-T tokens with the following structure \( \lambda_0(p_i, \delta_{ac}), \psi(p_i), \lambda_0(p_i), \delta_{ac}, \kappa \) are put into the place \( p_i \in P, 1 \leq i \leq n \).
3.2. Graph of the PeNeFuST

The PeNeFuST can be represented by a bipartite directed multigraph. The circles represent the places, while the bars represent the transitions. The directed arcs connecting the places and the transitions are defined by means of an input function \( I \), while the arcs directed from the transitions to the places are defined by an output function \( O \). Multiple input places and multiple output places are represented by multiple arcs. The fuzzy spatio-temporal tokens are represented by dots (●) in the places. Due to the individuality of the tokens, every dot is labeled with \( m_k^i \in M; i = 1,2,\ldots,r; k = 0,1,2,\ldots \).

3.3. Execution of the PeNeFuST

In general, tokens give dynamical properties to marked Petri nets (PNs) and they are used to define the execution of the marked PNs [39]. In the PeNeFuST, the general rule of execution is slightly modified in the following manner: in the PeNeFuST a transition \( t_j \) is enabled if each of its input places has at least as many fuzzy S-T tokens, having a degree of confidence \( \kappa \) greater than or equal to the firing threshold \( \theta(t_j) \), in it as arcs from the place to the transition and if the time of duration of the object’s activity attached to the place has elapsed. Such S-T tokens are called movable S-T tokens. The firing of an enabled transition in the PeNeFuST is performed automatically and immediately after the transition is enabled. The number of fuzzy S-T tokens at the input and output places of the fired transition is changed in accordance with the basic definition for the original marked PN [39]. The firing of the enabled transition in the PeNeFuST removes the S-T tokens (ancestors) from its input places and simultaneously generates S-T tokens (successors) in its output places. At this moment the structure of a new S-T token is updated by information corresponding to the place \( p_i \) according to \( \Psi(p_i) \).

If the number of movable S-T tokens at the input place \( p_i \) is larger than \( \#(p_i, l(t_j)) \) for the enabled transition \( t_j \) (where \( \# \) denotes the number of appearances of \( p_i \) in the bag \( l(t_j) \)), then different selection of strategies for choosing the S-T tokens—ancestors can be used. For example: (i) the random selection of the S-T tokens; (ii) the LIFO (last-in–first-out) strategy, based on the order of the arrival of the tokens into the place; (iii) the FIFO (first-in–first-out) strategy; or (iv) the selection of the S-T tokens with the stormiest history, i.e., the S-T tokens with the most complex structure. In our knowledge-representation scheme, we use the strategy (iv) to specify the S-T tokens—ancestors, because these S-T tokens contain the richest information needed for the fuzzy spatio-temporal reasoning and they are used for the generation of S-T tokens—successors.

Example 1. Let us suppose that an agent is initially situated in a 2D world of a size 10 x 10 spatial units, at the position A with the approximate coordinates (3.5, 5), as depicted in Fig. 1(a). In Fig. 1(a), the intensity of the gray level represents the values of a fuzzy variable corresponding to the initial (A) and final (B) positions of the object. The degree of confidence that the agent is at the position A is 1.0. The agent is unmovable for approximately 10 time units, and after that it moves with a constant velocity for approximately 7 time units, traversing approximately 4 spatial units in the direction of the x-axis and 2 spatial units in the direction of the y-axis, reaching the final position B (Fig. 1(a)). After achieving the final position, the agent stays there forever.

The described simple scenario can be represented by the generic form of the PeNeFuST. The model consists of two places (\( p_1 \) and \( p_2 \)) and one transition (\( t_1 \)) (Fig. 1(b)) and can be formally represented as follows: \( P = \{p_1, p_2\} \); \( T = \{t_1\} \); \( l(t_1) = \{p_1\} \); \( O(t_1) = \{p_2\} \); \( M = \{m_1^0, m_1^1\} \).

The movement of the agent is modeled by the place \( p_1 \), while the place \( p_2 \) corresponds to the final state of the agent at its final position. The imprecise spatial and temporal information associated with the agent’s activities can be modeled by triangular fuzzy numbers. These

Fig. 1. Representation of a simple, imprecisely known agent movement, with a generic PeNeFuST structure. (a) A movement of an agent. (b) Generic PeNeFuST model.
fuzzy values are associated with the corresponding places of the PeNeFuST by means of the function $\Psi$:

$$\Psi(p_1) = \langle (3,4,5),(1,2,3),(5,7,9) \rangle$$ — the agent moves with a constant velocity for approximately 7 time units, traversing approximately 4 spatial units in the direction of the $x$-axis and 2 spatial units in the direction of the $y$-axis.

$$\Psi(p_2) = \langle (0,0,0),(0,0,0),(\infty,\infty,\infty) \rangle$$ — the agent stays at the final position forever.

The information about the initial position of the agent $\lambda_0$, the initial time of detainment $\delta_0$, as well as the degree of confidence $\kappa$ about the above information is specified in the initial structure of the token $m_1^0$:

$$m_1^0 = \langle (2,3,4),(4,5,6),(7,10,13),(p_1,\langle (3,4,5),(1,2,3),(5,7,9) \rangle,\lambda_c,\delta_0,\kappa) \rangle,$$ where:

$\lambda_0 = \langle (2,3,4),(4,5,6) \rangle$ — the agent is at the position A with the approximate coordinates (3.5).

