International Journal of Database Theory and Application
Vol.8, No.5 (2015), pp.1-14
http://dx.doi.org/10.14257/ijdta.2015.8.5.01

A Temporal Formal Languages Model Based on Categorical Methods

Decheng Miao, Jianqing Xi and Jindian Su
Institute of Information Science & Engineering, Shaoguan University, Shaoguan, China
Institute of Computer Science & Engineering, South China University of Technology, Guangzhou, China
tony10860@126.com

Abstract

Temporal data model ensures the correctness, validity and compability of data in temporal database, and temporal formal languages model describes the semantic specification of temporal database, both two are core and basis of temporal database system design and development. The paper has exploited the quo of temporal data models and temporal formal languages models, presented FTDM (Formal Temporal Data Model) and L(FTDM) over FTDM by denotation semantics methods. Referring to thinking of software reuse, this paper proposed the definition of languages reuse. Based on the previous works the authors further made temporal formal languages models family {Li} with its particular categorical properties analyzed in this paper by categorical methods, also demonstrated rigorously inherent relationships between temporal formal languages models in {Li}. Their works provided a category theoretic method with universality and flexible expansion for temporal formal languages models, meantime, proposed solid theoretical foundations for design and development of temporal database system.

Keywords: *formal languages; data model; temporal database; category theory; denotation semantics*

1. Introduction

Temporal data model is the mainline of trends in temporal database technology, and it is the core and basis of temporal database system development. ER model is an important kind of data model, whose graphical specification which is natural and easy to understand makes it become a prevalent concept modeling tool of database system. With the increasing of a large number of database application demands related to time, temporal extension on ER model to accurately capture information on time change has become one of research hot spots [1]. At present, the temporal database technology is still in the stage of research and development. Although academia and industry have proposed dozens of temporal data models, the current temporal data models are immature enough to form complete international standards [2]. Most models just focus on the research into temporal attribute of database at data processing level. Some problems such as presentation of temporal logic information, knowledge inference and integrity constraints are not solved well, therefore, the ability to process temporal information is weak, temporal data calculation is not complete, temporal data dependence is lack of support by standard temporal normalization theory, and existing integrity constraints are difficult to normalize temporal data models effectively.

ISSN: 2005-4270 IJDTA

Traditional formal languages models based on Turing machine and Von Neumann architecture can deal with deterministic problems of temporal database effectively, but don't make substantive progress on uncertainty algorithms. Many relevant problems are attributed to NP - complete problems, so only choose the non-optimal solutions by reducing standards of solving problem, adopt some optimization algorithms whose complexities are not always polynomial time, or some optimization algorithms only satisfy part application examples.

Category theory, widely applied to theoretical computer science including formalization theory of software system, formal semantics and programming methods, provides a kind of general theory tools, thinking methods and research techniques for mastering relationships between computer architecture, software specifications and programming. Category theory also furnishes a unified mathematical framework for describing the diversified links to structured systems effectively, researches the universality and similarity of objects, attaches importance to relationships between objects without limiting to internal structure of objects, which is suitable for modeling at higher abstract level. Researching on formal languages modeling theory by categorical methods and universal algebraic tools is a hot topic at present [3-6].

The remainder of this paper is organized as follows. Section 2 reviews some related works about the current research achievements of temporal data models and temporal formal languages models. Some basic concepts about time models are introduced firstly in section 3, then presents FTDM (Formal Temporal Data Model) at higher abstract level. Section 4 further makes a temporal formal languages model L(FTDM) over FTDM by denotation semantics. Section 5 proposes the definition of language reuse by thinking of software reuse, and puts forward temporal formal languages models family {Li} with its particular categorical properties analyzed in this paper by categorical methods, also demonstrates rigorously inherent relationships between temporal formal languages models in {Li}. Finally concludes this paper and proposes some future works.

This paper is a revised and expanded version of the conference paper in [7]. Whereas the conference paper covers only making the formal temporal data model FTDM, this paper also presenting a temporal formal languages model L(FTDM) by denotation semantic methods. Accordingly, the material in section IV is entirely new.

2. Related Works

Some representative temporal data models are concluded in Figure 1. It points out that most data structures are relations, from which extending traditional relational data model, and relational database is as a special case of temporal database. Three kinds of time types are supported by temporal database, namely, VT (Valid Time), TT (Transaction Time) and UDT (User Defined Time) in [1], VT and TT are two kinds of orthogonal dimensions, and temporal data model support one of dimensions at least. All temporal data models support UDT, TempEER and TimeER in [8], BCDM in [2] in Figure 1 support TT. There are three kinds of temporal data structures in VT, *i.e.*, instant, interval and period in [8]. The both two express a period of time, but the former eliminates start and end points, only BCDM supports all data types of VT in Figure 1. Two time granularities are defined in temporal database, that is, single-granularity and multi-granularity in [1], five temporal data models support multi-granularity in Figure 1, and others delay the choice of time granularities in logical design phase of temporal database considering different time granularities for concrete applications. Temporal data manipulation extends traditional relation manipulation, adds some manipulations such as temporal select, temporal project, temporal join et al. Most models in Figure 1 provide VT operation including relation expression, temporal expression and SQL expression, RAKE in [8] only supports ordinary SQL operation, but MOTAR in [8] and TERC+ in [8] extend object-oriented function of temporal data model. All data manipulations in Figure 1 are functions, some of which can be nested, at the same time, TempEER, BCDM and TimeER provide VT and TT operations. Upward compatibility [8] to old versions provides a smooth transition to new version for enhancing function, seven temporal data models in Figure 1 has such upward compatibility.

At present most query languages of temporal data models extend the temporal data query function of SQL and Quel, but the temporal query function is limited, with low query efficiency. Query language of BCDM includes TempSQL, Tquel, TSQL2, *etc*. Four temporal data models in Figure 1 do not support temporal constraints, TERM in [8] and TERC+ have the best standardization even though the both support temporal constraints, but they cannot express inherent dependency constraints between UDT changing attributes flexibly.

