Chapter 2
30 Years of Contributions to Conceptual Modeling

Jordi Cabot, Cristina Gómez, Maria-Ribera Sancho and Ernest Teniente

Abstract This chapter is aimed at summarizing the contribution of Antoni Olivé to the field of conceptual modeling over the last three decades. It starts with his initial proposals around the year 1986 and it finishes with his most recent, not to say current, work on the field. The summary encompasses different topics, beginning with the deductive approach to conceptual modeling and its application to deductive databases, evolving later to object-oriented conceptual schemas and, more recently, to conceptual-schema centric development. All in all, the trajectory covers a wide range of topics, all of them of great importance at the time they were treated, and has meant an important advance of the knowledge in this area during all these years.

2.1 Introduction

Trying to summarize 30 years of research of Antoni Olivé at the Universitat Politècnica de Catalunya in just a few pages is not an easy task. On the one hand, because of his huge contribution to the field, with more than eighty papers (most of them in the most prestigious journals and conferences in the field), fourteen PhD thesis advised and multitude of talks and keynotes. On the other, because it is also a summary of our life. At least of its academic part. We feel, directly or indirectly, disciples of

Jordi Cabot
ICREA, e-mail: jordi.cabot@icrea.cat
Universitat Oberta de Catalunya

Cristina Gómez
Universitat Politècnica de Catalunya,e-mail: cristina@essi.upc.edu

Maria-Ribera Sancho
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center, e-mail: ribera@essi.upc.edu

Ernest Teniente
Universitat Politècnica de Catalunya, e-mail: teniente@essi.upc.edu

J. Cabot et al. (eds.), *Conceptual Modeling Perspectives*,
https://doi.org/10.1007/978-3-319-67271-7_2

Antoni and we owe much of our research to what we have learned from him during all these years. Therefore, since we are so thankful to him, it is very difficult for us to be fully objective although we will do our best to be as fair as possible while summarizing his most important contributions.

Antoni carried out all his research as a Professor at the Universitat Politècnica de Catalunya. During all these years, more than thirty people were members of his research group at one time or another. Therefore, it would be unfair saying or thinking that Antoni did all the contributions alone. In fact, he always believed in the strength of the group, above the specific individuals, and for that reason he always tried to be fair in relation to the contributions of each one of us. Nevertheless, and despite the risk of not being always understood, we will only use his name while describing the different proposals since he has been an important contributor to all of them and because this chapter is a tribute to his career. For this same reason, we have not considered those contributions to conceptual modeling from people in the group not having Antoni as a coauthor.

This chapter is divided into four different sections, each one of them addressing a different period and a different center of interest for research. The first period (1986-1989) was devoted to analyze the deductive approach to conceptual modeling. Then, Antoni moved towards techniques for deductive databases and deductive conceptual models (1989-1999) to make later an important turn and move on to object-oriented conceptual schemas (1999-2007). Finally, the last periode (2007-present) has been dedicated to deepen into conceptual schema-centric development and contributing to a number of related research problems.

2.2 The Deductive Approach to Conceptual Modeling

The beginning of the research of Antoni Olivé in conceptual modeling of information systems goes back to the remote year 1986, i.e. almost 30 years ago. At that time he published a seminal paper comparing the operational and the deductive approaches to conceptual information systems modeling [12].

Intuitively, the main feature of the deductive approach is that the basic part of the Information Base (IB) (i.e. the information explicitly stored) contains only the events that happen in the domain — aka *Universe of Discourse (UoD)*, as named in the paper. All other informations of the IB are deduced by means of derivation rules which allow to define the knowledge about the concepts in the domain from the stored events. Time plays also a major role in this approach because for every information about the UoD is associated with a time point which states the time when the information holds.

As an example, assume that an information system provides with the ability of starting projects by means of an event like: *startProj(proj,end,dept,t)*; where *end* is the expected date to finish the project, *dept* is the department running it and *t* is the time at which the project starts. Then, from this event we could define active projects as follows in the deductive approach:

```
activeProj(p,t) ← startProj(p,e,d,t1), t1<=t, ¬cancelled(p,t)
```

i.e. project *p* will be active at time *t* if it was started at a previous time and it has not yet been cancelled.

As a conclusion of this analysis, Antoni claimed that “deductive languages show a number of advantages, which might justify to pursue their development at a level comparable to that of operational languages. Model verification and efficient implementation methods would be the main issues for research (...)”. It is worth mentioning that, as we will see in Section 2.3, Antoni was premonitory since a significant amount of research was devoted later to these topics by his first PhD students. Not to mention also that this paper will never be forgotten by his closest collaborators at that time because of the efforts they had to devote to understand and assimilate this deductive approach.