$\delta_0 = (7,10,13)$ — the agent is unmovable for about 10 time units.

$\langle p_1,\langle (3,4,5),(1,2,3),(5,7,9) \rangle \rangle$ — the agent is moving approximately 4 spatial units in the direction of the $x$-axis and 2 spatial units in the direction of the $y$-axis;

the time duration of its moving is about 7 time units.

$\lambda_c = \lambda_0$ and $\delta_{ac} = \delta_0$.

$\kappa = 1.0$.

Note that all the above 3-tuples represent triangular fuzzy numbers, as shown in Fig. 2.

After approximately 7 time units the token $m_1^0$ becomes movable and the transition $t_1$ is fired, resulting in the removal of the token $m_1^0$ from the place $p_1$, and the simultaneous placement of its successor $m_1^1$ at the place $p_2$.

The structure of the token $m_1^1$ is

$$m_1^1 = \langle \lambda_0,\delta_0,\langle p_1,\Psi(p_1) \rangle,\langle p_2,\Psi(p_2) \rangle,\lambda_c,\delta_{ac},\kappa \rangle,$$

where

$$\langle p_2,\Psi(p_2) \rangle = \langle (0,0,0),(0,0,0),(\infty,\infty,\infty) \rangle$$

indicating that there is no additional moving of the agent in the direction of the $x$-axis and $y$-axis and $(\infty,\infty,\infty)$ indicates that the agent stays at the final position B forever.

The current position of the agent (position B) is $\lambda_c = \langle (6,7,8),(6,7,8) \rangle$, the total accumulated time is $\delta_{ac} = (12,17,22)$, and $\kappa = 1.0$.

---

Fig. 2. Fuzzy values for Example 1: (a) fuzzy values corresponding to the initial position $\lambda_0$ of the agent (the top and middle rows show $x$ and $y$ coordinates, respectively) and the initial temporal delay $\delta_0$ (bottom row); (b) fuzzy values corresponding to the $\Psi(p_1)$ (top and middle rows correspond to the changes in the $x$ and $y$ coordinates, respectively, while the bottom row corresponds to the temporal duration of the activity); (c) fuzzy values corresponding to the $\Psi(p_2)$ (top and middle rows correspond to the changes in the $x$ and $y$ coordinates, respectively, while the bottom row corresponds to the temporal duration of the activity—the object stays forever).
4. FuSpaT knowledge-representation scheme

4.1. Formal definition

The FuSpaT is defined as the following 8-tuples:

\[ \text{FuSpaT} = (\text{PeNeFuST}, \text{TLM}, \text{SLM}, \text{STLM}, \alpha, \beta, \mathcal{L}, F). \]

The components of the above definition can be described as follows:

PeNeFuST is a Petri net with fuzzy spatio-temporal tokens.

TLM is a temporal logical module that supports temporal inference. A detailed description of the TLM will be given in Section 4.1.1.

SLM is a fuzzy spatial logical module that is capable of inferring about the fuzzy spatial relationships between objects. A detailed description of the SLM will be given in Section 4.1.2.

STLM is a fuzzy spatio-temporal module that integrates the spatial information obtained from the SLM and the corresponding temporal information from the TLM into spatio-temporal information. It is described in more detail in Section 4.1.3.

A function \( \alpha : P \rightarrow (Ac \cup Cs) \) is a bijective function from a set of places \( P \) to a union of a set of activities and/or states \( Ac \) and a set of control states \( Cs \).

A surjective function \( \beta : T \rightarrow (Ev \cup Ce) \) is a mapping from a set of transitions \( T \) to a union of a set of events \( Ev \) and a set of control events \( Ce \).

The functions \( \alpha \) and \( \beta \) give to the FuSpaT a semantical interpretation of the model.

\( \mathcal{L} \) is a linguistic variable used to express the degree of confidence (related to the temporal, spatial or spatio-temporal relationships) in a user-friendly form. The values of the linguistic variable \( \mathcal{L} \) are from the following set: \{not true, minimally true, minorly true, more-or-less true, moderately true, considerably true, very true, extremely true, always true\}. The values of the linguistic variable \( \mathcal{L} \) are transformed to the intervals according to Table 1 [50].

<table>
<thead>
<tr>
<th>Linguistic variable</th>
<th>Numerical interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always true</td>
<td>[1.00, 1.00]</td>
</tr>
<tr>
<td>Extremely true</td>
<td>[0.95, 0.99]</td>
</tr>
<tr>
<td>Very true</td>
<td>[0.80, 0.94]</td>
</tr>
<tr>
<td>Considerably true</td>
<td>[0.65, 0.79]</td>
</tr>
<tr>
<td>Moderately true</td>
<td>[0.45, 0.64]</td>
</tr>
<tr>
<td>More or less true</td>
<td>[0.30, 0.44]</td>
</tr>
<tr>
<td>Minorly true</td>
<td>[0.10, 0.29]</td>
</tr>
<tr>
<td>Minimally true</td>
<td>[0.01, 0.09]</td>
</tr>
<tr>
<td>Not true</td>
<td>[0.00, 0.00]</td>
</tr>
</tbody>
</table>

The evaluation of the \( tr \), \( sr \) and \( str \) is based on information that is carried by the S-T tokens from places \( p_i \) and \( p_j \). This information is transferred to the TLM or SLM or STLM by sending copies of the S-T tokens to it. The destination \((TLM, SLM \text{ or } STLM)\) of the S-T token copies depends on the type of specified relationship.

