Conceptual Modeling of Events Based on One-Category Ontology

Sabah Al-Fedaghi

Computer Engineering Department, Kuwait University, Kuwait

Summary
In previous works, we have proposed a one-category (called thimac) conceptual model called a *thinging machine* (TM), which integrates staticity (e.g., objects) and dynamism (e.g., events) without losing valuable aspects of diagrammatic intuition in conceptual modeling. We proposed applying TM in conceptual modeling in software engineering (e.g., on or above the level of UML as a conceptual modeling language). In this paper, to show such an application in software engineering, we first present a complete high-level description of a library services system to demonstrate the TM's applicability. Furthermore, we explore the TM's features, emphasizing the realization of thimacs as *events*. The purpose is to develop better understanding of the TM notions by contrasting them with their uses in related fields. The notion of an event plays a prominent role in many fields of study, including philosophy, linguistics, literary theory, probability theory, artificial intelligence, physics, and history. A TM event is *a static thimac that has a time breath (time subthimac)* that infuses dynamism into the thimac. It arises from how the TM static region is infected with time. Such a view is contrasted with some philosophical and linguistics definitions of an event (e.g., unit of experience – Whitehead). We also raise interesting issues (e.g., event movement) in this study.

Keywords: *Conceptual modeling, events, change, thinging machine model, events*

1. Introduction

Software engineering, because it must often interact with the outside world, needs tools to develop semantic models prior to software implementation. Conceptual modeling entails developing an abstract model with appropriate simplification of reality [1]. The model involves explanation of the real system, which is capable of producing all possible input–output behaviors and integrating various components of a system to be refined into a more concrete executable model. In short, the conceptual model defines what is to be represented and how it is to be represented [1]. Two types of conceptual models are typically identified: a domain-oriented model that provides a detailed representation of the domain and a design-oriented model that describes in detail the model's requirements [1]. This paper focuses mainly on the domain-oriented conceptual modeling.

It is claimed that modeling is more of an "art" than "science" [2]; therefore, according to Karagöz [3],

> it is generally assumed defining methodical ways to develop conceptual models is difficult. The evolution of newer engineering fields, such as systems and software engineering, has shown that using well-defined modeling notations, following defined processes, and utilizing software tools definitely improve effectiveness. [3]

1.1 Sample Current Approach

For example, over the last twenty years, many studies have promoted the utilization of object-oriented language UML for conceptual modeling [3] [4]. According to Breiner et al [4], UML has value only in a software project's early stages and is to be discarded in the project's later stages through implementation and testing. Additionally, according to Breiner [4], the UML approach suffers from the difficulty in learning and applying 14 types of diagrams and the problem of ensuring consistency across the diagrams [4]. Breiner et al [4] also claimed that

UML diagrams arose from a variety of needs and applications, and were not designed to work together. Its wide variety of constructions overlap, so that it is often unclear what type of model should be used to capture a particular observation. The underlying semantics for UML modeling was an afterthought, defined after the fact and rarely called on in practice.

As a possible solution, Breiner et al [4] proposed diagrammatic models that "look very similar to UML class diagrams" and are grounded in the mathematical category theory.

1.2 Alternative Conceptual Constructs

Conceptual modeling attempts to model a system based on *concepts*. Developing such models touches both the psychological and abstraction realms. This effort requires supplying unambiguous categorization with elements of

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.54

discreteness and hierarchically ordered representations of the modeled domain. Conceptual constructs' meanings have to be defined carefully using ontology to analyze and enrich the capacity to capture knowledge about an application domain. Often, however, rigorous definitions of these constructs are missing.

Categorization is the elementary task for the construction of our understanding of the world, through which ideas are recognized, differentiated, and understood. Categorization represents the "the most basic phenomenon of cognition" [5]. Historically, categorization has been intended to enumerate everything that can be expressed without composition or structure [5]. The representation of the modeled domain is formed utilizing natural language and diagrams with semantics determined by mental and non-mental factors. Conceptual representation may also require inviting and/or uncovering new categorizations.

Conceptual categories are fundamental to the mapping between a model and its domain. Bradley and Bailey [6] commented that it is hard to say much about a category under which *every thing* falls; nonetheless, candidates are available for such a category, including *thing*, *entity*, and—especially—*object*. According to Sinha and Gärdenfors [7], we are naively accustomed to thinking of objects as the most fundamental ontological category of the physical world.

One difficulty that is related to the notion of "object" is that the variety of the world seems to lie not only in the assortment of its objects "but also in the sort of things that happen to or are performed by them" [8]. There would seem to be a difference in mode of being between objects and *events* that are said to occur, happen, or take place [8]. Additionally, objects have relatively crisp spatial boundaries and vague temporal boundaries, and events have vague spatial boundaries and crisp temporal boundaries.

According to Casati et al. [8], some philosophers would simply deny the conceptual distinction between events and objects and would simply treat the distinction as one of degree: a thing is "a monotonous event; an event is an unstable thing." Some philosophers claim that although both objects and events are featured as "the basic units from which to build a descriptive system," the primacy of objects is strongly supported by phenomenological considerations (see sources in Casati et al. [8]).

1.3 Objective: Advocating Thimacs as a Base for Conceptual Modeling

In this paper, we claim that the subtle difference between objects' and events' fundamental conceptual construct plays an important role in constructing conceptual models. In previous works (e.g., [9]), we proposed a one-category conceptual model (called a *thinging machine* [TM]) that integrates staticity (e.g., objects) and dynamism (e.g., events) without losing valuable aspects of diagrammatic intuition in conceptual modeling. We applied a TM in conceptual modeling in software engineering (e.g., on or above the level of UML as a conceptual modeling language).

In this paper, we explore further the thimac notions, emphasizing the realization of *thimacs as events*. The purpose is to advocate this approach by developing a better understanding of the TM notions by contrasting them with their uses in related fields. As we will argue, a TM event is *a static thimac that has a time breath (time subthimac)* that infuses dynamism in the thimac. The event arises from how the TM static region is infected with time. We contrast such a view with some philosophical and linguistic definitions of an event (unit of experience – Whitehead [10]).

2. TM Modeling

The TM model is a conceptualization of how things/machines can be merged into a complex of interrelated thimacs (i.e., things that are simultaneously machines). A thimac is a thing that can be created, processed (changed), release, transferred, and/or received. It is also a machine that creates, processes, releases, transfers, and/or receives. Fig. 1 shows a general picture of a TM. The figure indicates a "field" with five "seeds" of potentialities of dynamism. Aristotle identifies matter with potentiality, e.g., the wood, as the matter, has the potential statue. FM contains five potentialities of action:

- The appearance (coming into existence in the system) of a new thing (create)
- The variation (change) in the same thing (called process)
- The movement from one field to another (release, transfer, and receive)

Appearance is the phenomenon of becoming or "existing" within the system, *variation* is a change in the same thing, and *movement* occurs among machines. They are "seeds" of potential actions.

