Abstract—Modeling of complex phenomena such as the mind presents tremendous computational complexity challenges. The Neural Modeling (NMF) Fields theory and Phenomena Dynamic Logic (PDL) address these challenges in a non-traditional way. The main idea behind its success is matching the levels of uncertainty of the problem/model and the levels of uncertainty of the evaluation criterion used to identify the model. When a model becomes more certain then the evaluation criterion is also adjusted dynamically to match the adjusted model. This process mimics processes of the mind and natural evolution at the neural level. This paper describes the generalization of P-DL for data fusion and mining of heterogeneous spatial objects in cyber-physical space.

1. INTRODUCTION

The emergence of the global cyber-physical space with a multiple static and moving sensors and other entities that communicate with each other and supply data at the unprecedented rate creates both new challenges and opportunities.


The goal of this paper is generalization of the Neural Modeling Fields theory and Phenomena Dynamic Logic for heterogeneous spatial data in cyber-physical space.

DMF technologies heavily depend on topology and geometry representation of spatial objects in cyber-physical space and appropriate similarity measures. Types of connections, orientation, shape, proximity, similarity, and their measurement need to be appropriately represented. Representations and their hierarchies should support computationally efficient reasoning with these spatial data.

There are two current trends in modeling cyber-physical phenomena and logic. In modeling cyber-physical phenomena, it is adding more logic structures to the classical mathematical techniques. In logic, it is "dynamification" of logic [Harel al 200; Baltag et al., 2008, van Bentlem, Harel et al 200; Baltag et al., 2008, van Bentlem, 1996, 2007, van Ditmarsch, 2007, Leitgeb, Segerberg, 2007], with various new non-classical logics about actions, rather than about propositions. The subjects in this area include Action Logic, Arrow Logic, Game Logic, Semantic games, dialogue logic, Belief Revision, Dynamic Epistemic Logic, Hoare logic, Dynamic Logic, Linear Logic, labeled transition systems, Petri nets, Process Algebra, Automata Theory, Game Semantics, coalgebras etc.

These complimentary trends can benefit both areas. In this paper, we use dynamic logic approach for developing a theory for fusion and mining spatial data in the cyber-physical space.

Spatial distribution of the dynamic network is a complex and dynamic mixture of its topology and geometry [Tossebro, Nygård, 2006]. The geometric and topological structures of such spatial objects as communication networks, roads, waterways, molecules, social networks, etc can be quite uncertain when noise, obstacles, temporary loss of communication and other factors involved. The change of the network structure over time is another challenge.

The net-centric activities such as tracking objects require sophisticated sensor network capabilities, where network topology and geometry play a critical role to be able to fuse and mine multi-source information. One of them is discovering patterns of routes that deviate from normal routes. The use of topology and geometry must go beyond the traditional data mining (DM) setting that exploits topology and geometry of routes that already in the database. In such traditional setting, we might not discover new anomalous routes that are not in the data base yet. Thus, topology and geometry of the network should be treated as additional data source relative to already recorded routes.

Deep analysis of social networks also requires sophisticated analysis of their dynamic topology and geometry that includes the dynamic change of the geolocation of groups and individuals.

These examples illustrate the need to effectively represent, fuse, mine, and reason with data that are both quantitative and qualitative at different levels of granularity, described over a network graph that is uncertain at some level itself. This requires to use sophisticated human-based similarity measures [Santini, Jain, 1999, Kamgar-Parsi et al, 2001]

To generalize the Neural Modeling Theory and Phenomena Dynamic Logic for heterogeneous spatial data in cyber-physical space we are developing a methodology for:
Axioms. The generality relation can be defined as follows:

If and only if $T_i \geq T_j$, where $T_i$ is a system of axioms, then $T_i$ is equal to or more general than $T_j$.

This paper is organized as follows. We start from the concept of the Phenomena Dynamic Logic (PDL) in Section II where the concepts of uncertainty, generality, and simplicity for models and evaluation criteria are established. Section III defines generalization of dynamic logic for data fusion and mining of heterogeneous data. It also provides examples of dealing with uncertain topology and geometry of spatial objects. Section V summarizes the paper and discusses future research.

