Formalising Temporal Attributes in Temporal conceptual data models

E. A. Nasubo Ongoma
Department of Computer Science, University of KwaZulu-Natal and UKZN/CSIR - Meraka
Durban, South Africa
212562258@stu.ukzn.ac.za

1. INTRODUCTION

Conceptual data models represent some aspect of the real world known as mini world. This is the first step before creating a database and it defines concepts, their properties and relations between them. Traditional databases are static, they show only one view of the data at a time (valid time), without showing the history of the database. However, the world changes with time thus the need for temporal databases, to keep track of the changes within an organisation.

Temporal research seeks to answer the question, how does the world change, and how do we capture this change? In two ways, either lifecycle, which is the succession of membership (status) and transition, the change of object properties over time with the rules that govern the change. Temporal databases maintain history, manage and keep track of changes so as to plan for the future.

The first step towards having temporal databases is to have a temporal conceptual data model that details succession of membership and change of object properties. This is done by extending traditional conceptual data models (e.g ER, UML and ORM) with temporal constructs. These contracts may vary and depending on how many constructs they are, they may make the diagram hard to interpret and would require the modellers to understand them [7]. Although there are a lot of temporal models, we do not have a compete model with reasoning capabilities that can show inconsistencies on the diagram,as a result, save time.

Representing temporal data in conceptual data models and ontologies is required by various application domains to record attributes that change over time. Consider these application areas:

- Administration: Companies require the history of employee to determine future rewards, for example, employee cannot get a bonus if they have been suspended.
- Medical Information Systems need the patient's history to determine how they can treat patients, e.g. for example a HIV positive patient evolves to an AIDS patient after the attribute value of CD4 count drops below 180.
- Security systems need to monitor and authorise users as they log into systems e.g. time bound passwords
- Financial institutions need to record the history for future use, e.g. before giving a credit facility. Temporal attributes are used to determine the amount a client can receive and the time in which it can be repaid.

All these situations require the use of temporal attributes, but to represent this effectively, we require a language that is expressive enough to capture the operational semantics of time-varying information. This research looks at logic-based representation of temporal data at the conceptual level which provides a link between temporal conceptual data models and a temporal description logic. Description logics provide formal semantics that characterise temporal conceptual modelling constructs by specifying the structure of conceptual data models which in turn permits reasoning on implicit knowledge [5]. Previous research focussed on representing and reasoning on temporal classes and relationships, but had scant support for representing and reasoning over temporal attributes, if at all. Artale et al [4] formalised the temporal ER model ER_{VT} for classes and relations, but only formalised timestamping on temporal attributes. Little formalisation of temporal attributes prevents the full utilisation of temporal conceptual data models and tracking the interaction of temporal attributes with temporal classes. If temporal attributes are not properly represented in our conceptual models, one will face inconsistent databases with respect to the constraints they ought to hold.

We extended the very expressive temporal description logic language $\mathcal{DLR}_{\mathcal{US}}$ [1, 2, 4, 8] with a precise syntax and semantics for temporal attributes [11]. This extension allowed us to formalise temporal attributes by defining their lifecycle and their transitions. The results from this research are two fold, the first result introduces the notion of status attributes [9] as done for status classes [4] and status relations, [8], which captures the evolution of a temporal attribute as it moves along a temporal object. The second result is attribute transition, which concerns the change of attribute properties over time as the attribute migrates from one temporal class to the next. We gave the formalisation of status and transition for temporal attributes as well as their logical implications which permit the correct behaviour in subsumption and transition in temporal classes.

SAICSIT Masters and Doctoral Symposium '14, Sept. 29, Centurion, South Africa

With the formalisation of attributes, we now have a temporal ER model that is fully formalised and we can check for satisfiability (inconsistencies) in the diagram and can get some implications that are hidden. $\mathcal{DLR}_{\mathcal{US}}$ is very expressive and allows full temporalization of attributes but at the same time it is undecidable. With these results, we are a step closer towards having a fully formalised temporal model.