A TM may be viewed as a thing when considered as a whole; therefore, the thing may flow to yet another machine, as Fig. 2 shows.

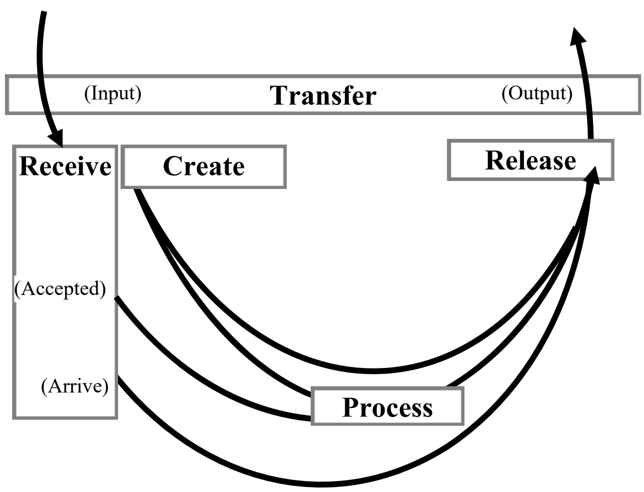


Fig. 1 A thinging machine.

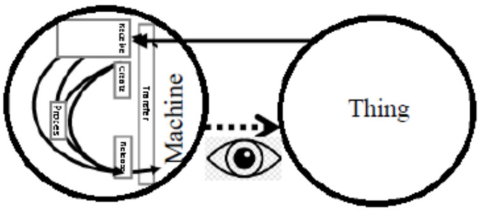


Fig. 2 A machine may be viewed as a thing that flows to a machine.

2.1 Preliminary Notes of the Notion of Action Potentiality

Initially, an *action* is a state of readiness to become an *event*. Prior to an event's emergence, an action in its potentiality state is part of a static description. An action as part of a static state (which seems at first contradictory) is just a static object in the common meaning. What moves (changes) an object that has the capacity of motion (change) is the *event* that emerges from the (potential) *action.* and *time* "possess" (as in spirit possession) each other. Therefore, a verb in a sentence does not refer to an (actual) activity unless the sentence represents an event. Natural language ambiguity, usually, blurs such a difference between potential action and actual action (event).

Returning to conceptualizing the notions of action and event, we find that, for Aristotle, an action is a type of event "with an inherent end" [11]. For Aristotle, an action is a "potentiality" (a capacity for action), and time ignites "actuality" (a type of event that means the existence of the thing [11]). Note that Aristotle did not explicitly include events in his categories [11]. Aristotle's event-related analysis is based on linguistic forms in which verbs are viewed as dynamic beings. This linguistic approach continued in various works. In recent times, this linguistic-based conceptualization can be seen in Davison's analysis of sentences searching for event structures [11].

TM is based on recognizing generic actions. TM "beings" (i.e., thimacs) have in themselves a principle of motion (generic actions). We take from Aristotle the notion of "potentiality"; that is, actions need a time element to exhibit dynamism. Rest is actualized by time change with no internal change. In a TM, actuality is the fulfillment of generic actions that potentially exist.

2.2 The Thimac

A thimac (thing/machine) forms an arena in which a potentiality acts on a thing that happens to be there according to its position in one of the five seeds of potential actions. It is a static field in the sense that there is no dynamism. Similar to the concept of "field" in the physical sciences, the TM field is a region in which potentiality acts on (is applied to) things but in five ways. When this static field is joined by a time field (thimac), each thing is stimulated according to its seed (create, process, release, transfer, and receive). We call this combination of static and time fields a *generic event*.

A TM may be viewed as a thing when considered as a whole; therefore, the thing may flow to yet another machine. Thimacs are created, and they create their subthimacs. Thimacs first have to be created so they can create, process, and move things. Any thimac, there should "exists" (in the TM diagram), so it, in turn, creates other thimacs. For simplicity's sake, we consider the presence of a box a sign that the thimac exists.

Thimacs are conceptual (mind-made) fields of seeds of potential actions developed to make sense of the world. A thimac exhibits sufficient "togetherness" to form a bounded whole of subthimacs. Therefore, a thimac is a generally mechanistic ontology in which we see a thing that is conceptualized as the mereological totalities of subthimacs.

All things are created, processed, and transported (acted on), and all machines (thimacs) create, process, and transport other things. Things "live" or "pass through" other machines. The thing is a presentation of any "existing" (appearing) entity that is able to be "counted as one" and coherent as a unity. A noun is usually used as a label for things and what we perceive and can identify, even if we have no words to name it.

Machines house other things and provide pathways for their flow. The unity of thing and machine forms a thimac. In such a blend, a single thimac is a fusion of two manifestations, flow and machines, for other flowing things. The actions in the machine are ordered in a specific way (Fig. 1). As Fig. 1 shows, a TM can be viewed as a coordinated system of flow. The flow is not a type of link (e.g., a class of relationship in ER); it is rather a transformation from one potentiality of action to another.

2.3 The TM

Fig. 1 can be described in terms of the following generic (has no more primitive action) actions:
Arrive: A thing moves to a machine.
Accept: A thing enters the machine. For simplification, we assume that all arriving things are accepted; hence, we can combine the arrive and accept stages into one stage: the **receive** stage.
Release: A thing is ready for transfer outside the machine.
Process: A thing is changed, but no new thing results.
Create: A new thing is born (being found/manifested) in the machine. It is realized from the moment a thing arises .

(emergence) in a thimac. Things come into being in the model by "being found." Creation in metaphysics involves bringing the entities from the state of nonbeing into existence. The TM model limits this creation to appearance in the model. *Create x* in a model means "there is" *x*. After the instance of creation, the entity may move to be processed or released, or it may stay in the creation state.
Transfer: A thing is input into or output from a machine.

Additionally, the TM model includes the mechanism of triggering (denoted by a dashed arrow in this study's figures), which initiates a flow from one machine to another. Multiple machines can interact with each other through the movement of things or through triggering. Triggering is a transformation from one series of movements to another.

3. Complete Example of TM Modeling

In this example, (from [12]) the librarian can list the library books, and from there, a book may be selected for addition or a new book may be created. Both of these use cases include a list of the related book copies (See Fig. 3). The librarian is also able to list the books, and he/she may select related authors. Fig. 4 shows the TM model of this subset of a librarian's use cases in a library system.