The model of the phenomenon or the system is not a structure but a theory, often in a formal language [Hodes, 2005]. For short, we call a model of phenomenon a P-model in contrast with a logic model denoted as an L-model that rather represents a structure. For more formal definitions, see [Kovalerchuk, Perlovsky, 2008].

The next PDL concept is a similarity (or correspondence) measure $L(M, E)$ between empirical data $E$ and a priori model $M$ that is assigned individually to each specific problem and $E$: $L: \{(M, E)\} \rightarrow R$, where $R$ is a set or real numbers.

II. PHENOMENA DYNAMIC LOGIC

A. P-models: Uncertainty, generality and simplicity relations

Below we introduce the concepts of uncertainty, generality, and simplicity as partial order relations. These concepts can be specified to both logic and MFT models.

An uncertainty relation between P-models is denoted as $\geq_{Mu}$, relation $M_i \geq_{Mu} M_j$ is read: “Model $M_i$ is equal in uncertainty or more uncertain than model $M_j$,” or “Model $M_j$ is no less certain than model $M_i$”.

A generality relation between P-models is denoted as $\geq_{Mg}$ and relation $M_i \geq_{Mg} M_j$ is read: “Model $M_j$ is a specialization of the measure $M_i$,” or “Model $M_i$ is a generalization of the measure $M_j$”.

A simplicity relation between P-model is denoted as $\geq_{Ms}$ and relation $M_i \geq_{Ms} M_j$ is read: “Model $M_i$ is equal in simplicity of simpler than model $M_j$”.

For MFT models that are represented as a system of axioms the generality relation can be defined as follows.

Definition. $T_i \geq_{gen} T_j$ if and only if $T_i \subset T_j$, i.e., system of axioms $T_i$ is equal to or more general than the system of axioms $T_j$ if and only if $T_i$ contains less axioms than $T_j$.

B. Similarity measures: uncertainty, generality and simplicity relations

Concepts introduced below can be applied to both logic and MFT similarity measures.

An uncertainty relation between similarity measures is denoted as $\geq_{Lu}$ and relation $L_i \geq_{Lu} L_j$ is read: “Measure $L_i$ is equal to in uncertainty or more uncertain than measure $L_j$.” This is a partial order relation.

A generality relation between similarity measures is denoted as $\geq_{Lg}$ and relation $L_i \geq_{Lg} L_j$ is read: “Measure $L_i$ is a specialization of measure $L_j$ and measure $L_i$ is a generalization of measure $L_j.” This relation also is a partial order.

A simplicity relation between similarity measures is denoted as $\geq_{Ls}$ and relation $L_i \geq_{Ls} L_j$ is read: “Measure $L_i$ is equal in simplicity or simpler than measure $L_j”. This relation also is a partial order.

Definition. Mapping $F$ between a set of P-models $\{M\}$ and a set of similarity measures $\{L\}$

$F: \{M\} \rightarrow \{L\}$

is called a match mapping if $F$ preserves uncertainty, generality and simplicity relations between models and measures in the form of homomorphism from a relational system $<\{M\}, \geq_{Mg}, \geq_{Mu}>$ to a relational system $<\{L\}, \geq_{Lg}, \geq_{Lu}, \geq_{Ls}>$, i.e.,

$\forall M_a, M_b ( M_a \geq_{Mu} M_b \Rightarrow F(M_a) \geq_{Lu} F(M_b))$,

$\forall M_a, M_b ( M_a \geq_{Ms} M_b \Rightarrow F(M_a) \geq_{Lu} F(M_b))$.

Two different models can be at the same level of uncertainty ($M_1 \geq_{Lu} M_2$), one model can be more uncertain than another one ($M_1 >_{Lu} M_2$), or these models can be incomparable for uncertainty. Thus, we have a partial order of models.

