Ontologically Founded Design Patterns for Situation Modeling

Patryk Burek
Independent *burek@*
Lublin, Poland *informatik.uni-leipzig.de*

Frank Loebe
ScaDS.AI Dresden/Leipzig, Leipzig University *frank.loebe@*
Leipzig, Germany *informatik.uni-leipzig.de*

Ralph Schäfermeier
IMISE, Leipzig University *ralph.schaefermeier@*
Leipzig, Germany *imise.uni-leipzig.de*

Alexandr Uciteli
IMISE, Leipzig University
Leipzig, Germany *auciteli@imise.uni-leipzig.de*

Bartosz Kondracki
Medical University of Lublin
Lublin, Poland *bartosz.kondracki@umlub.pl*

Heinrich Herre
IMISE, Leipzig University
Leipzig, Germany *heinrich.herre@uni-leipzig.de*

Abstract

Situation modeling is a common challenge in modeling systems in non-trivial domains. The General Formal Ontology (GFO) is a top-level ontological theory that includes various notions for representing complex temporally extended situations. It has been used for situation modeling in diverse biomedical domains. We have analyzed and compared such GFO-based application cases. In accordance with this study, we present derived ontology-based design patterns as a conceptual toolset for situation modeling.

Keywords: conceptual modeling, situation modeling, design pattern, GFO

1. Introduction

Situations are ubiquitous entities in reality. Many real-world information systems and application scenarios demand the processing of information of complex situations, often also evolving over time. Situations are commonly perceived as integrated wholes [10], yet having an internal structure of multiple objects in interrelations that are of relevance in that situational context. There exist numerous analyses of situations, e.g., in philosophy [12], [18], and linguistics [2]. The complexity of understanding 'situation' is reflected in practical modeling in that it is effortful, requires many decisions, and it benefits from experience of the modeler.

Design patterns for recurrent design and modeling issues are a well-established means in software engineering. In ontology engineering, ontology design patterns have been defined as "a template to represent, and possibly solve, a modelling problem" [9], cf. [4]. In this work, a design pattern for situations is a representational blueprint based on situations, a selection of participating objects, and their relevant interrelations and qualities. Such patterns are created with the aim to ease situation modeling.

Beyond the value of information systems modeling in general, we argue that conceptual modeling patterns grounded in ontological theories have additional advantages. The grounding allows for high-quality models that can be extended and that can be integrated more easily with further models grounded in the same ontology. Despite many approaches and frameworks for modeling situations (up to ontology design patterns), we see a lack of appropriate modeling patterns that are properly grounded in established ontologies.

The General Formal Ontology (GFO) [10, 11] has been successfully applied in several projects involving situation modeling, see [14, Sect. 5], partially with differing terminology and building on different (though compatible) variants of the GFO theory. For the distillation of design patterns, so far we investigated four of those use cases, all in non-trivial biomedical domains: (1) cell tracking experiments [5, 6], (2) surgical navigation [15, 16], (3) medical risk assessment [17] and (4) pregnancy (not yet published).

The current paper reports on a step towards our overarching objective of establishing harmonized patterns for situation modeling that are based on GFO. We synthetize variants in prior work into a family of design patterns, thereby making use of a revised terminology. Those patterns can be used interchangeably, depending on the modeling context. They further allow for capturing situations temporally at time points or as time-extended entities.

Section 2 presents a first set of GFO-based situation modeling patterns as the main contribution. Related work on situation patterns is considered with hindsight in Section 3. Section 4 concludes the paper with a brief discussion and an outlook on future work.

2. Situation Design Patterns Founded on GFO

We identified several commonalities in the way in which situations are modeled (including in combination with which related kinds of entities), but also some differences. Despite targeting patterns that are founded on GFO, we aim at only minimal ontological commitments and adapt terminology to conceptual modeling, both to maximize adoption. The minimal ontological commitment involves the notions of *objects*, *qualities* and *relations*. An *Object* (aka gfo:Continuant) is a mutable entity with a finite lifespan, in contrast to an immutable, time-extended process (aka gfo:Occurrent). A *Quality* links an entity to a value specifying its characteristic, while *Role* describes an entity in the context of a *Relation*.