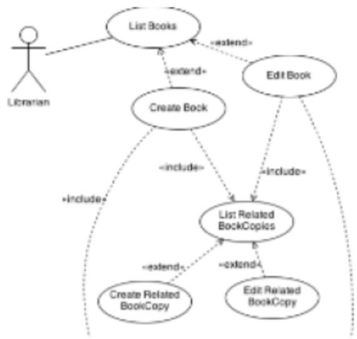


Fig. 3. The use case model of the library system (partial, from [12]).

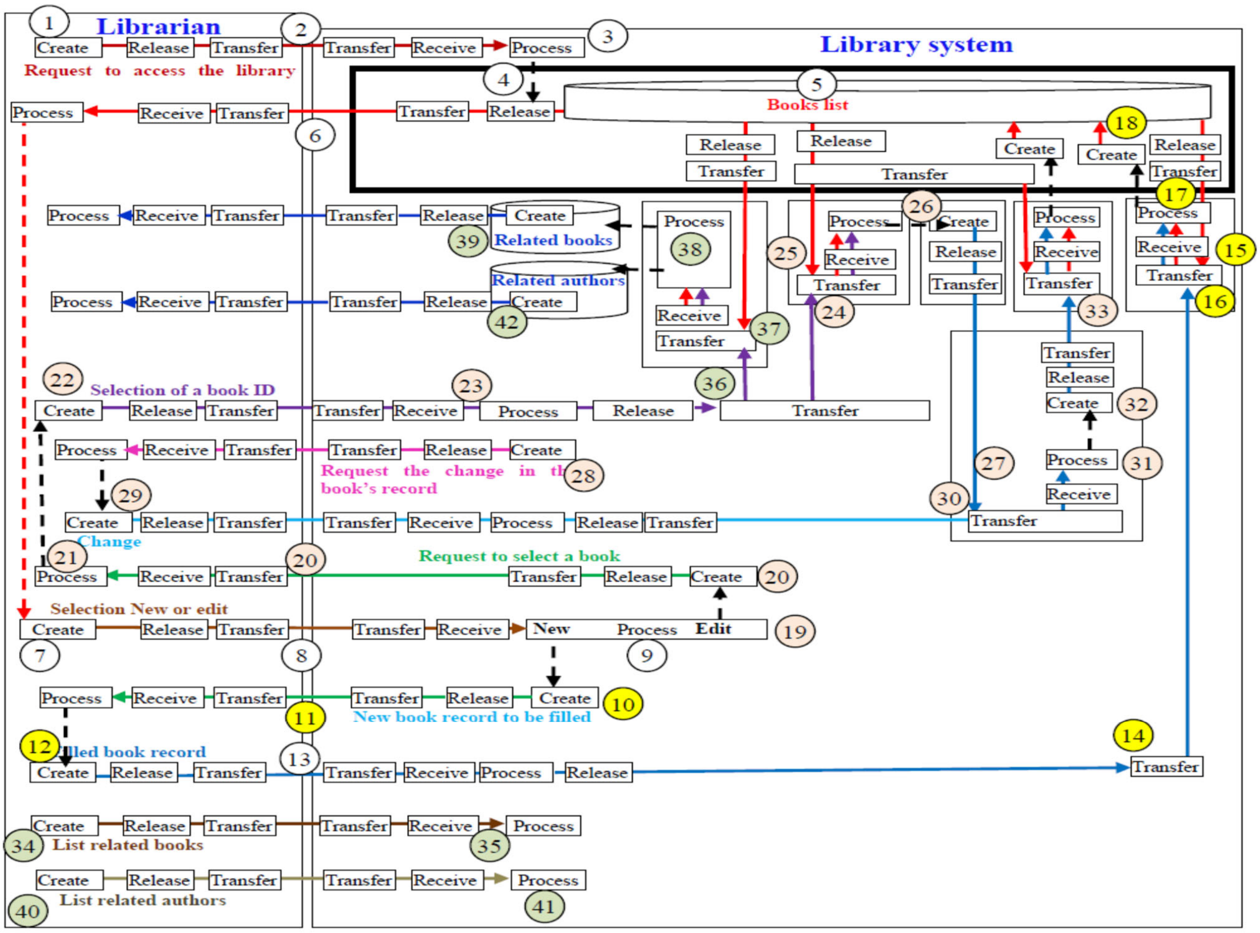


Fig. 4 The TM static model of the library system.

We will model this example in a TM with some modifications related to the general understanding of the case. For example, related books can be extracted from the main list of all books and not updated separately as Cruz [12] seems to indicate. We can also produce the related books and authors by inputting the book ID without tying such a list to new or updated books. Additionally, for simplification, we ignore the case of books copies. Some boxes are eliminated for simplification. There are two first-level machines: the librarian and the library system. The box with the broad oval at the top of the library system can be labeled a "books list" machine; however, we opt not to label it because its function is clear: the list of books is sent and updated (created) either to the librarian or to the library system.

3.1 Description of the TM Static Model

In Fig. 5,

- The librarian requests access to the library system (circle 1 in the figure), and such a request flows to the system (2), where it is processed (3). Note that all generic actions (create, release, transfer, receive, and process) are potentialities that reflect dynamism. The request is a thing, and its machine spread across the librarian and the system.
- We can consider the sequence of potentialities that extend between circles 1-3 the construct of the conceptual field of "the librarian accesses the library system."
- This is our understanding of this type of interaction between the librarian and the system. Of course, such an interaction needs data to be realized, but regardless of the type and size of data, the flow between circles 1 and 3 remains the same.
- Assuming that the process determines that the request is acceptable, it triggers (4) the download of the list of books (5) to the librarian (6). Additionally, with the list, the system prompts (dashed red line to the left of the figure) the librarian to determine whether he/she intends to add a new book or edit some data about an existing book (the downward vertical red dashed arrow in the librarian machine). Accordingly, the librarian makes (7) his/her selection, which goes (8) to the system, where it is processed (9).

If the librarian selects "new," the system creates a book record to be filled (10) and sends it to the librarian (11), who supplies data to create a new filled book record (12) and sends it to the system (13). The system sends (14) that record to be added to the list of books, which requires the retrieval of the current list (15) and the new record (16) to be processed (17) to create a new list (18).

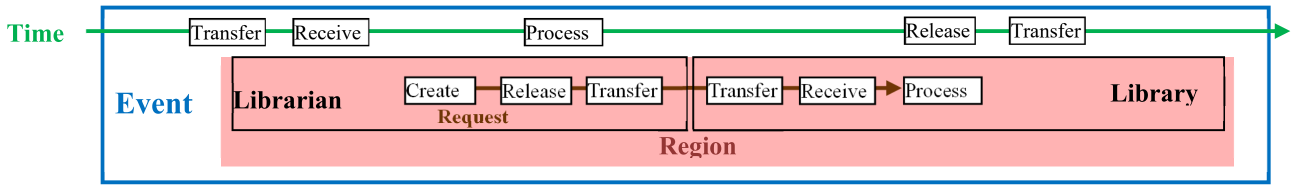


Fig. 5 The event of *The librarian requested a list of authors*.