With the promoting of complexity and application requirements of temporal database system, some technique problems caused by non-functional factors such as reliability, security and uncertainty are increasingly prominent in temporal database development process. Improving the ability of existing formalism methods to solve problems, researching and developing some new methods and technologies with universality, flexibility and efficiency has become new research focus on temporal formal languages models, especially in some performances such as highly abstract, flexible express, unified description and so on, there exists less special research institutions and significant achievements. Zsyntax modeling for biochemical system was a formal languages model [9], and its biochemical processes were rewritten and analyzed as a kind of logical deduction, but some problems including effective proof of Zsyntax theorems need to be further solved. $A^{xml(T)}$, a formal languages model, presented in [10] based on RBAC (Role Based Access Controls) with temporal constraints, provided access control to information stored in XML document formatted by converting logic programs from security policy database. Breaking through the confine of traditional formal languages models, in [11] presented a categorical formal languages model, analyzed the correlations between objects in different categories deeply by bi-functor, which provided a new way of thinking to deal with uncertainty problem effectively.

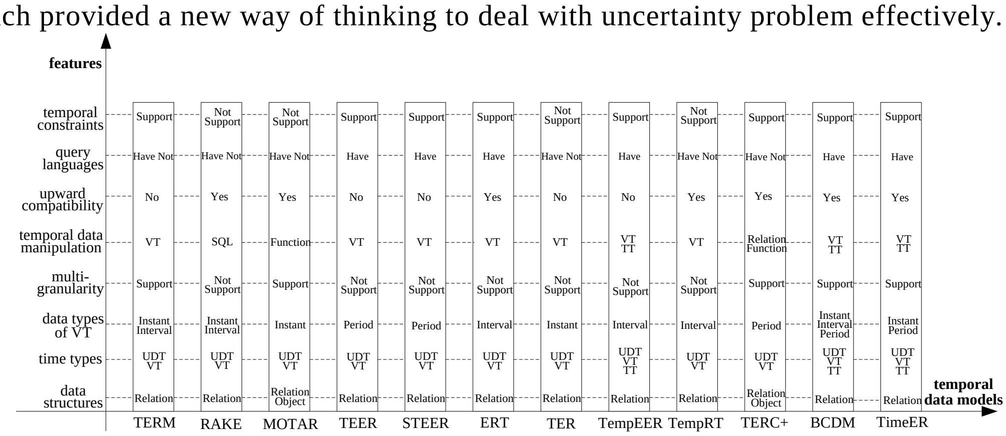


Figure 1. Some Representative Temporal Data Models

Existing temporal formal languages models effectively capture temporal characteristics of data in concept by different ways wholly, but each temporal formal languages model that has stronger orientation to its application does not possess universality, which is hard to meet all requirements of temporal characteristics for temporal database. At the same time, to break the limitation of traditional formal languages models for temporal database system applications, many scholars have done a lot of works, and bring forward many new temporal formal languages models, but there exists less temporal formal languages models with pervasive computing and flexible extensibility. This paper presents FTDM, proposes a kind of temporal formal languages model L(FTDM) on FTDM, makes temporal formal languages models family {Li} using language reuse, and analyzes inherent relationships between temporal formal languages models in {Li} deeply by categorical methods, which provides an efficient and convenient formalization theory framework for studying temporal formal languages models.

3. Formal Temporal Data Model

3.1 Time Model

Let $S=\{s_1,s_2,\cdots,s_n\}$ be finite set, $S_1\times S_2\times\cdots\times S_n=(s_1,s_2,\cdots,s_n)$ be Cartesian product, $s_i\in S_i, 1\le i\le n$, $M(S)=\{\{s_1,s_2,\cdots,s_n\}\}$ be multiple set, $S^+=<s_1,s_2,\cdots,s_n>$ is non-null finite domain over S. Write $S_1\uplus S_2$ for disjoint union, and $\perp$ for undefined element of set.

Based on discrete linear time model, the time domain in this paper which is isomorphic to finite subset of natural number set $\mathbb{N}$ is a total order finite set, whose elements are called time quantum in [2], denoted to $c_i, i\in\mathbb{N}$. Time baseline is divided into many independent time slices which have equivalent length by c_i, $g=|c_i|$ is time granularity, which is explicitly specified by the applications, reflects the minimum of temporal database time. Two time variants, namely, *now* and *UC* [8], the former depends on current time for its effective value, and the latter refers changed time of tuples in TT.

Let VTE be VT of entity in temporal database, VTA be VT of attribute in temporal database, and different VT domains of attribute be $D^g_{VTA}=\{c^a_1,c^a_2,\cdots,c^a_k\}$, all VT domains of attribute be $\mathcal{D}_{VTA}=\bigcup_g D^g_{VTA}$, different VT domains of entity be $D^g_{VTE}=\{c^e_1,c^e_2,\cdots,c^e_{now}\}$, and all VT domains of entity be $\mathcal{D}_{TT}=\bigcup_g D^g_{TT}$.

3.2 Formal Temporal Data Model

Data structures, data manipulations and integrity constraints are three basic components of data model. Data structures are the foundation of data model, and temporal data structures describe object types that reflect static characteristics of temporal database system, all of which constitute the basic compositions of temporal database. Data manipulations represent dynamic characteristics of database system, and temporal data manipulations define operation collection allowed to perform by temporal database. The exact meaning, operation symbols and operating rules of temporal data manipulations must be defined by temporal data model. Integrity constraints are semantic constraints and dependent rules between data and its relationships in model data, temporal constraints restrict states and its change in temporal database, which ensure the correctness, effectiveness, and compatibility of data in temporal database.