The \( p_{i1}, p_{i2}, \ldots, p_{i13} \), where \( \alpha(p_{i1}) \in Cs, \alpha(p_{i2}) \in Cs, \ldots, \alpha(p_{i13}) \in Cs \), in the flag \( f_i \) specify the places in which the TLM, SLM or STLM puts the tokens, depending on the result of the evaluation specified in \( (tr \text{ or } sr \text{ or } str) \). Such places are called control places. A transition having one or more control places as an input place is called a control transition. The tokens, called control tokens, are treated as S-T tokens without temporal and spatial histories.

A degenerative type of flag \( f_{ig} = (p_{i1}, \ldots, p_{ig}, \ldots, p_{i13}) \), \( 1 \leq g \leq n \), \( n \) is the cardinality of a set \( P \), is used to denote the goal state of the system; where \( p_g \) denotes the place that corresponds to one of the goals of the modeled system.

4.1.1. Temporal logical module (TLM)

A TLM is a temporal logical module that supports, in general, the following temporal relationships: 13 Allen’s time-interval relationships [47], five relationships between the time point and the time interval, and three temporal relationships between the time points. The relationships time-point–time-interval and time-point–time-point are obtained by letting one or both of the intervals degenerate to a time point(s) [51].

The inputs to the TLM are copies of two fuzzy S-T tokens and a specification of the temporal relationship that has to be evaluated. An output of the TLM is a control token with a value \( \kappa \) equal to the degree of confidence that the relationship is satisfied. The degree of confidence is internally expressed by a value from the interval \([0, 1]\). It can be represented to user or expert by means of a value of the linguistic variable \( \mathcal{L} \). The additional output of the TLM is a temporal relation with its degree of confidence, which is directed to the spatio-temporal logical module STLM. The temporal relationships are generalized to the fuzzy case in the following manner. First, the 13 temporal relationships have to be expressed by means of the relationships between the starting \((A^- \text{ or } B^-)\) and ending \((A^+ \text{ or } B^+)\) time points of the intervals \( A \text{ and } B \), possibly connected by means of logical operators (Table 2) [52]. There are three possible relationships between the starting and ending time points of the intervals: before \((.<.\) ), after \((>.\) ) and equal \((=.)\). For example,
Based on the above extension principle, the confidence degrees of the fuzzy relationships \( \leq_f \) and \( \geq_f \) are first defined as a basis for the relationships \( <_f, >_f \) and \( =_f \). For two triangular fuzzy numbers \( A = (a_1, a_2, a_3) \) and \( B = (b_1, b_2, b_3) \) the fuzzy relationships \( A \leq_f B \) and \( A \geq_f B \) are defined as

\[
\mu_{A \leq_f B} = \begin{cases} 
0 & \text{for } a_1 > b_3 \text{ & } b_2 < a_2, \\
\frac{b_3 - a_1 + a_2 - b_2}{b_3 - a_1 + a_2 - b_2} & \text{for } a_1 \leq b_3 \text{ & } b_2 \leq a_2, \\
1 & \text{for } a_2 \leq b_2.
\end{cases}
\]

\[
\mu_{A \geq_f B} = \begin{cases} 
0 & \text{for } b_1 > a_3, \\
\frac{a_3 - b_1 + a_2 - b_2}{a_3 - b_1 + a_2 - b_2} & \text{for } b_1 \leq a_3 \text{ & } a_2 < b_2, \\
1 & \text{for } b_2 \leq a_2.
\end{cases}
\]

The logical operators \( \land, \lor \) and \( \neg \) are generalized to the fuzzy case using well-known equations [25,49]:

\[
\mu_{A \land B} = \min(\mu_A, \mu_B),
\]
\[
\mu_{A \lor B} = \max(\mu_A, \mu_B),
\]
\[
\mu_{\neg A} = 1 - \mu_A.
\]

The fuzzy relationships \( =_f, <_f \) and \( >_f \) can then be expressed as

\[
A =_f B \iff (A \leq_f B) \land (A \geq_f B),
\]
\[
A <_f B \iff (A \leq_f B) \land \neg(A \geq_f B),
\]
\[
A >_f B \iff (A \geq_f B) \land \neg(A \leq_f B),
\]

which gives

\[
\mu_{A =_f B} = \min(\mu_{A \leq_f B}, \mu_{A \geq_f B}),
\]
\[
\mu_{A <_f B} = \min(\mu_{A \leq_f B}, 1 - \mu_{A =_f B}),
\]
\[
\mu_{A >_f B} = \min(\mu_{A \geq_f B}, 1 - \mu_{A =_f B}).
\]

Example 2. Two fuzzy intervals \( A = (A^-, A^+) \) and \( B = (B^-, B^+) \) are specified by their starting and ending points that are defined as triangular fuzzy numbers: \( A^- = (1, 2, 3); A^+ = (5, 6, 7); B^- = (1, 3, 5); B^+ = (7, 9, 10), \) as shown in Fig. 3.