2.1. Object-Situation Patterns

Situations—complex entities comprehended as wholes [2], [11]—are often equipped with labels, say, political situations like World War 2. The latter is highly intricate, but already facts exhibit an internal structure. Note that we assume situations with at least one object.

A simple situation—i.e., a fact—can be either a single object together with one of its qualities, or two or more objects linked with a relator (the GFO term for a relation instance). In the cell-tracking domain, two cells touching each other exemplify a fact composed of two objects and a relator. Yet situations can be wholes that are more complex than facts, involving multiple objects and facts. For example, beyond just two cells, consider multiple cells that are interrelated in a bigger situation, up to snapshots of a full cellular genealogy. Finally, situations can also be quality bearers in GFO; e.g., a risk degree may be attributed to a pregnancy situation of Mary. Against this background, Figure 1 depicts our proposed *Object-Situation Patterns* by integrating three levels of representational granularity.

The first level alone (above the upper dotted line) constitutes the *Simple Object-Situation Pattern*, which models a situation as a named/identified collection of participating objects. For both, the situation and its objects, qualities can be captured. Already this simple object-situation pattern is applicable in many cases, where no further details on the interplay between objects, or between objects and the situation need to be expressed. To leave even the object(s) of the situation implicit in a model, qualities of the object(s) can be attached directly to the situation. The hasParticipantQuality relation—as used in the risk identification project [17]—allows for distinguishing such object-related qualities from genuine characteristics of the situation; e.g., it may link the age of a patient directly to a risk situation. Its derived semantics can be given via an ontological usage scheme, cf. [13].

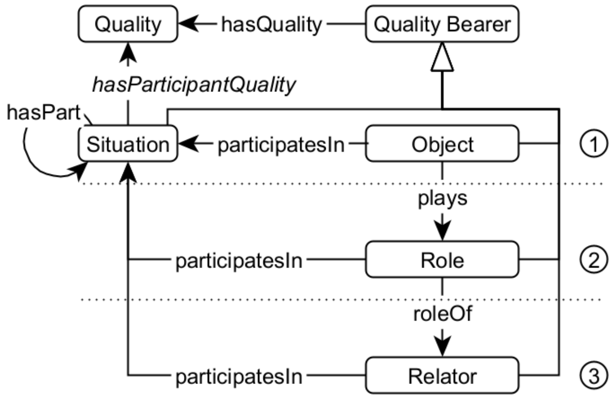

Fig. 1. Object-situation patterns, displayed in three levels of representational detail (numbered at the right). A situation model may comprise a situation with (1) just objects or additionally include (2) roles up to possibly (3) relators. Each of those model elements is subject to being equipped with qualities.

Levels 2 and 3 of Figure 1 turn the simple object-situation pattern into the *Extended Object-Situation Pattern*. This incorporates either only roles (Level 2), or both roles and relators (Level 3). Commonly it is sufficient to model only certain roles, then leaving relators as their contexts implicit. Objects can also play roles within the situation itself, and the same object can play different roles in the same situation. Qualities can be attributed to roles and/or relators in a model, analogously to objects and situations at Level 1.

2.2. Time-Indexed Situation Patterns

The object-situations patterns are agnostic about time, i.e., at best they capture situations with an implicit reference to time. As time information is essential in many cases, we propose *Time-Indexed Situation Patterns* based on time entities and temporal entities in GFO. Figure 2 shows the relevant distinctions and relations, with *Time Points* (gfo:TimeBoundary) and *Time Intervals* (gfo:Chronoid) at the bottom, indexing *Presentic* and *Time-Extended Entities* via the *time* relation. Note that Time-Extended Entity encompasses Object (and gfo:Continuant), Presentic Entity includes snapshots of objects (gfo:Presential).

Time-indexed situation patterns result from amalgamating object-situation patterns with entities in Figure 2. Through the left half, thus focusing on one time point, the *Presentic Situations Pattern* enhances the object-situations pattern (Level 1 of Figure 1) by adding a time index of type time point to quality bearer. Thereby we view all modeling elements as presentic entities existing at one time point. A usual example is an observation at a single time point, e.g., a single microscopy picture or a single blood test. That situation and all objects, roles and relators involved are assumed to exist at the very same time point.