Definition 1. The FTDM presented in this paper is a triple, namely, $FTDM = (TDS, TDM, TSC)$, TDS is temporal data structure, TDM is temporal data manipulation, TSC is temporal constraint, which are expressed by BNF (Backus-Naur Form) separately:

$$TDS ::= SCH \mid E_D \mid R_D \mid A_D \mid T_s \mid t$$

SCH is schema of temporal database, E_D and R_D are entity type and relation type separately, A_D is attribute definition, T_s is time domain, and t is tuple.

$$TDM \supseteq \{\cup_T : E_D \times E_D \to E_D, -_T : E_D \times E_D \to E_D, \times_T : E_D \times \cdots \times E_D \to E_D, \sigma_T : E_D \to E_D,$$
$$\pi_T : E_D \to \{t[A] \mid t \in E_D\}\}$$

$t[A]$ is component of attribute A for t . Temporal union, temporal difference, temporal Cartesian product, temporal select and temporal project are five basic operations of temporal data manipulations, other temporal manipulations such as temporal intersection, temporal join and temporal division may be expressed by this five basic operations.

$$TSC ::= TSC_1; TSC_2 \mid EIC \mid WEIC \mid RIC \mid IIC \mid UIC \mid PIC \mid TC$$
$$EIC ::= SPK \mid TIK \mid TK$$
$$UIC ::= IS\_A \mid HAS\_PART\_OF \mid GC$$
$$GC ::= GP \mid GT$$
$$PIC ::= \textit{Snapshot Participation of } E \textit{ in } R \textit{ is } (\min, \max) \mid \textit{Valid Time Participation of } E \textit{ in } R \textit{ is } [\min, \max] \mid$$
$$\textit{Participation of } R \textit{ to } E \textit{ is } p_1, p_2$$
$$p_1 ::= \textit{disjoint} \mid \textit{overlapping}$$
$$p_2 ::= \textit{total} \mid \textit{partial}$$
$$TC ::= DEX \mid DEV$$

Temporal constraints have seven types including EIC (Entity Integrity Constraints), WEIC (Weak Entity Integrity Constraints), RIC (Referential Integrity Constraints), IIC (Inherent Integrity Constraints), UIC (User-defined Integrity Constraints), PIC (Participation Integrity Constraint) and TC (Transition Constraints). SPK (Simple Primary Key), TIK (Time Invariant Key) in [12] and TK (Temporal Key) in [13] are three types of EIC. SPK determines an entity uniquely in every snapshot of temporal database. TIK, which is a kind of SPK, does not change over time in VT of existing entity, for example, a TIK is that worker number is unique in 50 years, but the same worker number gives another worker 50 years later. TK is a kind of TIK, which is unique in VT of entity wholly. The existence of weak entity must be on the premise of VT of its owner entity, and RIC defines rules referred between foreign key and key. IIC is a particular constraint type of temporal database, and attribute VT must be subset of its entity VT, at the same time, attribute and entity TT must also be subset of the entity VT when a transaction is performed in temporal database. IS_A in [8,14], HAS_PART_OF in [8] and GC (Generation Constraints) in [15] are three types of UIC; GC restricts target entity generated from source entity, which has GP (Generation Product) and GT (Generation Transition) according to whether the source entity is retained or not. The former retains the source entity, for instance, target entity PROJECT is generated by relation MANAGEMENT and the source entity MANAGER is still retained in temporal database; the latter does not retain the source entity, for example, department re-organized is different to the original department. TC (Transition Constraints) in [16] constraints the evolution of source entity to target entity, which has DEX (Dynamic Extension) and DEV (Dynamic Evolution) according to whether the entity roles are compatible before and after evolution. The roles of DEX are compatible, for instance, role of a worker is still employee after the worker is promoted to manager. The roles of DEX are not compatible, for example, the roles are not compatible after the department manager is promoted to senior director. DEV cannot apply to parent entity and child entity, and DEX of two disjoint entities is DEV.

4. Temporal Formal Languages Model

Referred by text expressive method of TimeER in [8], this paper presents a temporal formal languages model L(FTDM) over FTDM based on formal languages theory and denotation semantics methods, namely, L(FTDM) $=(SYN, SEM, AUF, SEF)$. SYN and SEM are syntax domains and semantic domains of L(FTDM) separately, AUF is auxiliary function, whose prefix is temporal data type returned, square brackets of auxiliary function definition are optional items, and SEF is semantic function. Some meta variants symbols are introduced in this paper, E、$\textbf{\textit{Weak}}\ E$、$R$、$A$ is entity name, weak entity name, relation name and attribute name separately, $B \in 2^A$ is subset of attributes, and bold format is participle parsed by compile system.

<u>*SYN*</u>

$SCH ::= E_D; R_D$

$E_D ::= E_{D1}; E_{D2} \mid \textbf{\textit{Entity Type}}\ E\ \textbf{\textit{has}}\ A_D \mid \textbf{\textit{Entity Type}}\ E\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D \mid \textbf{\textit{Weak Entity Type}}\ E\ \textbf{\textit{has}}\ A_D \mid$
$\textbf{\textit{Weak Entity Type}}\ E\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D \mid \textbf{\textit{Entity Type}}\ E_1\ \textbf{\textit{IS\_A}}\ E_2\ \textbf{\textit{has}}\ A_D \mid \textbf{\textit{Entity Type}}\ E_1\ \textbf{\textit{IS\_A}}\ E_2\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D$

$R_D ::= R_{D1}; R_{D2} \mid \textbf{\textit{Relationship Type}}\ R\ \textbf{\textit{has}}\ A_D \mid \textbf{\textit{Relationship Type}}\ R\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D$

$A_D ::= A_{D1}; A_{D2} \mid \textbf{\textit{Attribute Type}}\ A\ \textbf{\textit{is}}\ d \mid \textbf{\textit{Attribute Type}}\ A\ \textbf{\textit{is}}\ d\ \textbf{\textit{with}}\ T_s$

$d ::= \text{int} \mid \text{boolean} \mid \text{string}$

$T_s ::= T_{s1}; T_{s2} \mid (dim, ts, g)$

$dim ::= VTE \mid VTA \mid TT$

$ts ::= instant \mid period$

$g ::= sec \mid min \mid hour \mid day \mid week \mid month \mid year$

<u>*SEM*</u>

Let $D_S \cup \{\perp\}$ be substation set in [12], $D_S^E \subseteq D_S$ be substation set of entity E, $D_{VTA} = \mathcal{D}_{VTA} \cup \{\perp\}$ be VT domain set of attributes, $D_{VTE} = \mathcal{D}_{VTE} \cup \{\perp\}$ be VT domain set of entities, $D_{TT} = \mathcal{D}_{TT} \cup \{\perp\}$ be TT domain set, $\mathcal{D}[\![d]\!]$ be domain set of basic data types, $PRED$ be predicates set, $Role : E_D$ is roles set, and D_{dim}^g is time set.