The fuzzy relationship “during” (\( A \text{ during } B \)) can be evaluated as follows. The relationship “during” is defined as: \( A \text{ during } B = (A^- > B^-) \land (A^+ < B^+) \) (Table 2). Therefore, the degree of confidence for the relationship “during” can be expressed as

\[
\mu_{A \text{ during } B} = \min(\mu_{A^- > B^-}, \mu_{A^+ < B^+})
\]
\[
= \min(\mu_{A^- > B^-}, 1 - \mu_{A^- =_B B^-}), \min(\mu_{A^- =_B B^-}, 1 - \mu_{A^- =_B B^-}), \min(\mu_{A^- =_B B^-}, 1 - \mu_{A^- =_B B^-}))
\]
\[
= \min(\mu_{A^+ < B^+}, 1 - \min(\mu_{A^- > B^-}, \mu_{A^- =_B B^-}), \min(\mu_{A^- =_B B^-}, 1 - \min(\mu_{A^- > B^-}, \mu_{A^- =_B B^-}))))
\]
By using the equations for the fuzzy relationships $\geq_f$ and $\leq_f$, the following values can be calculated:

$$
\mu_{A^+ B^-} = \frac{-3+1-3+2}{3-1+3-2} = \frac{2}{3},
$$

$$
\mu_{A^- B^+} = 1,
$$

$$
\mu_{A^+ B^+} = \frac{7-7}{7-7+9-6} = 0,
$$

$$
\mu_{A^- B^-} = 1.
$$

Substitution of the calculated values into the equation for $\mu_{A^+ B^-}$ gives

$$
\mu_{A^+ B^-} = \min(\mu_{A^+ B^-}, \mu_{B^+ A^-}, \mu_{A^- B^-}, \mu_{B^- A^+})
= \min(\min(\frac{2}{3}, 1-\frac{2}{3}), \min(1, 1-0))
= \min(\frac{2}{3}, 1) = \min(\frac{2}{3}, 1) = \frac{2}{3}.
$$

Therefore, the fuzzy relationship “during” between the fuzzy intervals $A$ and $B$ is satisfied with the degree of confidence $\kappa$ equal to $\frac{2}{3}$. In the user friendly form, the degree of confidence $\kappa = \frac{2}{3}$ is represented by the value of the linguistic variable “more or less true” (Table 1).

### 4.1.2. Spatial logical module (SLM)

A SLM is a fuzzy spatial logical module that is capable of inferring about the fuzzy spatial relationships between objects. In general, the combinations of possible 2D spatial relationships between two non-concave objects are represented by 169 relationships [54]. These relationships are based on an analogy with the 13 well-known Allen’s time-interval relationships [47], extended to 2D space. These 169 relationships can also be considered as an extension of the RCC8 [55] where regions are represented by the minimum bounding boxes, in the sense that they give more precise information about 2D spatial relationships.

In many applications, however, the positions of the physical objects or agents can be represented by points (for example, in robot-vision systems the position of an object is often represented by a point corresponding to the centroid); therefore, in the paper we consider only 2D relationships between objects that are reduced to points. The spatial relationships of the objects that are represented by points have an analogy to Allen’s time-interval relationships extended to 2D, where the time intervals degenerate to the time points. In this case, there are only three relationships in the time domain [51], and consequently only nine crisp relationships in the 2D spatial domain (Fig. 4). These nine spatial relationships between the objects $A$ and $B$ can be denoted as: $A \text{ lb } B$ ($A$ is to the left and below $B$), $A \text{ b } B$ ($A$ is below $B$), $A \text{ rb } B$ ($A$ is to the right and below $B$), $A \text{ l } B$ ($A$ is to the left of $B$), $A \text{ m } B$ ($A$ meets $B$), $A \text{ r } B$ ($A$ is to the right of $B$), $A \text{ la } B$ ($A$ is to the left and above $B$), $A \text{ a } B$ ($A$ is above $B$), and finally $A \text{ ra } B$ ($A$ is to the right and above $B$).

The inputs to the SLM are copies of two fuzzy S-T tokens and a specification of the spatial relationship that has to be evaluated. An output of the SLM is a control token having the degree of confidence $\kappa$ corresponding to the degree of confidence that the relationship is satisfied (in the interval $[0, 1]$). The additional output of the SLM is a spatial relation with its degree of confidence, which is directed to the spatio-temporal logical module STLM. The evaluation of the degree of confidence of the spatial relation is also based on the extension principle. In order to calculate a degree of confidence for the specified relationship between the two fuzzy values, nine relationships have to be expressed by means of the relationships between the $x$ and $y$ coordinates of the objects, connected with logical operators (Table 3). The crisp relationships between the coordinates of the objects are: less ($<$), greater ($>$) and equal ($=$). For example, the relationship $A \text{ lb } B$ (object $A$ is left and below the object $B$) can be expressed as $(x_A < x_B) \land (y_A > y_B)$, where $(x_A, y_A)$ and $(x_B, y_B)$ are the coordinates of the objects $A$ and $B$, respectively. All the values $x_A, x_B, y_A$ and $y_B$ can be generalized to the fuzzy values and represented by fuzzy triangular numbers. The relationships $<, >$ and $=$ are generalized to the fuzzy case in an identical way as in Section 4.1.1.

#### 4.1.3. Spatio-temporal logical module (STLM)

A STLM integrates the spatial information obtained from the SLM and the corresponding temporal
information from the TLM into spatio-temporal information. Based on the combination of 13 temporal relationships and nine spatial relationships (Fig. 4), a total of 117 spatio-temporal relationships are supported by the STLM (Table 4).