C. Parameterization

Below we parameterize uncertainty and generality of topology and geometry models of spatial connections. Consider an example of models with increasing levels of uncertainty starting from zero uncertainty at level 0:

Level 0: models $M_0$ with all network nodes and connections are known.

Level 1: models $M_1$ with one unknown connection, $c_1$.

Level 2: models $M_2$ with two unknown connections, $c_1, c_2$.

If $c_1 = c_2$ in $M_2$ then models $M_0, M_1$, and $M_2$ form a chain from a more specific and certain model $M_0$ at level 0 to a less specific and certain model $M_2$ (level 2). This is both a certainty and generality chain. Here $M_2$ is more general than $M_1$ and $M_0$. Note that if both $c_1$ and $c_2$ are not equal to $c_1$ then $M_0, M_1$, and $M_2$ form a certainty chain only, but do not form a generality chain. In this case, $M_2$ is not a generalization of $M_1$ and $M_2$ anymore. We cannot get $M_1$ by specializing $M_2$, because uncertain connections are different. Thus, we see that uncertainty and generality relations are different. We considered only 3 levels. In general, we assume n levels.

Another example provides to us the models at three uncertainty levels:

Level 0: models $M_0$ with all network nodes and connections are known.

Level 1: models $M_1$ with one unknown node, $e$.
Level 2: models $M_2$ with two unknown nodes $e_j, e_k$.
Here similarly the chain of uncertainty can differ from the chain of generalization.

Assume that the network has 4 nodes, $e_1, e_2, e_3$, and $e_4$. Each considered model has four uncertainty parameters, $p_1, p_2, p_3,$ and $p_4$. If a node $e_i$ is uncertain then $p_i=0$. For certain model $M_0$ we have $p_1 = p_2 = p_3 = p_4 = 1$, which is represented by a Boolean vector $(1111)$. Similarly, for $M_1$ we may have $(0111)$, if only node $e_1$ is uncertain, and $(0011)$ for $M_2$ if nodes $e_1$ and $e_2$ are uncertain. Thus, each model is represented as a Boolean vector, $v_i=(v_{i1}, v_{i2}, \ldots, v_{in})$: $M_2: v_2=(0011)$; $M_1: v_1=(0111)$; $M_0: v_0=(0011)$.

Similarly, uncertainty and generality order can be introduced for geometric characteristics such as node location:

Level 0: models $M_0$ with all network node locations and connections are known.

Level 1: models $M_1$ with unknown location of one node, $e_i$.

Level 2: models $M_2$ with unknown location of two nodes $e_j, e_k$.

In the same way, we introduce uncertainty and generality for direction/orientation of the node connections.

Level 0: models $M_0$ with all network node connections and directions of connections are known.

Level 1: models $M_1$ with one unknown direction of connections between nodes $(e_i, e_j)$

Level 2: models $M_2$ with two unknown directions of connections between nodes $(e_i, e_j)$ and $(e_k, e_l)$.

For shape, the first three uncertainty levels are:

Level 0: models $M_0$ with all shapes of objects associated with the network are known.

Level 1: models $M_1$ with unknown shape of one object $o_i$.

Level 2: models $M_2$ with unknown shape of two objects $o_i$ and $o_j$.

Definition. Parametric model $M_i$ is no less general than model $M_j$ if $v_i \geq v_j$, i.e., $\forall k \in \mathbb{N}$ $v_{ik} \geq v_{jk}$.

In accordance with this definition we have $1111 \geq 1110 \geq 1100 \geq 1000 \geq 0000$, that is isomorphic to $M_4 \geq M_3 \geq M_2 \geq M_1 \geq M_0$. Now we apply learning operator $C(M_i, E)$ and produce a chain of models, where each model $M_{j+1}$ is more specific than model $M_j$ with decreasing parameters $p_{ij}$.

$$M_n \geq M_{n-1} \geq \ldots \geq M_{j+1} \geq M_j \geq M_{j-1} \geq \ldots \geq M_2 \geq M_1 \geq M_0.$$  \hspace{1cm} (1)

A more detailed uncertainty parameterization can be developed if Boolean vectors are substituted by k-valued vectors $u_i=(u_{i1}, u_{i2}, \ldots, u_{ik}, \ldots, u_{in})$ with $u_{ij} \in U=\{0,1/(k-1),2/(k-1),\ldots,k-2/(k-1),1\}$ and as similar definition of model generality order based on $u_i \geq u_j$.