<u>*AUF*</u>

The parameters of binary auxiliary function $inSch$ are entity name and relation name or database schema, whose returning value is true if the entity or relation is in this schema, else is false.

$$\text{boolean} : inSch(E, SCH) = \text{boolean} : inSch(E, E_D) = \begin{cases} ture\ \ if\ \ \textbf{\textit{Entity Type}}\ E\ [\textbf{\textit{with}}\ T_s]\ \textbf{\textit{has}}\ A_D \in E_D \\ false\ \ otherwise \end{cases}$$

$$\text{boolean}: inSch(\textbf{\textit{Weak}}\ E, SCH) = \text{boolean}: inSch(\textbf{\textit{Weak}}\ E, E_D) = \begin{cases} ture & if\ \textbf{\textit{Weak Entity Type}}\ E \\ & [\textbf{\textit{with}}\ T_s]\textbf{\textit{has}}\ A_D \in E_D \\ false & otherwise \end{cases}$$

$$\text{boolean}: inSch(R, SCH) = \text{boolean}: inSch(R, R_D) = \begin{matrix} true & if\ \textbf{\textit{Relationship Type}}\ R\ [\textbf{\textit{with}}\ T_s]\ \textbf{\textit{has}}\ A_D \in R_D \\ false & otherwise \end{matrix}$$

The parameter of unary auxiliary function $attOf$ is declaration of entity type or attribute definition, whose returning value is attribute name of the entity.

$$A_D : attOf(\textbf{\textit{Entity Type}}\ E\ [\textbf{\textit{with}}\ T_s]\ \textbf{\textit{has}}\ A_D) = A_D : attOf(A_D)$$

$$A_D : attOf(A_{D1}; A_{D2}) = A_D : attOf(A_{D1}) \cup A_D : attOf(A_{D2})$$

$$A_D : attOf(\textbf{\textit{Attribute Type}}\ A\ \textbf{\textit{is d}}\ [\textbf{\textit{with}}\ T_s]) = \{A\}$$

The parameter of unary auxiliary function $entPar$ is entity type name or weak entity type name or relation name, whose returning values are names of all entities, weak entities participated in this relation. The symbol of $E_i \in R$ represents that entity E_i participates in relation R, the result of single entity or weak entity in this relation is multi-set.

$$E_D : entPar(R) = \begin{matrix} \{E_1, E_2, \cdots, E_n\} & if\ E_i \in R, 1 \le i \le n \\ \bot & otherwise \end{matrix}$$

$$E_D : entPar(R_1; R_2) = E_D : entPar(R_1) \cup E_D : entPar(R_2)$$

$$E_D : entPar(E) = \{\{E\}\}$$

$$E_D : entPar(\textbf{\textit{Weak}}\ E) = \{\{E\}\}$$

The parameters of binary auxiliary function $belTo$ are weak entity name and relation name, whose returning value is entity name affiliated by this weak entity.

$$E_D : belTo(\textbf{\textit{Weak}}\ E, R) = \begin{matrix} \{entPar(R) - \textbf{\textit{Weak}}\ E\} & if\ \textbf{\textit{Weak}}\ E \in R \\ \varnothing & otherwise \end{matrix}$$

The parameter of unary auxiliary function $attList$ is name of entity or weak entity, whose returning value are all attribute names. Meantime, the returning values are attributes names of this sub-entity and its parent if the parameter is sub-entity; the returning values are only all attributes names of this weak entity if the parameter is weak entity.

$$A_D : attList(E) = \begin{cases} attOf(\textbf{\textit{Entity Type}}\ E \cdots) & if\ \textbf{\textit{Entity Type}}\ E \cdots \in E_D \\ attOf(\textbf{\textit{Entity Type}}\ E\ \textbf{\textit{IS\_A}}\ E' \cdots) \cup A_D : attList(E') & \\ \quad if\ \textbf{\textit{Entity Type}}\ E \cdots \in E_D & \\ \bot\ otherwise & \end{cases}$$

$$A_D : attList(\textbf{\textit{Weak}}\ E) = attOf(\textbf{\textit{Weak Entity Type}}\ E \cdots)$$

The parameter of unary auxiliary function $entNum$ is relation, whose returning value is count of entities participated in this relation. R is natural join between all entities participated, which has n attributes, $R.A$ is attributes set of R, $\pi_{T\,A_i}(R)$ is temporal project for attribute A_i of R.

$$\text{int}: entNum(R) = \begin{cases} 0 & if\ R.A = \varnothing \\ cnt(R -_T E) & if\ \cup_{1 \le i \le n} \pi_{T\,A_i}(R) \ne E \\ cnt(R -_T E) + 1 & if\ \cup_{1 \le i \le n} \pi_{T\,A_i}(R) = E \end{cases}$$

<u>*SEF*</u>

For specified type of entity and relation or attribute definition, semantic function $\mathcal{T}$ returns time domain supported by special temporal type, whose function declarations and semantic equations are followed:

$$\mathcal{T}: T_s \rightarrow D_{VTE} \cup D_{VTA} \cup D_{TT}$$
$$\mathcal{T}[\![\textbf{\textit{with}}\ T_{s1}; T_{s2}]\!] = \mathcal{T}[\![T_{s1}]\!] \times \mathcal{T}[\![T_{s2}]\!] \tag{1}$$
$$\mathcal{T}[\![(dim, instant, g)]\!] = D_{dim}^{g} \tag{2}$$
$$\mathcal{T}[\![(dim, period, g)]\!] = 2^{D_{dim}^{g}} \tag{3}$$