A STLM is implemented as a look-up table with three inputs: (i) the spatial relationship obtained from the SLM accompanied by its degree of confidence; (ii) the temporal relationship obtained from the TLM accompanied by its degree of confidence; and (iii) a set of flags F. The output is coded and represented in the form of a semantical interpretation of the spatio-temporal relationship (Table 4). For example, if the inputs are “A left and below B” (spatial relationship) and “X=Y” (temporal relationship), the corresponding entry from Table 4 “X left and below Y” is interpreted as: “The agent X is to the left and below the agent Y and their activities are simultaneous”. The result of the evaluation of the spatio-temporal relationship by the STLM is a control token containing a degree of confidence that the spatio-temporal relationship is satisfied (in the interval [0, 1]), obtained as a fuzzy logical and (\(\wedge\)) between the degrees of confidence for the corresponding spatial and temporal relationships. The value of the linguistic variable L is used for a user-friendly interpretation of the degree of confidence that the spatio-temporal relationship is satisfied.

4.2. Reasoning

The spatial and temporal information is contained in the FuSpaT model of the world. The reasoning process in the proposed knowledge-representation scheme is defined as a spatio-temporal data-driven process as follows: The input in the reasoning process is the initial distribution of S-T tokens in the FuSpaT model. This distribution determines the current positions, the activities and the states and spatio-temporal relationships between the activities of the objects or the agents. As time proceeds, the enabled transitions are automatically fired and the S-T tokens are distributed through the graph of the PeNeFuST. This corresponds to changes of the objects’ or agents’ positions and their activities. Depending on their path through the graph, the time duration of the activities (states), and the corresponding spatial changes of the objects, each S-T token carries a history of the execution of the PeNeFuST. The firing sequences are additionally controlled by the TLM, SLM or STLM.

The main step of reasoning in the proposed model can be described as follows (Fig. 5). When an S-T token arrives at the place denoted by a flag, its copies are sent to the TLM and the SLM (in general, if the relationship specified by the flag is str—spatio-temporal relationship). At the moment when another S-T token arrives at another place denoted by the same flag, its copies are also sent to the TLM and the SLM. Note that the additional input for the TLM and the SLM is a set of flags F. When the TLM and the SLM have received both copies of the S-T tokens, the following simultaneous activities are performed:

(i) The TLM evaluates the temporal relationship specified by the flag (extracted from str) as it was described in Section 4.1.1. The output of the TLM is a control token with a degree of confidence of the specified temporal relationship which is sent to the knowledge base (the case when the flag specifies only the temporal relationship). The additional output of the TLM is the temporal relationship and its degree of confidence, directed to the STLM (in the case when the spatio-temporal relationship is specified). The optional output of the TLM is the user-friendly interpretation of the degree of confidence of the specified temporal relationship by means of a value of the linguistic variable L.

(ii) The SLM evaluates the spatial relationship specified by the flag (extracted from str) (see Section 4.1.2). The output of the SLM is a control token with a degree of confidence of the specified spatial relationship which is sent back to the knowledge base (the case when the flag specifies only the spatial relationship). Additionally, the output of the SLM can be the spatial relationship and its degree of confidence, directed to the STLM (in the case when the spatio-temporal relationship is specified). Also, as an optional output of the SLM there is the value of the linguistic variable L—the user-friendly interpretation of the degree of confidence of the specified spatial relationship.

The outputs of the TLM and the SLM (the temporal relationship and its degree of confidence, and the spatial relationship and its degree of confidence, respectively) are sent to the STLM. The STLM is realized as look-up table where the temporal relationships correspond to the rows of the look-up table, while the spatial relationships correspond to the columns. The spatio-temporal relationship for specific inputs of the STLM is obtained as the content of the table-entry on crossing of the corresponding row and column of the look-up table. The degree of

<table>
<thead>
<tr>
<th>Table 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition of nine spatial relationships between the objects A and B.</td>
</tr>
<tr>
<td>A lb B: (A left and below B)</td>
</tr>
<tr>
<td>A b B: (A below B)</td>
</tr>
<tr>
<td>A rb B: (A right and below B)</td>
</tr>
<tr>
<td>A l B: (A left of B)</td>
</tr>
<tr>
<td>A m B: (A meets B)</td>
</tr>
<tr>
<td>A r B: (A right of B)</td>
</tr>
<tr>
<td>A la B: (A left and above B)</td>
</tr>
<tr>
<td>A a B: (A above B)</td>
</tr>
<tr>
<td>A ra B: (A right and above B)</td>
</tr>
</tbody>
</table>
The confidence of the spatio-temporal relationship is determined as a minimum of the degrees of confidence for temporal and spatial relationships, i.e., as a fuzzy logical AND operation.

Based on the flag, the STLM sends control token(s) with the obtained degree of confidence to the control place(s) of the modeled system. Optionally, the additional user-friendly output of the STML is the spatio–temporal relationship with the degree of confidence expressed by the value of the linguistic variable $L$. The value of the linguistic variable $L$ is obtained by mapping the degree of confidence $k$ to the set of possible values of $L$ (see Table 1).