Above we encoded known parameters as 1 and unknown as 0. Now we can assign a level of parameter uncertainty $u_{ij}$ by computing $p_{ij}(M) / p_{2max}$ and assigning $u_{ij}$ as a nearest number from $\{0,1/(k-1),2/(k-1),\ldots,k-2/(k-1),1\}$.

D. Uncertainty and generality orders for spatial crossing models

Above we considered separately uncertainty of the models when nodes are uncertain and when connections are uncertain. Now we consider a situation when both take place. In Figure 1, the upper row contains a fully certain 4-crossing model, where all four connections (links) take place. The row directly below it shows all four less certain models situations where 3 out 4 possible connections to the crossing point are certain.

The next row shows all models where only two out of four possible connections to the crossing point take place. These models are shown in the same color. The forth row shows the models where only one out of four possible connections to the crossing point are certain. The last row shows the most uncertain crossing model, where all connections (link) are uncertain and even the existence of the crossing point is uncertain.

We assume that the third row covers models such as shown in Figure 2(a), where one of the lines goes through the crossing point. It is considered as the case shown in Figure 2(b).

The complete 16-node lattice is parameterized as a 4-dimensional binary cube, $E^4$, as shown in Figure 1. Thus, we have the uncertainty sequence of models here:

$$M_3 \geq M_2 \geq M_1 \geq M_0,$$  \hspace{1cm} (2)

where $M_0$ is a model from the top row and $M_3$ is a model from the bottom row.

Figure 1. Uncertainty hierarchy of crossing models that use both geometry and topology

Figure 2. Models with a line that crosses the crossing point.
While any models from these levels satisfy (1) for uncertainty relation, but it is not true for the generality relation. We have both

\[ 0000 \geq \mu_0 0100 \geq \mu_1 1100 \geq \mu_2 1101 \geq \mu_3 1111 \]

and

\[ 0000 \geq \mu_0 0100 \geq \mu_1 1100 \geq \mu_2 1101 \geq \mu_3 1111 \]

But for generality we have only

\[ 0000 \geq \mu_0 0100 \geq \mu_1 1010 \geq \mu_2 1101 \geq \mu_3 1111 \]

and have no 0100 \mu_0 1010 \geq 1101.

The lattice in Figure 1 is not planar, but it can be represented as a set of planar graphs. Each planar graph contains a specific set of non-overlapping chains of nodes of \( E^4 \). Together these graphs completely cover all nodes of this lattice. One of sets of chains in 4-dimensional binary cube, \( E^4 \) is called Hansel chains. This set of chains has several advantages that can be used to build a planar graph [Kovalerchuk, Perlovsky, 2008].

**E. Similarity measures**

Having any two models \( M_i \) and \( M_j \) that are shown in Figure 1 we can compute the similarity measure between them as a Hamming distance \( H(v_i, v_j) \) between 4-D binary vectors \( v_i \) and \( v_j \) that encode \( M_i \) and \( M_j \) respectively. Specifically if \( M_i \) and \( M_j \) are from the same level 2 of uncertainty, say, \( v_i = (0011) \) and \( v_j = (1100) \) then \( H(v_i, v_j) = 4 \). This Hamming distance is a similarity measure that is matched in uncertainty to these models. At first glance, it seem that \( H(v_i, v_j) \) for level 2 does not differ from \( H \) for \( v_i \) and \( v_j \) that are from different levels. It may not differ computationally but conceptually it is very different. We will denote it as \( H_2(v_i, v_j) \) to identify the level explicitly. Moreover, use \( H_2 \) in the situation where \( v_i \) corresponds to model \( M_i \), but \( v_j \) corresponds to data \( E_j \) which are at the level 2 uncertainty too. This gives us a measure of the similarity between a model and a data.