However, many phenomena to be modeled are extended over time. In analogy, the *Time-Extended Situations Pattern* indexes Quality Bearer with Time Interval and reinterprets all modeling elements of the object-situations pattern as time-extended. Next, observe that often information about time-extended entities is actually computed or inferred from point-indexed raw data. For instance, the raw data of time-lapse experiments in cell tracking usually contains only presential information, but nothing genuine about the cells as persisting objects. Hence, an even more complete picture interlinks both types of time-indexed patterns, where time-point-based raw data give rise to multiple presentic situations, which may be understood as time-indexed reifications of a time-extended situation.

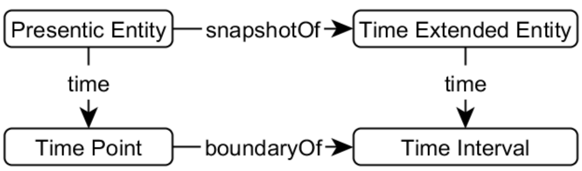

Fig. 2. Time and temporal entities that serve as foundation for time-indexed situation patterns.

There are different options for interlinking presentic and time-extended situations. Figure 2 contains the *snapshotOf* relation between Presentic and Time-Extended Entity. That can be directly used to link elements in instantiations of presentic and time-extended situation patterns. A natural assumption in this respect is that (presentic) objects are linked to (time-extended) objects, etc., i.e., faithfulness in accordance with the types of the elements is presumed. Another kind of interlinking situations refers to sequences of them. Presentic situations may follow one another, e.g., referring again to raw data with distinct time stamps. A relation *followUp* can be used to express corresponding connections across such situations and their elements. It can further serve as a foundation for eliciting their time-extended counterparts (and possibly snapshot connections). Completing the inclusion of *followUp* into some patterns and investigating forms of integrating presentic and time-extended situations that are more sophisticated remains future work.

3. Related Work

Design patterns—an established means for guiding programmers and modelers on recurrent matters—gained much traction in software engineering, e.g., via Gamma et al. [8], as well as in ontology engineering, cf. [4], [9]. Since the mid-2000s, a sub community on ontology design patterns established itself and a well-known portal[1] with 240 patterns.

Therein exists a modeling issue 'Situation classification' with five patterns, all part of the DUL ontology[2], one of which is labeled 'Situation'[3]. Formally, that 'Situation' pattern with one class dul:Situation and one relation pair dul:hasSetting/dul:isSettingFor is very distinct from our proposed patterns. Conceptually, some overlap may be drawn from "It can also be seen as a 'relational context' [...]"[3], but especially requiring a dul:Description for each dul:Situation, that pattern is of a very different nature.

Narrowing the context further to design patterns founded in top-level ontologies (TLOs), first, we are not aware of any other design patterns focusing on situations that are based on GFO [14], one of the systems covered in a 2022 special issue on active TLOs. This unawareness largely still applies when considering the other six TLOs therein, with two reservations. One is that DUL (above) has been created with some recourse to DOLCE. The other refers to the gentle Unified Foundational Ontology (gUFO) [1], which covers a notion of gufo:Situation, as well, with a conceptually very similar approach and referring to patterns [1, Sect. 2.9]. However, the patterns proposed in the present paper seem to make a case between gufo:Situation and its five direct subclasses, e.g., gufo:QualityValueAttributionSituation and gufo:TemporaryRelationshipSituation. The former class includes factual and counterfactual/possible situations, while the five subclasses select and refine aspects on which object-situations patterns and their time-indexed extensions are based.

Overall, the patterns presented herein share with [1], [7] the idea of modeling situations through their participants. Beyond that general idea, our patterns address varying levels of detail, for distinct modeling needs and optionally without modeling participants explicitly. Furthermore, the grounding in the time ontology of GFO [3] enables a particular modeling of phenomena of change and situation composition.

4. Conclusions and Future Work

Situations are complex entities, which presents a challenge for modelers across various domains. This observation indicates that modeling situations can benefit from cross-domain tools. The General Formal Ontology (GFO) is a domain-independent ontological framework aimed at supporting conceptual modeling since its inception [11]. It has been applied in several biomedical projects involving situation modeling, a. o., cell tracking and surgical navigation. Those projects employ aspects of GFO's situation theory and utilize it to build domain models that serve as key components of the developed systems.