-
- If the librarian chooses to edit a book record (19), then the system creates a request (20) to select a certain book (from the previously downloaded list) (6). The request flows to the librarian (21), who processes the request (20) and sends the book ID (22). The system receives the book ID (23) and sends it as a procedure (24) that compares it with IDs of records in the books list (25), thus finding the record of the input key (26). Note that this is indicated by a process that triggers the creation (appearance in the global view of the system) of the required record (26). The found record of the book flows to be edited (27).
- Editing the book's record requires (28) the librarian to input the changes (29 – e.g., change publication date). Hence, in the relevant procedure, the book record (28) and the change (30) are processed (31) to create a new record (32), which goes (33) to update the current list of books as before (18 and 19).
- Now we come to the part that is different from Cruz's [12] description due to a lack of a complete understanding on our part and to simplify the example. However, the TM model can be extended to accommodate any other parts that we do not cover. We assume that the librarian can list the books and authors for any given book separately from the operations of adding a new book or editing a current book. Therefore, (at 34 and 35, bottom left corner), the librarian requests a list of all books related to a certain book.
- He/she can identify the book as before (23 and 24). Hence, the book ID (23 and 24) is processed along with the current list of books (36 and 37) (38) to produce the related books (39). A similar procedure is followed to produce the list of relevant authors (40, 41, and 42).

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.54

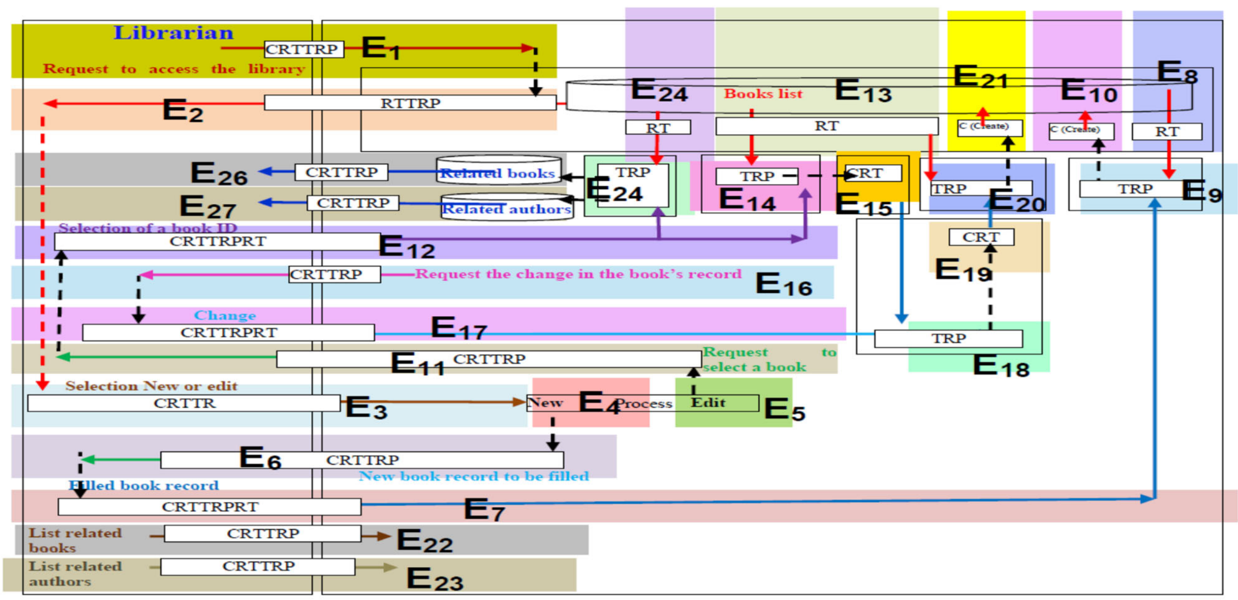
3.2 The Behavioral Model: Events

The TM's behavioral model is constructed as the chronology of events in the modeled system. An event is defined as a unity of a subdiagram of the TM static (standing still) model (called the region of the event) plus a time subthimac, which *activates* (conceptual space to come alive) the region. Time in this description is not a container; rather, it is the state of being dynamic (motion: flow of things), analogous to the life of a physical body. Motion is described as actuality of the unity mentioned above. The time thimac is always a subthimac of non-time thimacs.

For simplification purposes, the event may be represented by its region. As shown in Fig. 6, we identify the following events. The figure is simplified by denoting sequences of actions by their first letters. For example, CRTTRP denotes the sequence of actions: create, release, transfer, transfer, receive, and process.

E_1: The librarian requests to access the system.
E_2: The system downloads the books list, giving the librarian the choice of either starting a new book or editing a current book.
E_3: The librarian makes a selection from new/edit options.
E_4: The system processes the librarian's selection and recognizes the selection of a new book.
E_5: The system processes the librarian's selection and recognizes the selection of an edited book.
E_6: The system sends a request to fill the record of a new book.
E_7: The librarian supplies the new book data, and then the system receives the new book data and sends them to update the books list.

E_9: The books list and the new book record are processed to add the new book record.
E_{10}: The new books list replaces the old one as the latest list.
E_{11}: The system requests the edited book ID from the librarian.
E_{12}: The librarian supplies the book ID that is received by the system and used to retrieve the book record.
E_{13}: The book record is retrieved from the books list and then sent to the librarian to edit it.
E_{14}: The books list and the book ID are processed to retrieve the book record from the list.
E_{15}: The book record is extracted from the list and sent to the librarian to be edited. Note that C (create) in the event denotes the appearance of the book record as an independent entity.
E_{16}: The system requests the changes to be made for the book record.
E_{17}: The librarian sends the changes that are received by the system to be used to update the book record.



E_8: The current books list is downloaded to update it.

Fig. 6 The events model of the library system.

E_{18}: The book record and the changes are processed to update the record.
E_{19}: A new record that includes the changes is created and sent to the librarian to update the current list. Note that C (create) in the event denotes the appearance of the new version of the record as a new entity.
E_{20}: The books list and the new record are processed to replace the old version of the record.
E_{21}: A new version of the list is created that replaces the old version.
E_{22}: The librarian requests for a books list related to a given book.
E_{23}: The librarian requests for an author list related to a given author.
E_{24}: The books list is retrieved and sent to select relevant books and authors.
E_{25}: The books list is processed according to the given ID, and relevant books and authors are selected.
E_{26}: The relevant books are sent to the librarian.
E_{27}: The relevant authors are sent to the librarian.