Assume that we have several models \( M \) at level 2 and several datasets \( E \) that are at level 2 or more certain. We can convert all these dataset to level 2, i.e., we can ignore the additional information known about these data. In this way the space of search alternatives will shrink. If each dataset contains few crossings then we have no computational challenges, but having a dataset with thousands of spatially distributed crossings, we face very significant computational challenges. Thus, using the dynamic logic approach we may find faster a “best” model (a model that is closest in the similarity measure \( H \) to the generalized data).

Having \( H \) we can associate, mine and fuse data using many data mining and fusion algorithms that are based on distances, such as k nearest neighbors (k-NN) algorithm. This is applicable to spatial data geo-registration, discover track pattern, matching and fusing tracks, correlating measurements with tracks and tracklets. This includes ambiguous measurements that can be associated with different and densely located tracks. This ambiguity created potential uncertain “crossings” such as shown in Figure 1. We may have two measurements \( m_1(t) \) and \( m_2(t) \) at moment \( t \) and two new measurements \( s(t+1) \) and \( q(t+1) \) at moment \( t+1 \). Figure 3(a) shows a case of parallel tracks and 3(b) shows a track “crossing” case (model). The “crossing” case does not necessary mean actual geometric crossing of lines in 3-D.

The uncertainty of this situation can be represented and parameterized as follows. The case 3(b) is similar to Figure 1, where for more measurements are assumed for its tracking interpretation (each line is a set of consecutive measurements).

![Figure 3. Certain ::-models parameterized as (11)](image)

The joint model for cases 3(a) and 3(b) can be represented as a class of model that we denote as ::-model. Figure 3 shows fully certain versions of ::-model. Some less certain versions are shown in Figure 4. The most uncertain version of ::-model is the case when no node has connection. Different parameterizations are possible for ::-model. The one that does not distinguish between 3(a) and 3(b) will encode them as \( (v_1, v_2) = (11) \), where \( v_1 = 1 \) if link of \( m_1 \) is certain and \( v_2 = 1 \) if link of \( m_2 \) is certain. Thus the most uncertain case of ::-model is parameterized (00). The similarity between such parameterized models can be computed again by using Hamming distance, \( H \).

![Figure 4. Uncertain ::-models parameterized as (01) and (10)](image)

**F. Similarity Maximization**

A similarity maximization problem is a major mechanism of the dynamic logic that is formalized below.

**Definition.** A similarity \( L_{\text{fin}} \) measure is called a final similarity measure if \( \forall M, E, L_i, L_i(M, E) \geq L_{\text{fin}}(M, E) \)

The final similarity measure sets up the level of certainty of model similarity to the data that we want to reach.

**Definition.** The static model optimization problem (SMOP) is to find a model \( M_a \) such that

\[
L_{\text{fin}}(M_a, E) = \max_{M_i} L_{\text{fin}}(M_i, E)
\]

subject to conditions (2) and (3):

\[
\forall M_j \quad L_{\text{fin}}(M_a, E) = L_{\text{fin}}(M_j, E) \Rightarrow M_a \geq_{\text{fin}} M_j \quad \text{(2)}
\]

\[
\forall M_j \quad ((L_{\text{fin}}(M_a, E) = L_{\text{fin}}(M_j, E) \land (M_j \geq_{\text{fin}} M_a)) \lor \quad (M_j \geq_{\text{fin}} M_a)) \Rightarrow M_a \geq_{\text{fin}} M_j \quad \text{(3)}
\]

Conditions (2) and (3) prevent model overfitting and beneficial computationally if further specification of the model requires more computations. Condition (2) means that if \( M_a \) and \( M_j \) have the same similarity measure with \( E \), then
uncertainty of \( M_\pi \) should be no less than uncertainty of \( M_j \). Condition (3) means that if \( M_\pi \) and \( M_j \) have the same similarity measure with \( E \) and \( M_k \) and \( M_j \) are comparable relative to generality relation \( \geq M_k \), then \( M_\pi \) should be no less general than \( M_k \). \( M_\pi \) can be obtained by specification of \( M_k \).