2. TEMPORALISING ATTRIBUTES

The Description Logic $\mathcal{DLR}_{\mathcal{US}}$ [1] is an expressive fragment of FOL that combines the propositional temporal logic with *Since* and *Until* operators with the (non-temporal) description logic $\mathcal{DLR}$ [5] so that relationships and classes can be temporalised. $\mathcal{DLR}_{\mathcal{US}}$ is extended to include a precise syntax and semantics for attributes, mainly because we are dealing with conceptual data models that require attributes. A temporary attribute was defined earlier in [4], as a binary relation for each attribute, $A \in \mathcal{A}$, for $\langle A, C\rangle \in \mathcal{T}$, with its $\mathcal{DLR}_{\mathcal{US}}$ axiom as $\mathtt{C} \sqsubseteq \neg\exists[\mathtt{From}](\Box^*\mathtt{A})$. This entailed that temporal constructors can be used in front of attributes, which, however, was not included in the $\mathcal{DLR}_{\mathcal{US}}$ syntax and semantics. This formalisation introduces the use of temporal constructors explicitly for attributes. We have classes C (starting from atomic ones CN), n-ary relations R (DL roles, with $n \geq 2$, RN), binary attributes A between a class and a datatype, DL role components (U, of which F denotes a role component in an attribute, $F \subseteq U$, and $\{\mathtt{From}, \mathtt{To}\} \subseteq F$). The selection expression $U_i/n : C$ denotes an n-ary relation whose i-th argument ($i \leq n$) is of type C, and $F : C$ denotes is the role component $\{\mathtt{From}\}$ in the attribute.

$\mathcal{U}$ntil and $\mathcal{S}$ince together with $\bot$ and $\top$ suffice to define the temporal operators: $\Diamond^+$ (some time in the future) as $\Diamond^+ C \equiv \top\, \mathcal{U}\, C$, $\oplus$ (at the next moment) as $\oplus C \equiv \bot\, \mathcal{U}\, C$, and likewise for their past counterparts; $\Box^+$ (always in the future) and $\Box^-$ (always in the past) are the duals of $\Diamond^+$ and $\Diamond^-$.

$$
\begin{array}{ll}
C \to & \top \mid \bot \mid CN \mid \neg C \mid C_1 \sqcap C_2 \mid \exists^{\lessgtr k}[U_j]R \mid \exists[\mathbf{F}]\mathbf{A} \mid \\
& \Diamond^+ C \mid \Diamond^- C \mid \Box^+ C \mid \Box^- C \mid \oplus C \mid \ominus C \mid C_1\, \mathcal{U}\, C_2 \mid C_1\, \mathcal{S}\, C_2 \\
R \to & \top_n \mid RN \mid \neg R \mid R_1 \sqcap R_2 \mid U_i/n : C \mid \\
& \Diamond^+ R \mid \Diamond^- R \mid \Box^+ R \mid \Box^- R \mid \oplus R \mid \ominus R \mid R_1\, \mathcal{U}\, R_2 \mid R_1\, \mathcal{S}\, R_2 \\
\mathbf{A} \to & \top_{\mathbf{A}} \mid \mathbf{AN} \mid \neg\mathbf{A} \mid \mathbf{F} : \mathbf{C} \mid \\
& \Diamond^+\mathbf{A} \mid \Diamond^-\mathbf{A} \mid \Box^+\mathbf{A} \mid \Box^-\mathbf{A} \mid \oplus\mathbf{A} \mid \ominus\mathbf{A} \mid \mathbf{A_1}\, \mathcal{U}\, \mathbf{A_2} \mid \mathbf{A_1}\, \mathcal{S}\, \mathbf{A_2}
\end{array}
$$

Figure 1: Syntax the extended $\mathcal{DLR}_{\mathcal{US}}$, modified to include attributes (in bold face).

We will use the following equivalent abbreviations: $C_1 \sqcup C_2 \equiv \neg(\neg C_1 \sqcap \neg C_2)$; $C_1 \to C_2 \equiv \neg C_1 \sqcup C_2$; $\exists[U]R \equiv \exists_{\geq 1}[U]R$; $\forall[U]R \equiv \neg\exists[U]\neg R$; $R_1 \sqcup R_2 \equiv (\neg R_1 \sqcap \neg R_2)$ Furthermore, the operators $\Diamond^*$(at some moment) and its dual $\Box^*$(at all moments) can be defined for both classes and relations as $\Diamond^* C \equiv C \sqcup \Diamond^+ C \sqcup \Diamond^- C$ and $\Box^* C \equiv C \sqcap \Box^+ C \sqcap \Box^- C$ respectively. The syntax of the extended $\mathcal{DLR}_{\mathcal{US}}$ are included in Fig. 1 and its semantics can be found in [11].