The parameter of semantic function $\mathcal{A}$ is attribute definition, whose returning value is special temporal type; its function declaration and semantic equation are followed:

$$\mathcal{A}: A_D \times d \times T_s \rightarrow \mathcal{D}[\![d]\!] \cup (\mathcal{T}[\![T_s]\!] \rightarrow \mathcal{D}[\![d]\!])$$
$$\mathcal{A}[\![A_{D1}; A_{D2}]\!] = \mathcal{A}[\![A_{D1}]\!] \times \mathcal{A}[\![A_{D2}]\!] \tag{4}$$
$$\mathcal{A}[\![\textbf{\textit{Attribute Type}}\ A\ \textbf{\textit{is}}\ d]\!] = \mathcal{D}[\![d]\!](d) \tag{5}$$
$$\mathcal{A}[\![\textbf{\textit{Attribute Type}}\ A\ \textbf{\textit{is}}\ d\ \textbf{\textit{with}}\ T_s]\!] = \mathcal{T}[\![T_s]\!] \rightarrow \mathcal{D}[\![d]\!](d) \tag{6}$$

Semantic function $\mathcal{I}$ determines substation set of entity type, and temporal database automatically generates unique substation attribute in [12] of this entity in whole system to identify entity changed over time; its function declaration and semantic equation are followed:

$$\mathcal{I}: E \rightarrow D_S^E$$
$$\mathcal{I}[\![E]\!] = \begin{cases} D_S^E & if\ E \in E_D \\ \perp & otherwise \end{cases} \tag{7}$$

Semantic function $\mathcal{E}$ identifies entity instance by the definition of entity type, returns tuples set of this entity stored in temporal database. dom is domain function, a is substation attribute in entity semantic equation, s is substation operation in [12] in weak entity or entity inherited semantic equation, sE_i is unique substation attribute of entity E_i in temporal database, $E_2.T_s$ is temporal characters of parent entity E_2 in entity inherited semantic equation. The function declaration and semantic equation of $\mathcal{E}$ are followed:

$$\mathcal{E}: E_D \times T_s \times A_D \rightarrow D_S^E \times \mathcal{A}[\![A_D]\!] \cup (D_S^E \times \mathcal{T}[\![T_s]\!]) \times \mathcal{A}[\![A_D]\!]$$
$$\mathcal{E}[\![E_{D1}; E_{D2}]\!] = \mathcal{E}[\![E_{D1}]\!] \uplus \mathcal{E}[\![E_{D2}]\!] \tag{8}$$
$$\mathcal{E}[\![\textbf{\textit{Entity Type}}\ E\ \textbf{\textit{has}}\ A_D]\!] = \{t \mid t \in S^+ \wedge dom(t) = \{a, attOf(A_D)\} \wedge t[a] \in \mathcal{I}[\![E]\!] \wedge_{A_i \in attOf(A_D)} t[A_i] \in \mathcal{A}[\![A_D]\!] \wedge (\forall t_i, t_j, i \neq j \Rightarrow t_i[a] \neq t_j[a])\} \tag{9}$$
$$\mathcal{E}[\![\textbf{\textit{Entity Type}}\ E\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D]\!] = \{t \mid t \in S^+ \wedge dom(t) = \{a, attOf(A_D)\} \wedge t[a] \in (\mathcal{T}[\![T_s]\!] \rightarrow \mathcal{I}[\![E]\!]) \wedge_{A_i \in attOf(A_D)} t[A_i] \in \mathcal{A}[\![A_D]\!] \wedge \forall c^e \in \mathcal{D}_{VTE}\ \forall c^t \in \mathcal{D}_{TT} (\forall t_i, t_j, i \neq j \Rightarrow t_i[a] \neq t_j[a])\} \tag{10}$$
$$\mathcal{E}[\![\textbf{\textit{Weak Entity Type}}\ E\ \textbf{\textit{has}}\ A_D]\!] = \{t \mid t \in S^+ \wedge dom(t) = \{\cup_{E_i \in belTo(E,R)} sE_i, attOf(A_D)\} \wedge_{E_i \in belTo(E,R)} t[sE_i] \in \mathcal{I}[\![E_i]\!] \wedge_{A_i \in attOf(A_D)} t[A_i] \in \mathcal{A}[\![A_D]\!]\} \tag{11}$$
$$\mathcal{E}[\![\textbf{\textit{Weak Entity Type}}\ E\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D]\!] = \{t \mid t \in S^+ \wedge dom(t) = \{\cup_{E_i \in belTo(E,R)} sE_i, attOf(A_D)\} \wedge_{E_i \in belTo(E,R)} t[sE_i] \in \mathcal{T}[\![T_s]\!] \rightarrow \mathcal{I}[\![E_i]\!] \wedge_{A_i \in attOf(A_D)} t[A_i] \in \mathcal{A}[\![A_D]\!]\} \tag{12}$$
$$\mathcal{E}[\![\textbf{\textit{Entity Type}}\ E_1\ \textbf{\textit{IS\_A}}\ E_2\ \textbf{\textit{has}}\ A_D]\!] = \{t \mid t \in S^+ \wedge dom(t) = \{sE_2, attList(E_2), attOf(A_D)\} \wedge t[sE_2] \in \mathcal{T}[\![E_2.T_s]\!] \rightarrow D_S^{E_2} \wedge_{A_i \in attList(E_2)} t[A_i] \in \mathcal{A}[\![A_D]\!] \wedge_{A_i \in attOf(A_D)} t[A_i] \in \mathcal{A}[\![A_D]\!]\} \tag{13}$$

$$\mathcal{E}[\![\textbf{\textit{Entity Type}}\ E_1\ \textbf{\textit{IS\_A}}\ E_2\ \textbf{\textit{with}}\ T_s\ \textbf{\textit{has}}\ A_D]\!] = \{t \mid t \in S^+ \wedge dom(t) = \{sE_2, attList(E_2), attOf(A_D)\} \wedge t[sE_2] \in \mathcal{T}[\![E_2.T_s]\!] \times \mathcal{T}[\![T_s]\!] \rightarrow D_S^{E_2} \wedge_{A_i \in attList(E_2)} t[A_i] \in \mathcal{A}[\![A_D]\!] \wedge_{A_i \in attOf(A_D)} t[A_i] \in \mathcal{A}[\![A_D]\!]\} \quad (14)$$

5. Temporal Formal Language Models Family

5.1. Languages Reuse

The value of software reuse has been cognized by people in modern software engineering, so establishing reusable database of software development organization for engineering practice has become the consensus in the academia and industry, and the SEI (Software Engineering Institute) already has set up and used reusable software components as one of aims to improve CMM (Capability Maturity Model) for software development organizations. By the idea of software reuse and semantics conversion methods [17], and referring the views of inheritance, expansion and shielding in [18], the concept of languages reuse is put forward in this paper.