Fig. 7 shows the behavioral model of the library model in terms of the chronology of events. The resultant static, dynamic, and behavioral models can be used to build information and control systems for the library services.

4. Illustration Potential Actions and Events in The TM

To illustrate the relationship among actions and events, assume that our domain includes just two things, X and Y, which are shown in Fig. 8. X is the machine, and Y is the thing that is created by Y. Fig. 8 (left) is a picture of a *static* situation of potential action. Fig. 8 (right) shows another *static* picture where Y is now in the *process* stage of X. The TM static diagram is the union of all possible static situations or, in other words, the sum of the regions of (conceptual) space specified as a TM diagram. During the flow of a thing, the thing may be in any stage in the TM diagram. In each situation, the stage (potential action) is part of the *static* situation. A *change* occurs when time is involved.

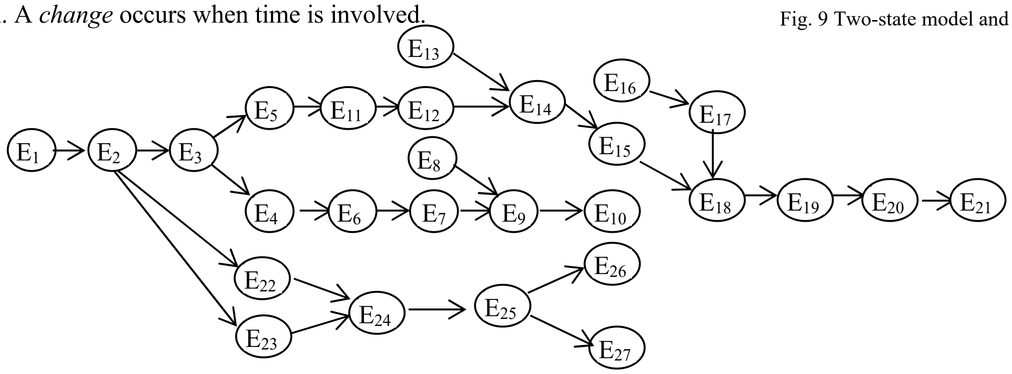


Fig. 7 The behavior model of the library system.

It is clear that the five actions—create, process, release, transfer, and receive—are not the so-called states. In Fig. 8, a thing in a stage does not change. For example, if the thing changes from a green to a blue color, this change occurs *inside* the *process* stage. When a thing moves from, say, the create stage to the release stage, the relevant change is in the context of the thing, not in the thing itself (its position, orientation, etc.). To see the involved changes in the presence of time, consider a typical two-state system such as a typical binary signal model. Let us describe it in terms of a two-level height, as shown in Fig. 9. Each stage and time form an event, as shown in Fig. 9.

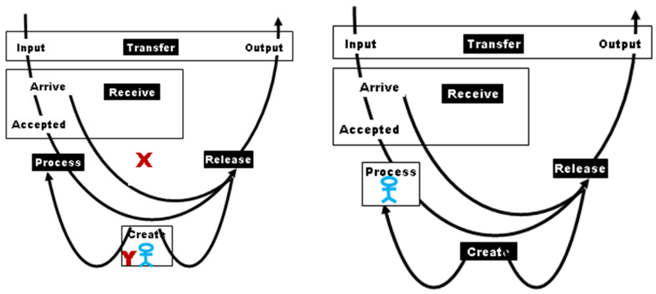


Fig. 8 Machine X and thing Y.

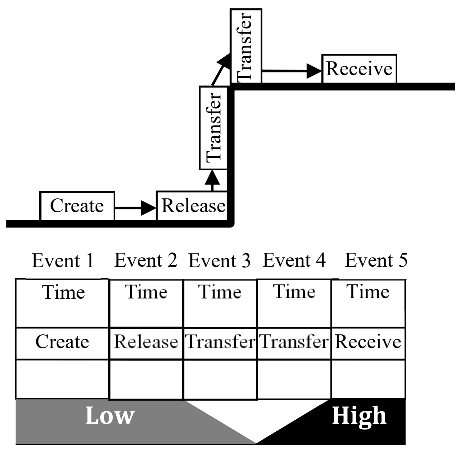


Fig. 9 Two-state model and TM states.

As can be observed, the two states of low and high are not generic events, whereas create, release, transfer, and receive are generic. The linking together of the various stages of the thing's path requires viewing a thing (e.g., signal) at different stages in time as a single thing. Fig. 9 shows what we previously called a potential action "possessed" by time durations" to generate events. The "fuzzy" events of transfer are an interesting phenomenon that needs more analysis to understand the nature of events.

5. A Glimpse on the Event Notion

The focus in the remaining part of this paper will be on exploring the notion of an event as manifested in different applications and relating it to TM events.

5.1 General

The notion of an event plays a prominent role in many fields of a study, including philosophy, linguistics, literary theory, probability theory, artificial intelligence, physics, and—of course—history [8]. Event perception, event recognition, event memory, event conceptualization, and segmentation have long been studied in several fields of psychological research [8]. In cognitive science, the perceived world is structured into objects, places, and actions that form parts of events [16], as well as numbers [17] as core knowledge domains, which form the framework of perceptual categories. Common sense typically construes events as "concrete, dated particulars, i.e., as non-repeatable entities with a specific location and duration" [8]. The structure of language attests to the primacy of the event in human cognition. Event structure (i.e., the combination of constituents encoding objects, actions, and location) is the fundamental building block for sentence meaning and grammar [7].

Whitehead [10] recognized that "the event is the ultimate unit of natural occurrence." Whitehead defined events as chunks in the life of nature that refer to the experience of activity (or passage) [13]. According to Shipley [14], "events appear to be a fundamental unit of experience, perhaps even the atoms of consciousness, and thus should be the natural unit of analysis for most psychological domains."

In common language, the term "event" encompasses wider range meanings, including things that happen on very short or very long timescales, such as interactions between subatomic particles or the orbit of Saturn around the sun [15]. The linguist's use of the notion of an event may not cohere with the vision scientists whom themselves have "changed their use and understanding of such notions over the years" [8]. Gärdenfors [16] suggested that events are an overarching category for combining different perceptual categories and gluing together objects, actions, and locations. Event structures are represented in terms of conceptual spaces—one for actions and one for results—and mappings between these spaces.