[1] http://ontologydesignpatterns.org
[2] http://ontologydesignpatterns.org/wiki/Ontology:DOLCE%2BDnS_Ultralite (DOLCE+DnS UltraLite (DUL))
[3] http://ontologydesignpatterns.org/wiki/Submissions:Situation

Targeting conceptual modeling, this work-in-progress paper outlines several design patterns for situations, with an ontological foundation in GFO as their novelty. We demonstrate that several basic patterns can be established on the GFO theory. Showing in-depth effects of GFO underneath and an evaluation of that foundation remain subject to an extended article. Altogether, we see these patterns as steps toward a toolset for modeling situations, their participants, qualities, dynamics, and the changes they undergo over time.

Moreover, we plan the inclusion or tight integration of the presented and extended patterns with modules of GFO 2.0 [14]. This will require further refinements of the underlying theory, e.g. extensions associated with cross-situation relations such as 'part-of' and others.

Acknowledgments

The authors did not make any use of AI software that goes beyond verifying spelling, grammar and punctuation.

References

1. Almeida, J.P.A., Falbo, R.A., Guizzardi, G., Sales, T.P.: gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO). https://purl.org/nemo/doc/gufo. UFES (2019)
2. Barwise, J., Perry, J.: Situations and Attitudes. MIT Press, Cambridge, MA, USA (1983)
3. Baumann, R., Loebe, F., Herre, H.: Axiomatic Theories of the Ontology of Time in GFO. Applied Ontology 9 (3-4), 171–215 (2014)
4. Blomqvist, E., Sandkuhl, K.: Patterns in Ontology Engineering: Classification of Ontology Patterns. In: ICEIS, Vol. 3. Insticc, Setubal, Portugal (2005)
5. Burek, P., Scherf, N., Herre, H. A Pattern-based Approach to a Cell Tracking Ontology. Procedia Computer Science, 159, 784–793 (2019)
6. Burek, P., Scherf, N., Herre, H. On the Formal Representation and Annotation of Cellular Genealogies. In: EKAW 2020. LNCS 12387. Springer, Cham (2020)
7. Costa, P.D., Guizzardi, G., Almeida, J.P.A., Pires, L.F., van Sinderen, M.: Situations in Conceptual Modeling of Context. In: Tenth IEEE EDOCW, IEEE, Los Alamitos, CA, USA (2006)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, Massachusetts, USA (1994)
9. Gangemi, A.: Ontology Design Patterns for Semantic Web Content. In: ISWC. pp. 262–276. Springer, Berlin, Heidelberg (2005)
10. Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., Michalek, H.: General Formal Ontology (Version 1.0). Onto-Med Report N. 8. Leipzig University, Germany (2006)
11. Herre, H.: General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling. In: Poli, R., Healy, M., Kameas, A. (eds.) Theory and Applications of Ontology: Computer Applications. pp. 297–345. Springer Netherlands, Dordrecht (2010)
12. Ingarden, R.: Der Streit um die Existenz der Welt (in German). Niemeyer, Tübingen (1964, 1965, 1974)
13. Loebe, F.: Ontological Semantics. PhD Thesis. Leipzig University, Leipzig, Germany. (2015)
14. Loebe, F., Burek, P., Herre, H.: GFO: The General Formal Ontology. Applied Ontology 17 (1), 71–106 (2022)
15. Schäfermeier, R., Herre, H., Paschke, A. Ontology Design Patterns for Representing Context. In: Advances in Pattern-based Ontology Engineering. Studies on the Semantic Web 51, pp. 183–203. IOS Press, Amsterdam (2021)
16. Tahar, K., Uciteli, A., Röppischer, P., Herre, H., Siemoleit, S.: Ontology-Guided Markerless Navigation and Situational Awareness for Endoscopic Surgery. In: German Medical Data Science (GMDS), pp. 83–87, IOS Press, Amsterdam (2018)
17. Uciteli, A., Neumann, J., Tahar, K., Saleh, K., Stucke, S., Faulbrück-Röhr, S., Kaeding, A., Specht, M., Schmidt, T., Neumuth, T., Besting, A., Stegemann, D., Portheine, F., Herre, H.: Ontology-based Specification, Identification and Analysis of Perioperative Risks. Journal of Biomedical Semantics 8 (1), 36 (2017)
18. Wittgenstein, L.: Tractatus Logico-philosophicus. Routledge & Kegan Paul, London, England (1922)