Definition 2. If formal languages model $L_P =< SYN_P, SEM_P, AUF_P, SEF_P >$ is known, four basic elements of a new formal languages model L_R , namely, $SYN_R, SEM_R, AUF_R, SEF_R$, are expressed by L_P adopted some transforming rules, then $L_R =< SYN_R, SEM_R, AUF_R, SEF_R >$ is made. L_R is called reusable formal languages model of L_P, and L_P is called basic formal languages model of L_R.

Definition 3. For two formal languages models L_R and L_P , $L_R =< SYN_R, SEM_R, AUF_R, SEF_R >$, $L_P =< SYN_P, SEM_P, AUF_P, SEF_P >$,

i) L_R is called simple reuse of L_P if $SYN_R = SYN_P, SEM_R = SEM_P, AUF_R = AUF_P, SEF_R = SEF_P$;

ii) L_R is called expansive reuse of L_P if $SYN_R = SYN_P \bigcup_T SYN, SEM_R = SEM_P \bigcup_T SEM, AUF_R = AUF_P \bigcup_T AUF, SEF_R = SEF_P \bigcup_T SEF$, SYN, SEM, AUF, SEF is non-null, and $SYN \not\subset SYN_P, SEM \not\subset SEM_P, AUF \not\subset AUF_P, SEF \not\subset SEF_P$;

iii) L_R is called selective reuse of L_P if $SYN_R = SYN_P -_T SYN, SEM_R = SEM_P -_T SEM, AUF_R = AUF_P -_T AUF, SEF_R = SEF_P -_T SEF$, SYN, SEM, AUF, SEF are the same as ii).

5.2. Temporal Formal Languages Models Family

Using temporal formal languages model to describe software specifications of temporal database system, based on L(FTDM), some temporal formal languages models with different levels can be made by applying definition 3, namely, simple reuse, expansive reuse and selective reuse, to L(FTDM) finite times, which consist of a temporal formal languages models family $\{L_i\}$. Isomorphic relation between temporal formal languages models implicated by simple reuse provides more selectivity and larger flexibility to solve specific problems for temporal database system developers. Expansive reuse makes evolution process of version sequences of temporal formal languages models family $\{L_i\}$ keeping efficiency and scalability. Selective reuse is a pruning and backtracking of versions evolution process, deletes some elements of basic formal languages model L_P and adds new components by expansive reuse to generate reuse formal language model L_R . Every temporal formal languages model in $\{L_i\}$ is the closures of L(FTDM), and L_i describes abstract expression degree of software specifications of temporal database system at different levels. Temporal formal languages model lies in higher level, whose abstract expression becomes clearer and its formal description becomes stronger, so as to make temporal database system developers to code and test easily.

Nowadays researching temporal formal languages model by categorical method has been a hot topic, and category theory has applied widely in theoretical computer science including formal semantics, programming methods and software specifications. Comparing with other methods of temporal formal languages model, categorical method presents a universal theory tools, thinking methods and research means for understanding computer architectures, software specifications and relations between programming.

Theorem 1. Let all temporal formal languages models in temporal formal languages models family $\{L_i\}$ be objects, and morphisms between temporal formal languages models in $\{L_i\}$ be morphisms, then the system including these objects and morphisms above constructs a category $\mathcal{C}$.

Proof. Write $Obj\,\mathcal{C}$ for objects set of $\mathcal{C}$ whose elements are all temporal formal languages models in $\{L_i\}$, $Mor\,\mathcal{C}$ is morphisms set of $\mathcal{C}$ whose elements are all morphisms between temporal formal languages models in $\{L_i\}$, $dom: Mor\,\mathcal{C} \to Obj\,\mathcal{C}$ is domain function, $cod: Mor\,\mathcal{C} \to Obj\,\mathcal{C}$ is codomain function, $\circ: Mor\,\mathcal{C} \times Mor\,\mathcal{C} \to Mor\,\mathcal{C}$ is composite relationship between functions. The following is the process proving the system $\mathcal{C} = (Obj\,\mathcal{C}, Mor\,\mathcal{C}, dom, cod, \circ)$ is a category.

i) Satisfy the matching conditions.

There exists three arbitrary formal languages models in $\mathcal{C}$, $L_i, L_j, L_k \in Obj\,\mathcal{C}$, two morphisms between them, namely, $f = L_i \to L_j \in Mor\,\mathcal{C}$, $g = L_j \to L_k \in Mor\,\mathcal{C}$, then $g \circ f = L_i \to L_k \in Mor\,\mathcal{C}$ is the composite of f and g , so these formal languages models and morphisms satisfy the matching conditions, namely, $dom(g \circ f) = L_i = dom(f), cod(g \circ f) = L_k = dom(g)$.

ii) Satisfy the associative laws.

Let $h = L_k \to L_m \in Mor\,\mathcal{C}$, the definitions of f, g is same as i). Let $h \circ g = L_j \to L_m \in Mor\,\mathcal{C}$, $h \circ (g \circ f) = L_k \to L_m \circ L_i \to L_k = L_i \to L_m$, and $(h \circ g) \circ f = L_j \to L_m \circ L_i \to L_j = L_i \to L_m$, so the associative laws defined by category theory are satisfied, namely, $h \circ (g \circ f) = (h \circ g) \circ f$.

iii) Satisfy the condition of identify existed.

For every object $L_i \in Obj\,\mathcal{C}$, there exists an identify $1_{Li} \in Mor\,\mathcal{C}$ uniquely, satisfies $dom(1_{Li}) = cod(1_{Li}) = L_i$, and for every morphism $f \in Mor\,\mathcal{C}$, $f \circ 1_{Li} = f$ if $dom(f) = L_i$; $1_{Li} \circ f = f$ if $cod(f) = L_i$.

By i), ii), iii) above and definition of category in [19], the system $\mathcal{C} = (Obj\,\mathcal{C}, Mor\,\mathcal{C}, dom, cod, \circ)$ is a category, namely, let all temporal formal languages models in temporal formal languages models family $\{L_i\}$ be objects, and morphisms between temporal formal languages models in $\{L_i\}$ be morphisms, then the system including these objects and morphisms above constructs a category $\mathcal{C}$.

