A Brief Introduction to Temporal Modelling with TREND

1. Introduction

Modelling of temporal constraints for information systems has received attention since the mid-1990s (e.g., [1,2,3]), as it adds expressiveness to ensure data integrity. For instance, to ensure each Graduate must have been a Student at that university before (an *evolving object*), that a couple registered as divorcing in a census database must have been marrying before (an *evolving relation*), or that workers may not always have an Office (*temporal attribute*). The key things that can happen over time can be depicted graphically along a timeline as shown in Fig.1. The horizontal lines and dots depict an object's life—for a duration or an instant—as an instance of an entity type, a tuple as a member of a relationship, or an attribute. The 'snapshot' is also called 'rigid' and means that that object is always an instance of the entity type for the entirety of its lifetime, like an apple is always an apple, and you will always be a human for your entire life. For temporal entities, they are an instance of something for some time only, as indicated with the 'temporal options' (also called 'antirigid'). For instance, you as instance of Human are an instance of Employee only for some time of your life, and the attribute of a country's COVID-19 Lockdown level is temporary as well (it didn't exist 2 years ago and one may hope we won't have it anymore in 2 years' time). Also, things can happen to the instances over time, i.e., a transition may take place in the future, or have taken place in the past: some object k as an instance of type A may also in the future become (or in the past have been) an instance of type B for some time. Such temporal notions are typically considered from a reference point, which, for conceptual modelling is normally the 'now' that also has a past and a future.

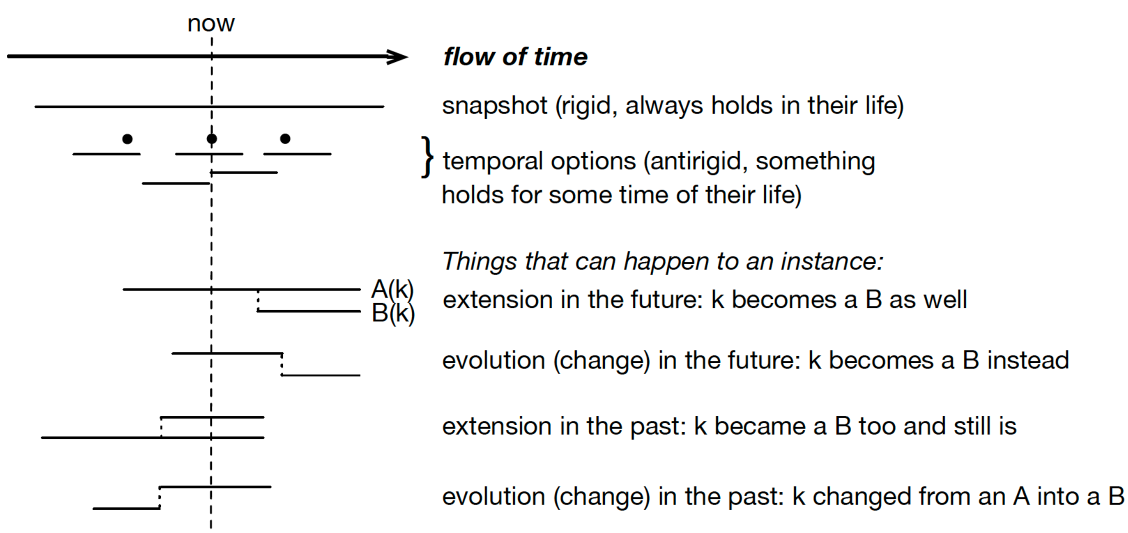


Figure 1. Graphical depiction of some key temporal aspects of an entity's life (first as A, then somehow B).

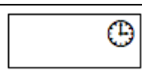
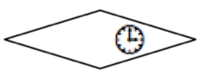
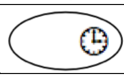
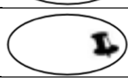
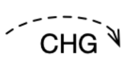
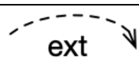
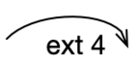
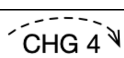
Also in Big Data and the Internet of Things, data is temporal, and thus requiring a temporal data model. For this assignment we will use an extended version of ER_{VT} [4] called TREND (**T**emporal information **R**epresentation in **En**tity-Relationship **D**iagrams) [5,6].

2. The TREND conceptual data modelling language

TREND focusses on temporalising entity types, relationships and attributes, and specifying temporal transition constraints. The core transition constraints are *dynamic extension (EXT)* and *dynamic evolution* that *changes (CHG)* things. In an extension, the entity is *also* an instance of the other entity whereas with the change, the entity *ceases* to be an instance of the source entity type. An example of extension is when an Employee becomes a Manager, but they still remain an Employee. An example of change is when a SeniorLecturer is promoted to the rank of AssociateProfessor, since then they are no longer a SeniorLecturer anymore. This temporal information cannot be represented in 'plain' atemporal EER that you have learned in class, but we need new elements to indicate that additional information. Those adornments have to be relatively easy for the analyst to use when modelling and easy to understand for the domain expert. After multiple iterations of design and evaluation, the following additions have been made, which are summarised in Table 1 below.