Each control token carries an appropriate degree of confidence $k \in [0, 1]$, corresponding to the degree of confidence that the spatial, temporal or spatio-temporal relationship is satisfied. These control tokens have an influence on the result of the firing sequence by means of enabling transitions (the transition $t_j$ becomes enabled if there are enough tokens with $k \geq \delta(t_j)$ in its input places). The combination of the S-T tokens associated with activities and the control tokens, both present at the same time, can be interpreted as a fuzzy spatio-temporally dependent if–then rule implementation: If there are enough S-T and control tokens at the corresponding input places with fuzzy confidence degrees $k$ greater than or equal to the transition firing threshold $\delta$, the transition is automatically fired, i.e., the spatio-temporally dependent rule is activated and the action (or conclusion) is

---

**Table 4**

Representation of the 117 spatio-temporal relationships supported by the STLM; $\circ$ and $\bullet$ represent the positions of the objects X and Y, respectively; $xxx$ and $yyy$ denote the corresponding time intervals.

<table>
<thead>
<tr>
<th>X &lt; Y</th>
<th>X b &lt; Y</th>
<th>X b &lt; Y</th>
<th>X rb &lt; Y</th>
<th>X I &lt; Y</th>
<th>X m &lt; Y</th>
<th>X r &lt; Y</th>
<th>X la &lt; Y</th>
<th>X a &lt; Y</th>
<th>X ra &lt; Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
<td>$\circ$</td>
</tr>
</tbody>
</table>

---

Confidence of the spatio-temporal relationship is determined as a minimum of the degrees of confidence for temporal and spatial relationships, i.e., as a fuzzy logical AND operation.

The confidence of the spatio-temporal relationship is determined as a minimum of the degrees of confidence for temporal and spatial relationships, i.e., as a fuzzy logical AND operation. Based on the flag, the STLM sends control token(s) with the obtained degree of confidence to the control place(s) of the modeled system. Optionally, the additional user-friendly output of the STML is the spatio–temporal relationship with the degree of confidence expressed by the value of the linguistic variable $L$. The value of the linguistic variable $L$ is obtained by mapping the degree of confidence $k$ to the set of possible values of $L$ (see Table 1).

Each control token carries an appropriate degree of confidence $k \in [0, 1]$, corresponding to the degree of confidence that the spatial, temporal or spatio-temporal relationship is satisfied. These control tokens have an influence on the result of the firing sequence by means of enabling transitions (the transition $t_j$ becomes enabled if there are enough tokens with $k \geq \delta(t_j)$ in its input places). The combination of the S-T tokens associated with activities and the control tokens, both present at the same time, can be interpreted as a fuzzy spatio-temporally dependent if–then rule implementation: If there are enough S-T and control tokens at the corresponding input places with fuzzy confidence degrees $k$ greater than or equal to the transition firing threshold $\delta$, the transition is automatically fired, i.e., the spatio-temporally dependent rule is activated and the action (or conclusion) is
generated. It is obvious that the above-described process is driven by spatio-temporal events. If some places in the graph are denoted as goal states, the scheme can conclude if these spatio-temporally dependent goals may be achieved. Based on information in the S-T token that has achieved the goal state, the sequences of activities (states) that lead to the goal can be registered. By varying the initial marking of the scheme it can be used for planning in spatio-temporally rich domains.

4.3. Formal analysis of the FuSpaT model

In general, analysis problems for Petri net-based models, such as safeness, boundedness, conservation, etc., are based on the reachability tree [39,56], which is a finite representation of all the markings of the Petri net that can be reached from the initial marking. The nodes of the reachability tree represent the markings of the Petri net and its arcs represent possible changes in state resulting from the firing of transitions.

In our proposed model, however, a direct application of the reachability tree is not possible, because the original reachability tree construction algorithm presupposes that new markings can result only by firing enabled transitions in the current marking. In the FuSpaT model, however, this is not the case, because new markings can be achieved by placing control tokens to control places. These control tokens can then result in enabling additional transitions that would not be enabled otherwise.

In order to overcome the above-described problem and to enable a formal analysis of the FuSpaT model, we propose the construction of a modified reachability tree called the conditional reachability tree. The construction of the conditional reachability tree is based on the distinction between the two possible cases of reaching the new marking. The unconditional immediate reachability is defined in the same way as the immediate reachability in the generalized Petri nets: the marking $\mu'$ is unconditionally immediately reachable from the marking $\mu$ if there exists a transition $t_j$ such that $t_j$ is enabled in $\mu$ and its firing results in $\mu'$:

$$\exists t_j : \delta^*(\mu, t_j) = \mu',$$

where $\delta^*$ is the next-state function [39], modified according to the execution of the PeNeFuST.

The conditional immediate reachability is defined to account for the markings that can be reached only if control tokens are placed to control places. The marking $\mu'$ is conditionally immediately reachable from the marking $\mu$ if there exists another marking $\mu''$ that can be reached from $\mu$ by placing certain control tokens to control places and $\mu'$ is unconditionally immediately reachable from $\mu''$:

$$\exists \mu'' \exists \mu_{ctrl} \exists t_j : (\delta^*(\mu, t_j) = \mu') \wedge (\mu'' = \mu + \mu_{ctrl}),$$

where $\mu_{ctrl}$ denotes the marking vector having non-zero values only at the positions corresponding to the control places, and $+$ denotes the vector addition.

Based on the above definitions, the algorithm of the conditional reachability tree construction can be developed. The algorithm is based on the original reachability tree algorithm [39], with appropriate modifications related to the conditional reachability.

The conditional reachability tree can be graphically represented in a similar manner to the reachability tree of the generalized Petri net, except that the arcs corresponding to the conditional reachability are represented by dashed lines, with the specification of the conditions that have to be satisfied specified near the arc.