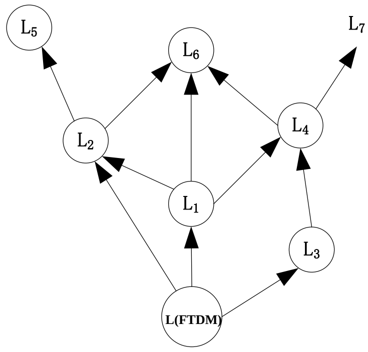


Figure 2. Temporal Formal Languages Models Family

In category $\mathcal{C}$, all objects, namely, all temporal formal languages models in temporal formal languages models family $\{L_i\}$, and all morphisms between objects constitute a poset according to reuse relations of formal languages models. Elements which have no minimum upper bound in the poset are called the maximal elements, and set of maximal elements consists of all terminal objects in $\mathcal{C}$. Every maximal element is also temporal formal languages model whose abstract ability is the highest in its application fields, which is suitable for software developers of temporal database system to code and test. From the point of views of category theory, graphs constituted by some objects and morphisms in Figure 2 are commutative diagrams in [19] such as from L_1, L_4, L_6 to L_1, L_2, L_6 , from $L(FTDM), L_3, L_4$ to $L(FTDM), L_1, L_4$, from $L(FTDM), L_2$ to $L(FTDM), L_1, L_2$ and so on. The relationship of commutative diagrams of some temporal formal languages models in $\{L_i\}$ are showing that there exists a variety of reusable ways of temporal formal languages models, for example, software engineers can directly present temporal formal languages model L_2 from L(FTDM) by application of simple reuse, selective reuse and expansive reuse finite times, meantime, also can firstly propose temporal formal languages model L_1 from L(FTDM), then make temporal formal languages model L_2 from L_1 . This flexibility reflected by the relationship of commutative diagrams provides great convenience for temporal database systems developers to choose proper temporal formal languages models, whose abstract expression ability is suitable to solve particular problems of specific developing phrase of temporal database according to developers' preferences and system requirements.

5.3. Categorical Method of Complex Temporal Formal Languages Model

For designing of large-scale and complicated formal languages model of temporal database system, this paper takes category theoretical functor as a basic tool to study it. For two different categories $\mathcal{C}_1$ and $\mathcal{C}_2$, the following definitions are presented to research objects in $\mathcal{C}_1$ and $\mathcal{C}_2$, namely, relationships between temporal formal languages models.

Definition 4. For two functors, $F, G : \mathcal{C}_1 \to \mathcal{C}_2$, natural transformation $\alpha : F \to G$ is sets followed: $\alpha = \{\alpha(M_1) : F(M_1) \to G(M_1) \in Mor\ \mathcal{C}_2 \mid M_1 \in Obj\ \mathcal{C}_1\}$, and for every morphism $f : M_1 \to M_2 \in Mor\ \mathcal{C}_1$, elements in sets α are satisfying $G(f) \circ \alpha(M_1) = \alpha(M_2) \circ F(f)$.

It is shown that category takes functors as objects from definition 4, whose morphisms are natural transformations. Natural transformation $\alpha : Obj\ \mathcal{C}_1 \to Mor\ \mathcal{C}_2$ analyzes the relationship between two different objects in $\mathcal{C}_2$ that are mapped by two different functors from the same object in $\mathcal{C}_1$. If there exists natural transformation $\beta : G \to F$ satisfying $\beta \circ \alpha = 1_F$, $\alpha \circ \beta = 1_G$, then α, β is natural isomorphism. The property of category isomorphism requires that categories have exactly the same structure, but the practical application of temporal database system is difficult to strictly meet the categorical isomorphism condition, so it may determine relatively weak property of category equivalence using natural isomorphism. Two categories $\mathcal{C}_1$ and $\mathcal{C}_2$ are equivalent if there exists two opposite functors F, G, $F : \mathcal{C}_1 \to \mathcal{C}_2$, satisfy that $G \circ F = 1_{\mathcal{C}_1}$ and $F \circ G = 1_{\mathcal{C}_2}$. Functor preserves the isomorphism of objects, and equivalence of categories constitutes equivalence relation between categories. Different temporal database systems have different categorical models, analyzing whether categories is equivalent or not by composite theorems of natural transformation in [19] to determine to utilize the same temporal formal languages model during the course of formal languages modeling process of temporal database system, which simplifies temporal database system development greatly.

6. Conclusions

This paper presents a convenient and efficient formal temporal data model FTDM at higher abstract level, puts forward the concept of language reuse, on the basis of the L(FTDM), a temporal formal languages model over FTDM, makes temporal formal languages models family $\{L_i\}$ by application of simple reuse, expansive reuse and selective reuse finite times, which constitutes temporal formal languages models facing different applications fields and development stages of temporal database systems, also analyzes inherent relationships of temporal formal languages models in $\{L_i\}$ by method of category theory.

Comparing with some temporal formal languages models in other literatures, the advantages of L(FTDM) in this paper are as follows. Firstly, choosing syntax domains and semantic domains of temporal database system application fields appropriately, temporal database system developers can easily make L(FTDM) to design complex temporal formal languages models family $\{L_i\}$ layered. Secondly, more practical significance implicated by diversity of temporal formal languages models designing not only provides a variety of choices for temporal database system developers, but it is also convenient to develop temporal database systems with clear structures, distinct layers, accurate and reliable, easy to maintain by the flexible selecting of $\{L_i\}$. Thirdly, organizing L(FTDM) properly and ensuring reliability and independence of L(FTDM), temporal database system developers can propose convenient and efficient temporal formal languages models family $\{L_i\}$ as the foundation to design temporal database system. Meanwhile, the convergence of $\mathfrak{M}$ in [20] for $\{L_i\}$ ensures to get all true propositions of fields knowledge of temporal database system eventually, and then to formally verify the domain system effectively; the interchangeability of Th in [20] for $\{L_i\}$ ensures that updating each finite software version can keep the validity of all theory closures and scalability of business logics of temporal database systems; the minimality of $\mathfrak{A}$ in [20] for $\{L_i\}$ can guarantee each version of temporal database system is tiny formalization theory, and all axioms in axiom system used to verify effectively the correctness of temporal database system are independent of each other. Finally, this paper deeply analyzes the relationships between temporal formal languages models in $\{L_i\}$ using the method of category theory. During the course of designing process of temporal formal languages model, this paper emphatically studies some methods and technologies including software reuse, semantic transformation, model abstraction and category theory.