2.1 Timestamping

To effectively manage and present temporalised attributes, it is important to state a time-scale and give the constraints to manage the evolution of data. Timestamping [4] ensures that data can be easily distinguishable if it is time changing or not. Below, we show the difference between a temporal and a non temporal attribute.

Snapshot attribute

$o \in C^{\mathcal{I}(t)} \wedge \langle o, d\rangle \in A_i^{\mathcal{I}(t)} \to \forall t' \in \mathcal{T}.\, \langle o, d\rangle \in A_i^{\mathcal{I}(t')}$

$\mathrm{C} \sqsubseteq \neg\exists\,[\mathtt{From}] : (\mathtt{A} \sqcap \Diamond^*\neg\mathtt{A})$

Temporal attribute

$o \in C^{\mathcal{I}(t)} \wedge \langle o, d\rangle \in A_i^{\mathcal{I}(t)} \to \exists t' \neq t.\, \langle o, d\rangle \notin A_i^{\mathcal{I}(t')}$

$\mathrm{C} \sqsubseteq \neg\exists\,[\mathtt{From}] : (\Box^*\mathtt{A})$

2.2 Evolution Constraints

These are rules that govern transitions in temporal conceptual data models, they put restrictions on the movement of data. These rules control the mechanism that rules dynamic aspects, permissible transitions from one state to the next [4], as well as the lifespan. To represent temporal attributes to the level of detail required and to have them interact with temporalised classes, we extended the notion of status classes and status relations to *status attributes* to specify the operational semantics of temporal attributes.

2.2.1 Status attributes

Status classes were introduced in [4] and status relations in [2], but fell short of status attributes to constrain the permissible states of affairs. Status attributes [9] also can have four different statuses: they either exist and are scheduled, active, or suspended, or they are disabled. Model theoretic semantics as well as the $\mathcal{DLR}_{\mathcal{US}}$ axiom, subsumption hierarchy and disjointness for temporal attributes in ER_{VT} model are given in [9] along with logical implications which help in deriving new constraints.

2.2.2 Attribute Migration

Attribute migration occurs when an attribute in the same object object migrates to another attribute, for example the attributes *has_degree* and *has_postgrad*, is a transition from one attribute to another. Attribute migration is more complex than object migration because it is bidirectional, i.e., it can cause migration of objects, triggering reclassification of objects as well as participate in an object migration. For example, a bank account (class) being frozen, due to expiry of a work permit.

Representation of attribute hierarchies is uncommon, mainly because it is not properly defined and formalised, but is relevant for modelling temporal data. New results on attribute hierarchies with subsumption in [9] allow us to capture the interaction between the permissible statuses of classes and status attributes in temporal transitions. A proper representation of transition in the conceptual model will enable a modeller to know how to design a temporal database.

3. DISCUSSION

Our results formalised temporal attributes for temporal conceptual data models using the extended description logic language $\mathcal{DLR}_{\mathcal{US}}$. This formalisation involved temporisation of attributes for status attributes and their transition of constraints. This work was the remaining gap towards obtaining a logic-based fully temporised modelling language that is essential for designing and maintaining temporal databases and knowledge bases. $\mathcal{DLR}_{\mathcal{US}}$ allows the temporal attributes to integrate well with the temporal entities and relations already defined with $\mathcal{DLR}_{\mathcal{US}}$ axioms over other logics. The major benefit, however, will be reaped for the modelling of temporal information, as $\mathcal{DLR}_{\mathcal{US}}$ provides a complete logic-based reconstruction of the temporally extended ER_{VT} conceptual data modelling language.

These results are twofold, formalising temporal attributes either thorough lifecycle or transition. We extended the notion of status, for attributes to map its lifecycle by giving logic based semantics to rule the permissible evolution. This allowed us to look at effect of subsumption of temporal classes (ISA) on temporal attributes. Subsumption adds temporal constraints that further enrich the data and yield more logical implications.

3.1 What solution offers

With the new insights presented in this paper, one can successfully build a *complete* temporal ER model that would be translated from the temporally extended EER to $\mathcal{DLR}_{\mathcal{US}}$ and therewith enable the option to check the consistency of a conceptual schema, hence, improve or guarantee its quality. More generally, disjointness and equivalence classes are easily overlooked when done manually, which can be checked upon for atemporal models already (e.g., [6]), but the results presented here could also validate any issues with temporal constraints.