5.2 Verbs and Events

From the linguistic point of view, the TM's five generic actions imply the reduction of all verbs to five generic verbs. According to Tversky et al. [18], verbs do not describe components of *events* the way nouns alone can describe. Consider, for example, the list of verbs: *take*, *spread*, *fold*, and *put*. Without knowledge of the *objects* being acted on, we cannot know if this is about baking a cake or putting away the laundry. This implies that verbs are not parts of the world (next to objects); rather, they are components (alongside with generic actions) in determining the structure of events. TM introduces a different picture see (Figs. 10 and 11) and divides the linguistic expression into two levels. In the TM static level, objects and verbs (specified as generic actions) form a "structure" that becomes a dynamic description of events when time is added to the structure. Fig. 10 shows the TM representation of these sequence of verbs: *take*, *spread*, *fold*, and *put*. The figure indicates that machine B takes a thing from machine A, performs two types of processing (spreading and folding), and puts the thing in machine C. According to Tversky et al. [18], folding flour into a batter and folding a sheet are achieved with very different movements of the body. By contrast, in the TM, assuming that this refers to hands that are part of a human arm, machine B's "hands" perform the same movement, diagrammatically, for different objects (e.g., a sheet and clay).

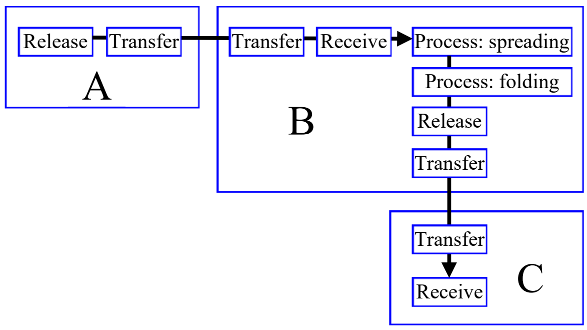


Fig. 10 TM static representation of *take*, *spread*, *fold*, and *put*.

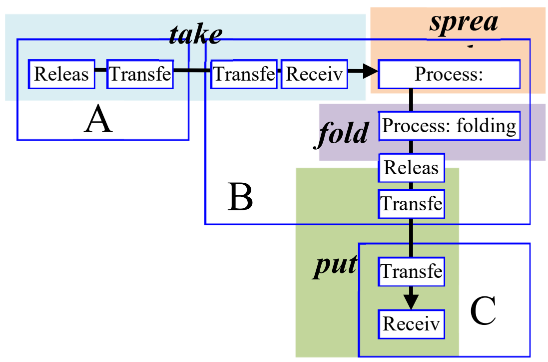


Fig. 11 TM representation of the events *take*, *spread*, *fold*, and *put*

The problem stems from verb genericity. For example, *take* is a non-generic verb because it can be expressed in the generic verbs *release* and *transfer*. It is not possible to take anything without the holder of the thing releasing and letting it go (output) and the receiver getting (input) and receiving it. *Take*, *spread*, *fold*, and *put* are not generic actions. For example, in the case of *take*, assume there are two agents (A and B in Fig. 11); hence, the first agent releases, and the second agent receives, in addition to the first agent output (transfer) and the second agent input (transfer).

We also emphasize here that, in TM, verbs (generic actions) and objects (things) form a static structure and that events are the dynamism of this structure when time is involved. A generic action (e.g., create) is a potentiality in the static structure in the same sense as a flow (an action) is represented by an arrow in the structure. The static description, as a stable all-encompassing frame of potentialities, does not specify individuals such as instances or events. In the static form (e.g., TM diagram/subdiagram), everything is there, nothing corresponds to time (past, present, or future), and nothing corresponds to, say, the principle of non-contradiction. However, what is "there" is loaded with potentiality that can be exemplified by actuality.

Additionally, these types of linguistic studies mix staticity with dynamism. Fig. 12 expresses the behavioral model that corresponds to Fig. 11. Here the verbs *take*, *spread*, *fold*, and *put* take their form as events that integrate their regions of the TM static description and time.

5.2 Events and Generic Events

According to a recent article [19], Davidson [20] showed (1967) that the same event may be compositionally described by multiple modifiers (e.g., *Jones buttered the toast* and *Jones buttered the toast slowly*). Such a type of analysis views a sentence as a whole "lump-sum" and mixes static representation with its corresponding dynamic semantics. TM representation converts the sentence into its generic actions and then identifies dynamic features in terms of events. For example, Fig. 12 shows the TM representation of *Jones buttered the toast*, and Fig. 13 shows the corresponding event representation. Fig. 14 shows the logical sequence of events.

The events in Fig. 14 reflect a high-level abstraction of elementary events that are shown in Fig. 13. Each of these elementary events takes time (change). Assuming that Jones handles the toast before handling the butter, we can develop movie shots of the events, as shown in Fig. 15: First, Jones appears (created), the toast appears, gapped by Jones (transfer, transfer, and receive), the Butter appears (created), gapped by Jones (transfer, transfer, and receive), and the toast is buttered. A few of these scenes are illustrated in Fig. 16.

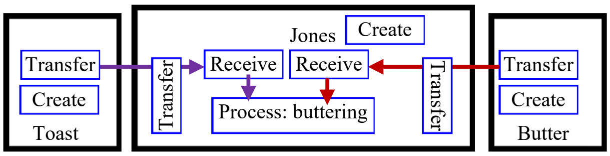


Fig. 12 The TM static representation of *Jones buttered the toast*.

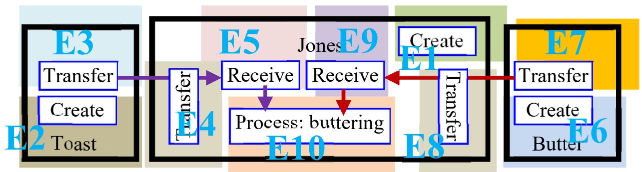


Fig. 13 The TM elementary events of *Jones buttered the toast*.

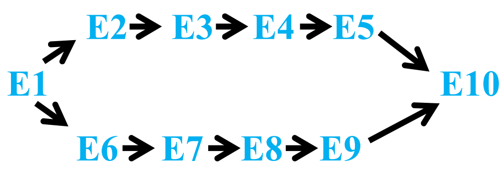


Fig. 14 The logical sequence of events of *Jones buttered the toast*.