Comparing with traditional methods of formal languages, Fibrations theory, a branch of category theory, provides a uniform axiomatic method whose particular priors reflect in syntax construction, behavior description and semantics computation of formal languages. Some problems such as minimality and completeness of FTDM, semantics analysis of seven kinds of temporal integrity constraints in L(FTDM), reliability of semantic transformation between the temporal formal languages models and its coordination are future research works by categorical methods such as Fibrations theory.

Acknowledgment

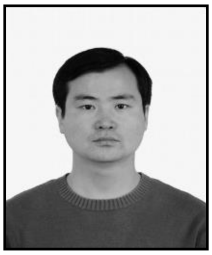
This study is supported by Natural Science Foundation of China under Grant(No. 61103038), Guangdong Province Natural Science Foundation, China(Grant No. S2013010015944), Guangdong Province Core Technology Research Project of Strategic Emerging Industries under Grant (No. 2011A010801008, 2012A010701011, 2012A010701003), Scientific and Technological Plan Research Foundation of Shaoguan Municipal Science and Technology Bureau, Guangdong, China (Grant No. 2013CX/K61).

References

[1] Z. X. Hao, Editors, J. Y. Geng and F. Tang, “Temporal database design theory”, Science Press, Beijing**(2009)**, pp. 5-38.
[2] Y. Tang, X. P. Ye and N. Tang, “Temporal information processing technology and its applications”, Science Press, Beijing **(2010)**, pp. 16-85.
[3] J. Diarmuid Pigott, J. Valerie Hobbs, “Complex knowledge modeling with functional entity relationship diagrams”. VINE. 2,41**(2011)** , pp. 192-211.
[4] I. David Spivak, E. Robert Kent, “Ologs: a categorical framework for knowledge representation”. Public Library of Science ONE. 1,7**(2012)** , pp. 1-22.
[5] I. David Spivak, T. Giesa, E. Wood, “Category theoretic analysis of hierarchical protein materials and social networks”. Public Library of Science ONE. 9,6**(2011)** , pp. 1-15.
[6] M. Johnson, R. Rosebrugh, R. J. WOOD, “Lenses, fibrations and universal translations”. Mathematical Structures in Computer Science. 1,22**(2012)** , pp. 25–42.
[7] D. C. Miao, J. Q. Xi and J. D. Su, “A category theoretic methods of temporal data model”. 4th International Conference on Emerging Intelligent Data and Web Technologies,**(2013)** Sept. 9-11;Xi’an,China, pp. 109-114.
[8] H. Gregersen, C. S. Jensen, “Temporal entity-relationship models—a survey”. IEEE Transactions on Knowledge and Data Engineering, 3,11**(1999)** , pp. 464-497.
[9] D. Boniolog, M. D. Agostino, P. P. Di Fiore, “Zsyntax: a formal language for molecular biology with projected applications in text mining and biological prediction”. PLoS ONE, 3,3**(2010)** , pp. 1-12.
[10] P.Sean, Z.Yan, “A formal language for specifying complex XML authorisations with temporal constraints”. Lecture Notes in Computer Science. 6151**(2011)** , pp. 443-457.
[11] D. C. Miao, “Study on category computation model of formal language”. Journal of Shaoguan University. 12,32**(2011)**, pp. 8-11.
[12] C. Combi, S. Degani, C. S. Jensen, “Capturing temporal constraints in temporal ER models”. Lecture Notes in Computer Science. 5231**(2008)** , pp. 397-411.
[13] P. McBrien, “Temporal constraints in non-temporal data modeling languages”. Lecture Notes in Computer Science. 5231**(2008)** , pp. 412-425.
[14] Q. Hoang, T. V. Nguyen, “Extraction of TimeER model from a relational database”. Lecture Notes in Computer Science. 6591**(2011)** , pp. 57-66.
[15] D. C. Miao, J. Q. Xi, “Formal languages models for temporal data”. Computer Science. 4,39**(2012)** , pp. 172-176.
[16] A. Artale, E. Franconi, “Foundations of temporal conceptual data models”. Lecture Notes in Computer Science. 5600**(2009)** , pp. 10-35.

[17] M. Broy, M. Wirsing, P. Pepper, "On the Algebraic Definition of Programming Language". Journal of the ACM Transactions on Programming Languages and Systems. 1, 9**(1987)**, pp. 54-99.
[18] N. X. Zhang, H. J. Zheng, "Abstraction of programming languages and a model of language family". Journal of Beijing University: Natural Science Edition. 5,33**(1997)** , pp. 650-657.
[19] M. Barr, C. Wells, Editor, C. A. R. Hoare, "Category theory for computing science", Prentice-Hall, New York**(1990)** , pp. 11-270.
[20] W. Li, Editor, P. Lin and Y. Z. Zhao, "The basic principle and calculation of mathematical logic", Science Press, Beijing **(2009)** , pp. 171-181.

Authors



Decheng Miao, received his PhD degree in Institute of Computer Science & Engineering from South China University of Technology, in 2012. He is currently an associate professor in Institute of Information Science & Engineering, Shaoguan University, China. His research interests include formal languages theory, categorical theory methods.

Jianqing Xi, received his PhD degree in Department of Computer Science & Technology from National University of Defense Technology, in 1992. He is currently a professor Institute of Computer Science & Engineering from South China University of Technology, China. His research interests include software theory, database and network computation.

Jindian Su, received his PhD degree in Institute of Computer Science & Engineering from South China University of Technology, in 2007. He is currently an associate professor Institute of Computer Science & Engineering from South China University of Technology, China. His research interests include formal methods, coalgebraic and bialgebraic.