A clock icon indicates a temporal element, and arrows labelled with CHG and EXT represent temporal transitions in the *future*. Labels are chg and ext when the transition constraints refers to the *past*, e.g. a Graduate must have been a Student before. To remember the past/future difference, one might think of it as 'the past has already gone down, lowercase' and 'the future is a step up, uppercase'. Both future and past transitions are needed when these transition constraints differ: e.g., Student becoming a Graduate in the future is optional, but Graduate having been a Student in the past is mandatory (required/compulsory). A dashed arrow shaft denotes an optional transition and a solid shaft denotes a mandatory transition. Any transition constraint can include a time quantity to indicate how much time must elapse before the transition can occur. An attribute with a drawing pin means that the attribute is *frozen*, i.e., is not allowed to change anymore.

Table 1. Selection of the notation of the TREND diagram language.

Icon	Name	Description
	Temporal entity type	Entities of this type are ***not always*** entities of this type
	Temporal relationship	Relations of this relationship are ***not always*** relations of this type
	Temporal attribute	The entity does ***not always*** have a value for this attribute
	Frozen attribute	Once set, retains (drawing pin) its value in perpetuity (***forever***)
	Mandatory dynamic extension in the future	***Must*** (solid shaft) in the future (uppercase) ***extend to*** (EXT) also be an instance of the target type
	Optional dynamic evolution in the future	***May*** (dashed shaft) in future (uppercase) ***change*** (CHG) instead into the target type
	Optional dynamic extension in the past	***May*** in the past (lowercase) ***have extended*** (ext) from also being a member of the source type
	Mandatory quantitative dynamic extension in the past	***Must*** in the past (lowercase) ***have extended*** (ext) from also being of the source type ***4*** time units earlier
	Optional quantitative dynamic evolution in future	***May*** in the future ***change*** (CHG) instead to the target type after ***4*** time units

3. Examples of TREND models

An example TREND diagram is shown in Figure 2. Office is a temporal attribute, as, over their period of employment, employees may have their own office at some times and not have their own office at other times. The mandatory transition ext indicates that a manager must have been working for the company as a regular employee before being promoted to manager, and thus that that transition from employee to manager happened in the past. Not all employees will be promoted to manager, hence, the optional EXT from employee to manager. Likewise, the transition from work to manage is optional.

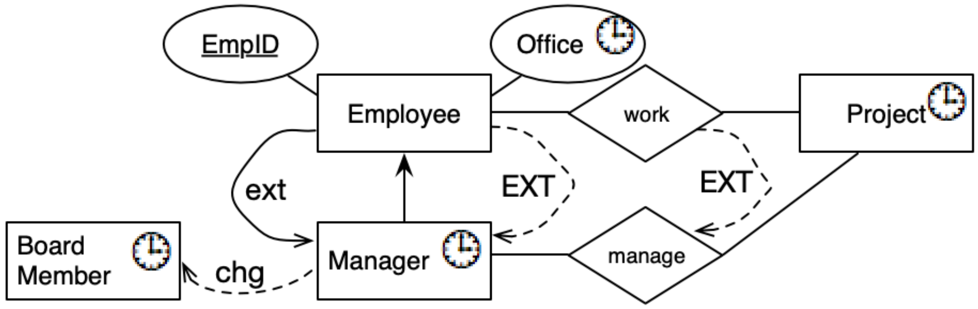


Figure 2. Example of a temporally extended ER diagram using TREND.

Figure 3 shows three examples of temporal entities and changes in the past. The clock icons indicate that an undergraduate is not always an undergraduate, and similarly for postgraduate, tadpole, frog, lawyer and articled clerk (every lawyer has to work as an articled clerk for 2 years before they can be a lawyer.) The arrows all have solid shafts indicating every frog must have been a tadpole before, every postgrad must have been an undergrad before, every lawyer must have been an articled clerk beforehand. Each diagram could have a CHG arrow added with a dashed shaft, as evolution in the future is not guaranteed for any of these cases: tadpoles can die, and many undergraduates and articled clerks fail to advance to postgraduate (or lawyer) status.

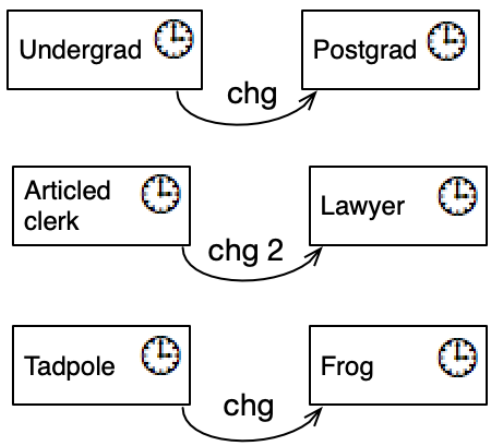


Figure 3. Examples of DEV- transitions using TREND.