The results presented here is the final theoretical step toward creation of a tool that would check automatically the consistency of temporal data models and thus would be able to spot such issues.

3.2 Challenges

$\mathcal{DLR}_{\mathcal{US}}$ does not have a value comparison operator that can be used to compare values to permit a migration. $\mathcal{DLR}_{\mathcal{US}}$ is undecidable, however, with the known consequences for a potential automated reasoner for it. While time-consuming computation is an acceptable trade-off for a modeller focussing on expressiveness, a slightly less expressive temporal language and better performance with the reasoner may be preferred by others. At the other end of this spectrum are results obtained with TDL-Lite and temporal OBDA [3]. It would be useful to conduct a detailed investigation as to what would be the best trade-off between a subset of temporal constructs that is most desired from a viewpoint of conceptual data modelling and the complexity 'costs' and what can be implemented in temporal OBDA. Once a temporal conceptual data modelling or ontology development tool is available, one can obtain quantitative results as to how often a construct is actually used, which further can inform the notion of 'preferred constructs' and the further development of decidable temporal logics.

4. CONCLUSION

We formalised temporal attributes using the description logic language $\mathcal{DLR}_{\mathcal{US}}$ on the temporal ER model ER_{VT}. These results are a step towards having a temporal ER model that can check its consistency.

Although we have chosen here a notation in EER, the results are just as well transferrable to UML Class Diagrams, thanks to having a textual syntax so that each element can be mapped onto a graphical element of UML, and, likewise together with some transformation rules regarding attributes and value types, it is extensible also to ORM2. In regards to DL-Knowledge bases, we need a complete translation from EER to a description logic language, to check quality properties of conceptual schema. Without the proper full temporisation of attributes, we cannot have a sound algorithm developed.

5. REFERENCES

[1] A. Artale, E. Franconi, F. Wolter, and M. Zakharyaschev. A temporal description logic for reasoning about conceptual schemas and queries. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the 8th Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*, pages 98–110. Springer Verlag, 2002.

[2] A. Artale, N. Guarino, and C. M. Keet. Formalising temporal constraints on part-whole relations. In G. Brewka and J. Lang, editors, *11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 673–683. AAAI Press, 2008. Sydney, Australia, September 16-19, 2008.

[3] A. Artale, R. Kontchakov, F. Wolter, and M. Zakharyaschev. Temporal description logic for ontology-based data access. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 711–717. AAAI Press, 2013.

[4] A. Artale, C. Parent, and S. Spaccapietra. Evolving objects in temporal information systems. *Ann. Math. Artif. Intell.*, 50(1-2):5–38, 2007.

[5] D. Calvanese and G. De Giacomo. *The DL Handbook: Theory, Implementation and Applications*, chapter Expressive description logics, pages 178–218. Cambridge University Press, 2003.

[6] P. R. Fillottrani, E. Franconi, and S. Tessaris. The icom 3.0 intelligent conceptual modelling tool and methodology. *Semantic Web*, 3(3):293–306, 2012.

[7] H. Gregersen and C. S. Jensen. Temporal entity-relationship models - a survey. *IEEE Trans. Knowl. Data Eng.*, 11(3):464–497, 1999.

[8] C. M. Keet and A. Artale. A basic characterization of relation migration. In *OTM Workshops*, volume 5231 of *LNCS*, pages 484–493. Springer, 2010.

[9] C. M. Keet and E. A. N. Ongoma. Temporal attributes: their status and subsumption. (in preparation), 2014.

[10] R. Lora, A. Sabaini, C. Combi, and U. Moretti. Designing the reconciled schema for a pharmacovigilance data warehouse through a temporally-enhanced ER model. In C. C. Yang, H. Chen, H. D. Wactlar, C. Combi, and X. Tang, editors, *Proceedings of the 2012 international workshop on Smart health and wellbeing (SHB'12)*, pages 17–24. ACM, 2012.

[11] E. A. N. Ongoma, C. M. Keet, and T. Meyer. Transition constraints for temporal attributes. In *Proceedings of the 27th International Workshop on Description Logics (DL'14)*, CEUR-WS, 2014. 17-20 July 2014, Vienna, Austria.