**Definition.** The dynamic logic model optimization (DLPO) problem is to find a model \( M_\pi \) such that
\[
L_d(M_\pi, E) = \max_{i \in I} L_\pi(M_\pi, E) 
\]
subject to conditions (3) and (4):
\[
\forall M_j \quad L_d(M_\pi, E) = L_j(M_j, E) \Rightarrow M_\pi \geq M_\pi M_j , \quad (5)
\]
\[
\forall M_j \quad ((L_{fin}(M_\pi, E) = L_j(M_j, E) \& ((M_j \geq M_\pi M_\pi) \lor ((M_\pi \geq M_j M_j))) \Rightarrow M_\pi \geq M_\pi M_j \quad (6)
\]

This is a non-standard optimization problem. In the standard one, only models \( M_i \) are changed but the optimization criterion \( L \) is not. It does not depend on the model \( M_i \). In DL, the criterion \( L \) is changing dynamically with models \( M_i \).

**Definition.** Mapping \( C: \{M\} \rightarrow \{M\} \) is called a **learning (adaptation) operator** \( C \), \( C(M, E) = M_{i+1} \), where \( E \) are data and \( M_{i+1} \geq M_{i+1} \).

This operation represents a **cognitive learning process** \( C \) of a new model \( M_{i+1} \) from a given model \( M_i \) and data \( E \). In other words, it is an adaptation of model \( M_i \) to data \( E \) that produce model \( M_{i+1} \).

It is advantageous to get a simple and quite universal operator \( C \) relative to a brute force algorithm that computes \( L(M,E) \) for every \( P \)-model \( M \) and selects \( M \) that provides \( \min L(M,E) \). Consider a direct modification of brute force algorithm to the situation with dynamic change of correspondence (similarity) measures \( L_i \), which is assumed in the dynamic logic. The assumption is that the space of highly uncertain models is relatively small. Thus, a brute force algorithm can work for these models in a reasonable time. Next, the best model \( M_i \) found at this step will produce a new correspondence measure \( L_i \) and this measure will be applied to a **new set of models** produced by \( M_i \) for evaluation (e.g., with a more dense grid around \( M_i \) or in another location if \( L(M_i) \) is low). To produce new models we use a new operator, \( H \), which is called a **specialization operator**, as follows:

- **Step 1.** Select initial \( P \)-model \( M_{00} \).
- **Step 2.** Produce set of \( P \)-models \( H(M_0) \).
- **Step 3.** Compute \( L_d(M, E) \) for every \( M \) from \( H(M_0) = \{M\} \) and find model \( M_1 = \arg \min_{M} L_d(M, E) \).
- **Step 4.** Test if \( L_d(M_1, E) > T \), i.e., is above the needed correspondence threshold. Stop it if true, else go to step 5.
- **Step 5.** Repeat steps 1-4 until all models tested or time limit reached.

A more complex strategies for \( H \) operator can be based on breadth first, depth first, and branch-and-bound strategies and the use of the theory of monotone Boolean functions [Kovalerchuk, Perlovsky, 2008]

### III. Generalization of Dynamic Logic for Data Fusion and Mining of Heterogeneous Data

#### 1. Correlating spatial geometrically and topologically heterogeneous data

For data fusion and mining, we need to correlate disparate data with significantly different geometric structures, topologies, spatial distribution (e.g., complex graphs of social relations) that were not possible to correlate previously. Typically, classical distance-based similarity measures \( D(a,b) \) and classical statistical correlation methods correlate objects \( a \) and \( b \) directly in one step without using one or more intermediate objects. We generalize these classical approaches to disparate structured qualitative data where one-step correlation is too simplistic to capture the richness of structured objects such as social or sensor networks and cyber-physical systems in general.
The intermediate object (model) $w$ may have geometric similarity with $a$ and topological similarity $D_T(w, b)$ with $b$. The sequence of intermediate objects $w_1, w_2, \ldots, w_n$ allows us to correlate objects $a$ and $b$. It also allows us to build a new representation for $a$ and $b$. Assume that $\{w_i\}$ is a set of objects with a standard representation, and $f_a$ is a correlation transform of $w_i$ to $a$, $f_a(w_i) = a$. Then representation for $a$ is a pair $(w_i, f_a)$. Similarly if $f_b(w_i) = b$ then a pair $(w_i, f_b)$ represents $b$. Here we may allow an approximate equality instead of the exact one if needed.