Figure 4 is given to see if you can understand the notation without relying on common knowledge of the real world. Here M, P and S are entity types. If an entity is of type P at any point in time then it is always of type P. If an entity is of type S at any point in time then it is always of type S. If an entity is of type M at any point in time then it may not be of type M at some other time. An M entity may have an attribute N. If an M entity has an attribute N at any point in time it may sometimes not have an attribute N i.e. it may not always have an attribute N. A P entity may also become an M at some time and still remain a P. Every M entity must also have been a P at some time and still is a P. Q and R are 2 relationships that may hold at times between S entities and P entities. If a P and an S are related through R at any point in time they may not always be related through R. If a P and an S are related through Q at any point in time they may not always be related through Q. If a P and an S are related through Q at any point in time then they may have been related through R at another point in time and if so that relationship has to have been 2 time units beforehand.

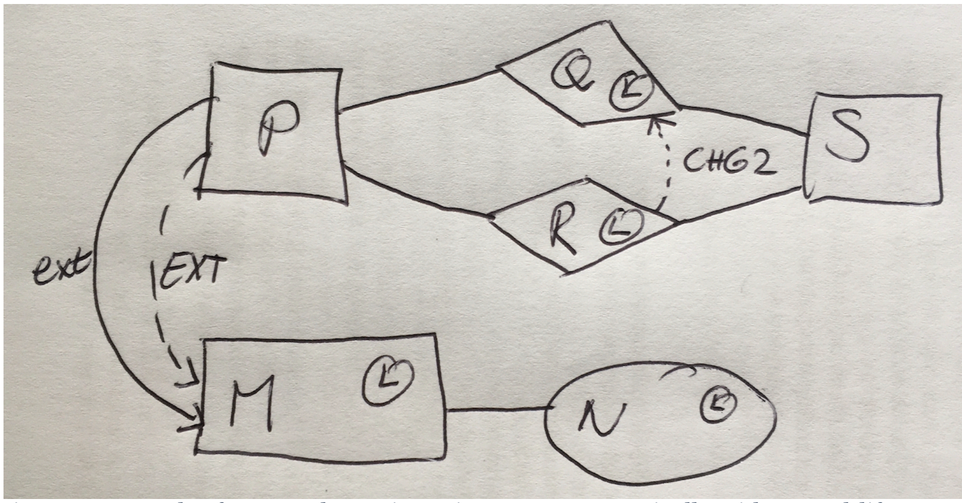


Figure 4. Example of temporal notation using TREND, generically without real-life examples.

4. Talking about temporal models and TREND

One may prefer visualisations of the information, but it also may be useful to be able to talk about the temporal information in a systematic way and some domain experts do prefer text over diagrams. Either way, generating natural language sentences from a conceptual model can be done automatically with a *controlled natural language* in software that then outputs natural language sentences. For instance, the dynamic extension among entity types can be captured with a template 'A(n) …. may also become a(n) …', so that with the sample model in Figure 2, it will fetch the elements of the EER diagram to generate sentences; in this case, it will generate the sentence 'An Employee may also become a Manager'. More examples are included in Figure 5.

Once the value for EmpID is set, it cannot change anymore.
An Employee may also become a Manager.
Each Manager was already an Employee.
Employee works for Project may be followed by
Employee manages Project some time later.
A Board Member may have been a Manager before, but is not a Manager now.

Figure 5. Examples of sentences generated from the sample diagram in Figure 2, using a controlled natural language for TREND.

References

1. Gianni, D., Bocciarelli, P., D'Ambrogio, A.: Temporal capabilities in support of conceptual process modeling using object-role modeling. In: Proc. of DEVS Inte- grative'14 (2014)
2. Keet, C.M., Ongoma, E.A.N.: Temporal attributes: their status and subsumption. In: Proc. of APCCM'15. CRPIT, vol. 165, pp. 61–70. CRPIT (2015)
3. Khatri, V., Ram, S., Snodgrass, R.T., Terenziani, P.: Capturing telic/atelic tem- poral data semantics: Generalizing conventional conceptual models. Transactions on Knowledge and Data Engineering 26(3), 528–548 (2014)
4. Artale, A., Parent, C., Spaccapietra, S.: Evolving objects in temporal information systems. Annals of Mathematics and Artificial Intelligence 50(1-2), 5–38 (2007)
5. Keet, C.M., Berman, S. Determining the preferred representation of temporal constraints in conceptual models. 36th International Conference on Conceptual Modeling (ER'17). Mayr, H.C., Guizzardi, G., Ma, H., Pastor., O. (Eds.). Springer LNCS vol. 10650, 437-450. 6-9 Nov 2017, Valencia, Spain.
6. Keet, C.M. Natural language template selection for temporal constraints. CREOL: Contextual Representation of Events and Objects in Language, Joint Ontology Workshops 2017, 21-23 September 2017, Bolzano, Italy. CEUR-WS Vol. 2050. 12p.