Based on the proposed model FuSpaT, an object-oriented simulator was developed. Some components of the program simulator are illustrated in the example that follows.

5. An example

In this section we give an example of using the FuSpaT model for planning in a simple dynamical scene. The
scene represents a detail of a simplified scenario from the world of robot soccer. Robot soccer is selected because it provides a test bed where different models and algorithms can be tested and where many real-world characteristics are present. Two robots are situated on the pitch with a size of 60 × 80 spatial units (Fig. 6).

Robot A (Team 1) is initially situated at the crisp position (30,60) and it initially possesses the ball. Robot B (Team 2) is situated at the crisp position (20,20). There are three possible strategies available to Robot A:

(a) to shoot the ball directly towards the goal;
(b) to try to bypass Robot B and then shoot the ball towards the goal;
(c) to shoot the ball towards the perimeter fence of the pitch in such a way that the ball rebounds in the direction of the goal.

Robot B chooses its activity based on the activity of Robot A. If Robot A chooses the strategy (a), Robot B moves towards the center of the pitch and tries to intercept the ball. If Robot A chooses strategy (b) or (c), Robot B moves towards the right and tries to intercept the ball.

Let us suppose that we want to find an answer to the following question: can Robot A achieve the goal state “the ball is in the goal”, and, if it can, which strategy should be selected to achieve the goal state?

The described scenario can be modeled by the FuSpaT model depicted in Fig. 7.

The sets of places and transitions, the input and output functions, as well as the semantic meanings of the places and transitions specified by means of the functions $\alpha$ and $\beta$ are denoted in Fig 7.

A firing threshold $\beta(t_i) = 0.4; i = 4, 5, \ldots, 11$ is experimentally assigned to the control transitions $t_{i_4}, t_{i_5}, t_{i_6}, t_{i_7}, t_{i_8}, t_{i_9}, t_{i_{10}}$ and $t_{i_{11}}$. All the other transitions have a firing threshold equal to 0:

$\beta(t_{i_1}) = \beta(t_{i_2}) = \beta(t_{i_3}) = 0,$

$\beta(t_{i_4}) = \beta(t_{i_5}) = \beta(t_{i_6}) = \beta(t_{i_7}) = \beta(t_{i_8}) = \beta(t_{i_9})$

$= \beta(t_{i_{10}}) = \beta(t_{i_{11}}) = 0.4.$

Let us further suppose that the temporal and spatial information about the activities is available based on prior knowledge about the characteristics of the robots and the speed of the ball movement, and that it can be modeled by means of the function $\Psi$ as follows.

For Robot A:

$\Psi(p_1) = (\langle(0,0,0),(0,0,0)\rangle,(0,0,0));$  \hspace{1cm} $\Psi(p_2) = (\langle(-5,0,5),(30,30,30)\rangle,(3,3,3));$  \hspace{1cm} $\Psi(p_3) = (\langle(15,20,25),(35,-30,-25)\rangle,(4,5,6));$  \hspace{1cm} $\Psi(p_4) = (\langle(25,30,35),(35,-30,-25)\rangle,(1,2,3));$  \hspace{1cm} $\Psi(p_5) = (\langle(0,0),(0,0)\rangle,(0,0,0));$  \hspace{1cm} $\Psi(p_6) = (\langle(0,0),(0,0)\rangle,(0,0,0));$  \hspace{1cm} $\Psi(p_7) = (\langle(0,0),(0,0)\rangle,(0,0,0));$  \hspace{1cm} $\Psi(p_8) = (\langle(-25,-20,-15),(35,-30,-25)\rangle,(1,2,3));$

![Fig. 6. A simplified scenario from the robot-soccer world.](image-url)
\[ \Psi(p_9) = \langle \langle -25, -20, -15 \rangle, \langle -35, -30, -25 \rangle \rangle, (1, 2, 3) \rangle; \]
\[ \Psi(p_{10}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle; \]
\[ \Psi(p_{11}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle; \]
\[ \Psi(p_{12}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle; \]
\[ \Psi(p_{13}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle; \]
\[ \Psi(p_{14}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle, (\infty, \infty, \infty) \rangle. \]

For example, \( \Psi(p_9) = \langle \langle -25, -20, -15 \rangle, \langle -35, -30, -25 \rangle \rangle, (1, 2, 3) \rangle \) specifies as follows: a pair \( \langle -25, -20, -15 \rangle, \langle -35, -30, -25 \rangle \rangle \) denotes the fuzzy change of the position of the ball in the \( x \) and \( y \) directions, respectively, while the triplet \( (1, 2, 3) \) represents the temporal duration of this activity (about two time units).