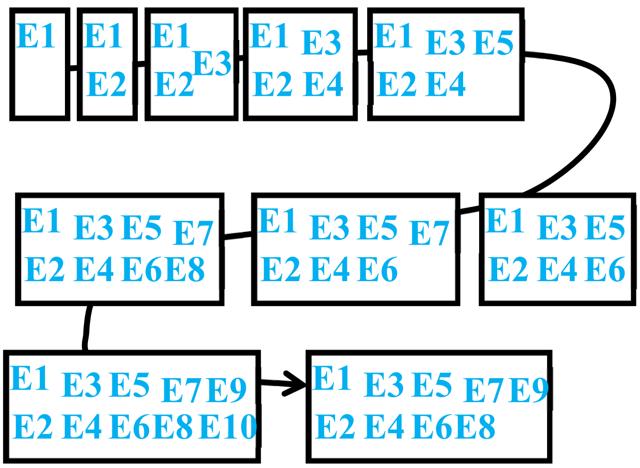


Fig. 15 Changes in the scene of *Jones buttered the toast*.

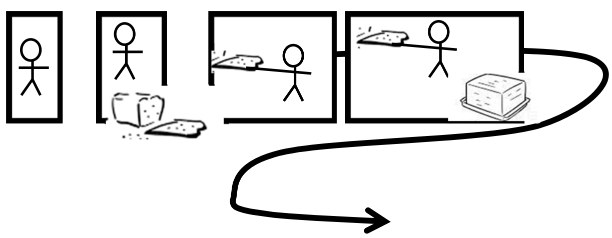


Fig. 16 A few of the changes in the scene of *Jones buttered the*

Events overlap, creating events at varying levels of granularity (may be called compound events [14]; e.g., a metal sphere is simultaneously rotating and getting warm), and then its rotation and its getting warm appear to be simultaneous distinct (generic) events within the same thimac (see discussion in Casati et al. [8]).

This phenomenon of hierarchical and overlapping events can be viewed as a mechanism of Gestalt grouping: The ongoing stream of activity is parsed into meaningful wholes [15]. All events, generic or at a higher level, are made of the same stuff: the five generic actions and time. High-level events form a coarse description, whereas generic events are the finest level of event segmentation. The events may also have other subthimacs of associated properties (e.g., intensity).

6 Events and Movement

A TM event is *a static thimac with a time breath (time subthimac)* that infuses dynamism in the thimac. It arises from how the TM static region is infected with time. Such a view is not far from the linguistics definition and structure of an event that consists of three parts [21]: a predicate (e.g., TM subdiagram); an interval of time on which the predicate occurs (TM time subthimac); and a situation under which the predicate occurs (TM). However, the TM event cannot be described only as a unit of experience (apprehending being – [10]); rather, it is made up of multilevel units of dynamic phenomenon based on, at the lower level, the five actions as units that are grasped by our experience. Dynamism is a regulating mechanism of the static form that aligns it with reality through such machinery as igniting and chronologizing actions, logicalizing, and executing/controlling processes. An event's characteristic is its singularity (in terms of time slot), but we say that an event is repeated, referring to its repeatability over the same *region*.

6.1 Event Movement

A TM event is intrinsically tied to the duration of time. An event may refer to a series of subevents. Dretske [22] observed that an event can move. However, it may be said that an event has moved in the sense that its TM regions have changed. Taking an example from Dretske [22], Fig. 17 shows a picnic that has moved from the building to a garden: The guests of the picnic event moved from the building to the garden. Fig. 18 shows an intermediate event (fuzzy event) where some of the guests are in the building and some are in the garden.

The classical definition is that movement (or motion) is simply a *change* in time. The physical movement is executed through infinite, continuous steps or a very large number of small chemical movements (e.g., ions moving through a membrane). It is a change in a spatial position. In this context, we have to distinguish between change in a thing (TM create and TM process) and change in thimac (TM release, TH transfer, and TM receive); for example, a thing (e.g., a chameleon) may change its color while it is in the same thimac.

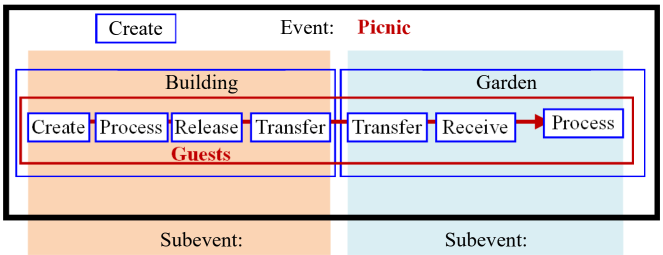


Fig. 17 An event starts in a building and moves to the garden.

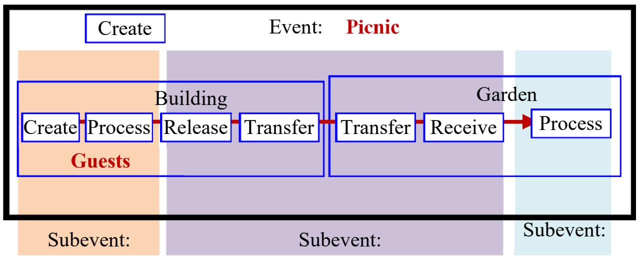


Fig. 18 An intermediate event of a picnic that starts in a building and moves to the garden.

The nature of movements and changes in this context is worth additional study.

6.2 Time and Movement

Time and movement (motion) are connected to each other. Commonly, the passage of time is not, as noted, relative to the change in position. Consider Fig. 19 (left). In a single photograph, we cannot be certain whether the dancer is moving or standing still. Observing her at different points in time, we can decide that the dancer has not changed its posture in the left picture of Fig. 19.

In TM terminology, there is no change in the region of the event (i.e., endurance through time). According to such a view, the enduring posture is a historic route of static thimac between, say, 11:45-12:00, with successive time thimacs.

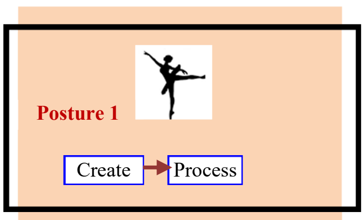


Fig. 19 The TM model of changing the posture.

Such a picture is similar to Whiteheads's process ontology where objects are stable patterns of sequential actual occasions. According to Shipley [14], continued existence of an object is an event because it requires a reference to time. An apple falling is an event, and an apple existing in time is an event (see what happens to it after a long time of existence).

In the picture in Fig. 20 (right), there is a change (legs, hands, and head movements) to reach this second posture; hence, now, three events are illustrated in Fig. 21. The transition (the dotted V in Fig. 20) is fuzzy in the sense that it is an unstable condition, which is more than potentiality but less than actuality (event). The three events in Fig. 21 (right) have existential order (left [before] posture, [between] fuzziness, and right [after] posture).

7. Conclusion

In this paper, we further explore the TM model that concentrates on the notion of an event as a dynamic phenomenon stemming from the five generic actions in TMs. We started by giving a sample application of a TM in software engineering in the form of a conceptual model of a library services system. From this general applicability of the TM model, we inquired deeper into the connection between the notions of potential action in the static description and the dynamism generated by events. It is interesting that the TM model can be used in expressing a typical business process, such as a library services system, and that similarly can be utilized to model the movement of a dancer at the static and dynamic levels. This implies the viability of the model as a general conceptual modeling tool.