In this representation $D_G(a, w)$ captures geometric correlation (similar positions & directions), and $D_T(w, b)$ captures topological correlation (see Figure 5). We call the algorithm to find $D_G(a, w)$ and $D_T(w, b)$ the Intermediate Geometry and Topology correlation (IGT-correlation).

Context 1: Incomplete geometry.

Step 1: Expand linear segments of object $a$ to the crossing points with other lines. This produces object $w$. In this case, the geometric correlation/similarity between $a$ and $w$, means that $a$ is a part of $w$.

Step 2: Correlate $w$ and $b$ topologically. They are identical topologically.

![Figure 5. Correlation with an intermediate object](image)

For more difficult cases, the algorithm selects segments to be expanded. It is an iterative process: first to connect some nodes to make a fully connected graph (network) as $w_i$ candidate then to check if it can be correlated with the topological structure $b$.

Context 2: Over complete geometry (some connections are erroneous, but we do not know which ones.) In this context, the algorithm cuts some links to produce a new topology.

$J$. Representation of Dynamic Topology for Data Fusion and Mining

Consider a situation shown in Figure 6. Case (a) shows waterways at the high water level and case (b) shows waterways at a low water level. Topology, geometry and sets of routes are different in (a) and (b). The dotted route in (a) is not available in (b).

![Figure 6. Example of dynamic geometry and topology](image)

The graph (topological representation) of (a) and (b) is shown in Figure 7(a) and 7(b) respectively. This representation can be encoded in the standard incidence matraces. Table 1 shows this matrix for case (b).

The geometric representation of the dotted track line shown in figure 6(a) is a set of points of this line. This representation is not efficient for data mining and fusion. It contains too many points and it is too specific. Recordings of the same track by different sensors will differ due to different sensor modalities, viewing geometry, sampling rate, accuracy etc.

![Figure 7. Spatial data representation](image)

Table 1. Incidence matrix for graph in Figure 7(a).

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>e</th>
<th>h</th>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>h</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>j</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

We build a sequence of representations that are intermediate between these two, and which can be used in data fusion and mining. Thus, the fusion mining schema is as follows:

(0). $T$: topology only setting. All geometry data are converted to topology by finding points A, B, E, H, J on the track as shown in figure 6(a).

(1). $TG_1$: Topology and geometry level 1: The geometry is preserved only partially by adding intermediate points C, D, F and G (see Figure 8(b)). These points are selected by algorithm EAD [Kovalerchuk, 2007] that provides them several desired properties such full affine invariance and robustness to noise in T-vicinity.
TG2: Topology and Geometry level 2: The geometry is preserved more completely relative to FG1 level by adding more intermediate points between points C, D, F and G using EAD algorithms that was used in TG1.

TG3: Topology and geometry level 2: TG3 is produced from TG2 in the same way. We call this process Binary Sequential Division (BSD).

(i) TGi: Topology and geometry level i. TGi is produced from TG(i-1) in the same way as was shown above by using the BSD process.

We record and represent each TGi level as an incidence matrix or as a set of binary relations (predicates) \( \{Q(x,y)\} \). These representations are equivalent, but \( \{Q(x,y)\} \) allows to use relational data mining techniques [Raedt 2006, Mitchell 1997, Kovalerchuk, Vityaev, 2000] directly without any additional representational effort. The incidence matrix shown in Table 1 needs to be converted to two standard binary relations \( Q_1(x,y) \) and \( Q_2(x,y) \) to avoid value 2 in this table that reflects two edges between nodes such as B and C. For instance, \( Q_1(x,y) \) can cover an upper edge that connects B and C, and \( Q_2(x,y) \) can cover a lower edge that connects B and C. In our example, the incidence matrix representation differs from the binary predicate representation only for level T (topology only).