For Robot B:
\[ \Psi(p_{15}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 1, 2, 3 \rangle \rangle; \]
\[ \Psi(p_{16}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle; \]
\[ \Psi(p_{17}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle \rangle; \]
\[ \Psi(p_{18}) = \langle \langle 8, 10, 12 \rangle, \langle 8, 10, 12 \rangle, \langle 1, 2, 3 \rangle \rangle; \]
\[ \Psi(p_{19}) = \langle \langle 25, 30, 35 \rangle, \langle 8, 10, 12 \rangle, \langle 4, 5, 6 \rangle \rangle; \]
\[ \Psi(p_{20}) = \langle \langle 0, 0, 0 \rangle, \langle 0, 0, 0 \rangle, (\infty, \infty, \infty) \rangle. \]

The interaction between the robots in the scene is modeled by a set of flags

\[ F = \{ f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{G1} \}. \]
where \( f_1 = (p_2, p_{18}, \text{str1}, p_3) \); \( f_2 = (p_3, p_{18}, \text{str2}, p_6) \); \( f_3 = (p_8, p_{18}, \text{str3}, p_{10}) \); \( f_4 = (p_2, p_{19}, \text{str4}, p_9) \); \( f_5 = (p_6, p_{19}, \text{str5}, p_{11}) \); \( f_6 = (p_2, p_{15}, \text{str6}, p_{16}) \); \( f_7 = (p_3, p_{15}, \text{str7}, p_{17}) \); \( f_8 = (p_4, p_{15}, \text{str8}, p_{19}) \); \( f_9 = (p_2, p_{20}, \text{str9}, p_{13}) \) and \( f_{11} = (p_{14}, -- , --) \).

For example, the flag \( f_1 = (p_2, p_{18}, \text{str1}, p_3) \) specifies the evaluation of the spatio-temporal relationship \( \text{str1} \) based on the information obtained from the S-T tokens at the places \( p_2 \) and \( p_{18} \), where the semantic interpretations of \( p_2 \) and \( p_{18} \) are “the ball is approaching the goal” and “Robot B is moving to the center of the pitch”, respectively. The S-T relationship \( \text{str1} \) is defined as \( \text{str1} = U_{\{m = \text{mm}, \text{mo}, \text{md}, \text{ms}, \text{mf}, \text{mmi}, \text{moi}, \text{mdi}, \text{msi}, \text{mf}\}} \) (see Table 4), meaning that \( \text{str1} \) includes all the possible S-T relationships (connected with logical or), except the relationships that represent the spatial meeting of the ball and the Robot B in the overlapping temporal periods. If the relationship \( \text{str1} \) is satisfied, the control token is placed at the control place \( p_9 \). If the token has a degree of confidence greater than or equal to 0.4, the control transition \( t_{14} \) will be enabled (see Fig. 7).

The meaning of the other flags can be described in an analogous way. In our example, the spatio-temporal relationships are \( \text{str2} = \text{str3} = \text{str4} = \text{str5} = \text{str1} \).

The flags \( f_2, f_3 \) and \( f_8 \) enable the selection of the Robot B’s activity. Since the selection of the Robot B’s activity depends only on the activity of Robot A, \( \text{str6} = \text{str7} = \text{str8} = U, \) where \( U \) represents a set of all possible S-T relationships.

Flag \( f_9 \) enables the entrance of the ball into the goal if the ball hits the spatial area of the goal, i.e., if the spatial relationship \( \text{str9} = m \) is satisfied.

The degenerative flag \( f_{11} \) denotes the goal state of the system, “The ball is in the goal”.

The initial marking of the PeNeFuST is

\[
\Omega_0 = \{(m_1^0),0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
The transition $t_6$ becomes enabled because the degree of confidence of control token is greater than 0.4, and it automatically fires. In the subsequent steps, the relationship $str_5$ (specified by the flag $f_5$) is evaluated by the STLM, the control token having $\kappa = 0.5$ is put into the control place $p_{11}$, and the fuzzy S-T token $m_3$ finally reaches the place $p_{14}$, denoted as a goal place. Therefore, it can be inferred that the strategy (c) leads to the goal. By resetting the simulation and trying other strategies, the user can find that other strategies do not lead to the goal for the same initial conditions.

6. Conclusion

The original high-level Petri nets, called Petri nets with fuzzy spatio-temporal tokens (PeNeFuST) are used for the modeling, planning and analyzing activities in spatio-temporal domains. Based on the PeNeFuST and the knowledge-representation scheme called FuSpaT, the model of the fuzzy spatio–temporal knowledge representation and reasoning is proposed. This model is used to build a knowledge base and support fuzzy spatio-temporal reasoning for problems that require the integration of both spatial and temporal information.

The proposed FuSpaT scheme uses the spatio-temporal logical module (STLM), which is implemented as a look-up table that integrates the spatial and temporal information obtained from the spatial (SLM) and temporal (TLM) logical modules. In our model, the STLM supports 117 spatio-temporal relationships (9 spatial relationships × 13 temporal relationships). The proposed model supports a spatio-temporal data-driven reasoning process, and unified representations of different temporal, as well as spatial and spatio-temporal information, the ability to use...
linguistic variables to represent user or expert belief in the truth of the temporal, spatial and/or spatio-temporal relationships, the ability of independent modeling of activities for each of the agents, and the specifying of their interactions by means of the flags. One of the advantages is the existence of well-defined methods (based on the modified PN theory) for the analysis of the different spatial and temporal relationships among...
the agents or objects by changing the initial marking and spatial and time values assigned to the S-T tokens and places. The proposed model allows a hierarchical representation of the scenes on the different abstraction levels (based on well-known concepts of the PN called refinement and abstraction). It is domain independent, but suitable for the independent modeling of multiple agents in multi-agent systems. Based on the proposed theory and model of fuzzy spatio-temporal knowledge representation and inference, the program simulator and tools for the analysis have been developed in the Ci+ environment, enabling the use of the model in different application areas. Future work will consist of a further experimental validation of the proposed model.

References