References

[1] Robinson, S.: *Conceptual Modelling for Simulation Part I: Definition and Requirements*. Journal of the Operational Research Society 59(3), 278–290 (2008) doi:10.1057/palgrave.jors.2602368

[2] Shannon, R.E.: *Systems Simulation: The Art and Science*. Prentice-Hall, Englewood Cliffs, NJ (1975)

[3] Karagöz, N., Demirörs, O.: *Conceptual Modeling Notations and Techniques*. In: Robinson, S., Brooks, R., Kotiadis, K., van der Zee, D.-J. (eds.) Conceptual Modeling for Discrete-Event Simulation 2011. CRC Press, (January 2011) doi:10.1201/9781439810385-c7.

[4] Breiner, S., Padi, S., Subrahmanian, E., Sriram, R.D.: National Institute of Standards and Technology Interagency or Internal Report 8358, Rep. 8358 (May 2021) doi.org/10.6028/NIST.IR.8358

[5] Zueva, E.A., Zueva, V.V.: *On the Issue of Category and Categorization.* Innovative Processes in Research and Educational Activity, 69–71 (2014). https://www.researchgate.net/profile/Slav-Petkov/publication/308222902_SBORNIK_2014/links/57decbb908ae72d72eac12a4/SBORNIK-2014.pdf#page=69

[6] Rettler, B., Bailey, A.M.: *Object*. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy 2017 (Winter 2017 Edition) https://plato.stanford.edu/archives/win2017/entries/object/

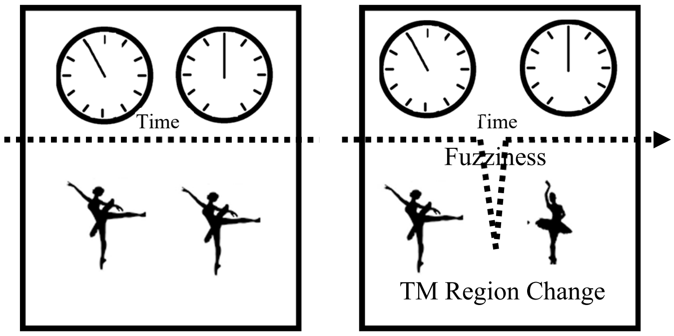


Fig. 20 Changing the posture including the intermediate event.

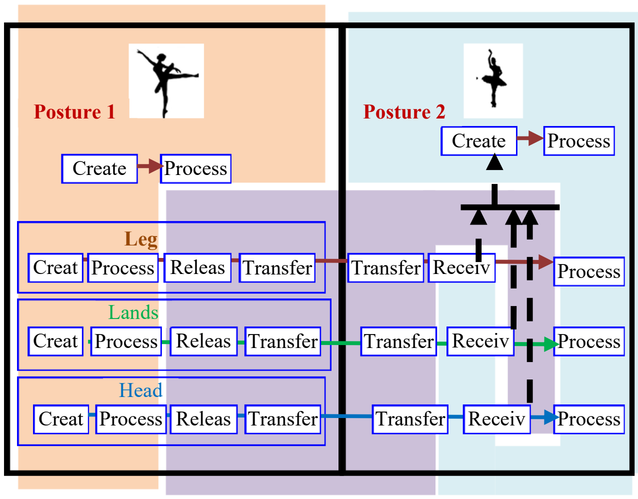


Fig. 21 The TM model of changing the posture including the intermediate event.

[7] Sinha, C., Gärdenfors, P.: *Time, Space, and Events in Language and Cognition: A Comparative View*. Science 13(26), 72–81 (2014) doi:10.1111/nyas.12491

[8] Casati, R., Varzi, A.: *Event Concepts*. In: Shipley, F., Zacks, J. (eds.) Understanding Events: From Perception to Action 2007. Oxford Scholarship Online, Oxford

[9] (2007) doi:10.1093/acprof:oso/9780195188370.003.0002

[10] Al-Fedaghi, S.: *UML Modeling to TM Modeling and Back*. International Journal of Computer Science and Network Security 21(1), 84–96 (2021) doi:10.22937/IJCSNS.2021.21.1.13

[11] Whitehead, A.N.: *Science and the Modern World*. Free Press, New York (1925/1997)

[12] Yu, L.: *Event Philosophy: Ontology, Relation and Process*. Advances in Literary Study 10, 120–127 (2022) https://doi.org/10.4236/als.2022.101009

[13] Rosado da Cruz, A.M.: *A Pattern Language for Use Case Modeling*. Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development, 408–414 (2014)

[14] Hertz, T., Mancilla Garcia, M.: *The Event: A Process Ontological Concept to Understand Emergent Phenomena*. Philosophy Kitchen Journal 11 (2019) doi:10.13135/2385-1945/4008

[15] Shipley, T.F.: *An Invitation to an Event*. In: Shipley, T.F., Zacks, J.M. (eds.) Understanding Events: From Perception to Action, 2008, pp. 3–30. Oxford University Press, Oxford (2008) https://doi.org/10.1093/acprof:oso/9780195188370.003.0001

[16] Zacks, J.M.: *Event Perception*. Scholarpedia 3(10), (2008).

[17] Gärdenfors, P.: *Primary Cognitive Categories Are Determined by Their Invariances*. Frontiers in Psychology

[18] (2020). https://doi.org/10.3389/fpsyg.2020.584017

[19] Carey, S.: *The Origin of Concepts.* Oxford University Press, Oxford (2009)

[20] Tversky, B., Zacks, J.M., Hard, B.M.: *The Structure of Experience*. In: Shipley, T.F., Zacks, J.M. (eds.) Understanding Events: From Perception to Action, pp. 3–30. Oxford University Press, Oxford https://doi.org/10.1093/acprof:oso/9780195188370.003.0001

[21] Baratella, R., Guizzardi, G., Guarino, N.: *Events, Their Names, and Their Synchronic Structure, Preprint in ResearchGate: Applied Ontology*. IOS Press (2022)

[22] Davidson, D.: *The Logical Form of Action Sentences.* In: Rescher, N. (ed.) The Logic of Decision and Action 1967, pp. 81–95. University of Pittsburgh Press, Pittsburg, PA (1967)

[23] Zhong, Z.: *Event Ontology Reasoning Based on Event Class Influence Factors*. International Journal of Machine Learning and Cybernetics 3, 133–139 (2012) doi:10.1007/s13042-011-0046-8

[24] Dretske, F.: *Can Events Move?* Mind 76, 479–492 (1967)