Situation 1: All tracks are available at the same TGi level. This means that we can use relational data mining algorithms directly (e.g., FOIL or MMDR [Kovalerchuk, Vityaev, 2000]).

Situation 2: Tracks are available at different TGi levels, say from T to TG1 as shown in Figure 6. This means that the additional representational effort is needed to use relational data mining algorithms to be able to use all richness of available data without commonly used downgrading data.

The only requirement for this is to preserve point naming as we did in Figure 8(a) and 8(b). Thus we get predicates \( Q_1(x,y) \) and \( Q_2(x,y) \) from T level and predicate \( Q_3(x,y) \) from TG1 level. These predicates \( Q_3 \) serves further as constrains for discovering rules by relational data mining algorithm MMDR [Kovalerchuk, Vityaev, 2000, 2008].

L. TG Data Fusion Example

Consider a road conflation task to match and conflate green roads and roads extracted from the image (blue) in Figure 7. Green roads should be transformed and aligned with the image using the extracted blue roads. The standard topological match of the road crossings does not work here due to node mismatch, gaps in connections and differences in road coverage.

To solve this task we consider a set of intersection models:

- X-models are connections of 4 lines (Figure 1).
- T-models are connections of 3 lines (T-intersection).
- L-models are connections of 2 lines (L-intersection).
- I-models are “connections” of a single line (I-model represents a gap or a node on single line).
- O-models are terminal nodes of the line that are not connected to other lines.
- D-models are disconnected nodes.

Yellow circles in Figure 8 show types of road connections from the list above. We call these data respectively, X-data, T-data, L-data, and I-data. These data are at different levels of uncertainty with missing connections, gaps and incorrect locations due to imperfect feature extraction and multiple other sources of errors.

The idea of the algorithm is first matching data to models at the highest level of uncertainty and then specifying models and data.

The outline of the algorithm for Figures 9-11:

1. Identify all connections in green and blue lines independently of their types and levels of uncertainty, which is at the highest level of uncertainty.
2. Build a network for green lines using yellow circles as its 9 nodes.
3. Build a network for blue lines using blue circles as nodes. This network contains five nodes, where one node has no connections.
4. Identify types (X,T,L,I) of yellow and blue nodes and label each node with its type.
5. Identify nodes to match according to types (Figure 11).
6. Measure orientation similarity of nodes of the same types.
7. Match nodes using information from 1-6.
In Figure 11 the blue nodes are numbered in italic. Yellow lines contain two O-nodes (#1,#9) and seven =T-nodes (#2-, #8). Blue lines contain one D-node (#1), two L-nodes (#2,#5), and two T-nodes (#3,#4).

The correct matches are: 2 → 2; 2→3; 3→4 and 4→ 5, that is yellow node 2 is matched to two nodes. Such complex match requires first to assume a rich model that allows one-to-many match.

To avoid considering all possible such models, e.g., with mapping 3 →3; 3→4 we exploit this topology in combination with geometric structures of lines that connect nodes and their matrix representation [Kovalerchuk, Schwing, 2005]. We had shown that this combination is computationally efficient.

V. CONCLUSION AND FUTURE WORK

This paper presents a generalization of the Neural Modeling Fields theory and the Phenomena Dynamic Logic for heterogeneous spatial data in cyber-physical space. This generalization is based on the concept of a set of intermediate objects (models) to be able to link and represent uncertain geometry and topology of spatial objects. From our viewpoint, the most interesting and useful future research in the Neural Modeling Theory and PDL is discovering an adaptive learning mechanism of the changing models and similarity measures. Humans demonstrated capabilities to switch evaluation criteria and similarity measures instantaneously in dynamic environment. This is the area where logic, mathematical modeling and cognitive science can provide mutual benefits in discovering mechanisms of the changing models and similarity measures.

VI REFERENCES