These ideas guided the initial steps of Antoni’s research and lasted for almost five years. He did not had properly a research group at that time (at least as they are currently understood) but work in this area gave rise in 1987 to the PhD thesis of Jaume Sistac†, whose main ideas were related to the automatic generation of information system prototypes from a deductive conceptual model [29].

Also in 1987, Antoni published a first proposal about the design and implementation of information systems from deductive conceptual models (DCMs) [13], one of the open areas of research he identified from the beginning. In this paper, he presented a formal method to derive from a DCM a new model, called the *internal events model*, which is much easier to implement. This model was a useful basis from which several design alternatives could be systematically developed and evaluated. Possible uses of the internal events model in data base and transaction design were also outlined.

The last claim of the paper was, again, premonitory: "we also expect that the internal events model (...) can be useful in the field of deductive databases although this has not been elaborated in the paper". Thirty years later, the main notions provided by the internal event rules are still being used and applied to different settings such as handling updates in UML/OCL schema or in Description Logics.

2.3 Techniques for Deductive Conceptual Models

After this initial period, Antoni expanded his research and proposed techniques based on the deductive approach and applied them to three different areas: deductive databases, deductive conceptual models of information systems and, finally, to object-oriented deductive conceptual models.

His research group grew significantly during this period, which lasted approximately from 1989 to 1999, and which gave also as a result the doctoral thesis of six of his PhD students at that time: Ernest Teniente (1992), Toni Urpí (1993), Maria-Ribera Sancho (1994), Dolors Costal (1995), Joan Antoni Pastor (1997) and Carme Quer (1999).

2.3.1 Deductive Databases

The name *deductive*, in the deductive approach to conceptual modeling, came from its similarity to *deductive databases*, where deductive rules play also a major role. Moreover, as we have just seen, the idea of applying the internal events model to deductive databases was already stated when this model was proposed. Therefore, it is not surprising that he also significantly contributed to this field. In particular, addressing the problems of ***change computation*** (and its applications to *integrity checking, materialized view maintenance and condition monitoring*) and ***(consistency preserving) view updating***.

2.3.1.1 Change Computation

A deductive database consists of three finite sets: a set of *facts* that are explicitly stored in the database; a set of *deductive rules*, that allow to define new knowledge in the form of derived predicates from base and other derived predicates; and a set of *integrity constraints*, specified in terms of base or derived predicates and defining conditions that every state of the database should satisfy.

Change computation refers to the general problem of computing the changes induced by an update on the base factes on one or more derived predicates. Efficient change computation is essential in several capabilities of a deductive database, such as integrity constraints checking, view maintenance or condition monitoring. These problems are still relevant and need to be solved in all contexts that use any kind of rules to define intensional information in terms of that explicitly stored.

One of the most cited papers from Antoni was published in this topic in 1992 [36]. This paper proposes a general method for change computation that can be applied in all database capabilities mentioned above. It is based on the use of transition and internal events rules, which explicitly define the insertions, deletions and modifications induced by a database update on the contents of derived predicates. The method computes the changes once the database has been updated, providing more efficient ways of change computation than those of previous research. These ideas were later extended and applied also to active databases in [35].

2.3.1.2 Consistency-Preserving View Updating

Transition and insertion event rules where also applied to deal with the problem of consistency-preserving view updating [30, 31]. View updating is related to the problem of translating a request for updating the contents of a derived predicate in terms of updates of the underlying base facts.

However, some of the obtained translations may not satisfy all the integrity constraints. For this reason, view updating is usually followed by an integrity enforcement process in order to ensure consistency of the data and this is why we call the whole approach consistency-preserving. In particular, [30, 31] follow an integrity maintenance approach aimed at finding repairs, i.e. additional updates of the base facts, for each constraint violation so that the final set of solutions is ensured to satisfy all constraints. In general, there may be several solutions and the user must select one of them. In some cases, no such repair exists, and the view update must be rejected.

As a result, this work resulted in a method that uniformly handles both insert and delete requests and that allows for complex updates, such as mixed multiple updates or modification requests. It also naturally encompasses several additional features like preventing side effects on other views, repairing inconsistent knowledge bases or maintaining transition integrity constraints. All in all, the method extended the functionalities of those previously proposed and its contribution has been extensively recognized by many citations. It is worth mentioning also that the method was proved to be sound and complete, but termination was not guaranteed and efficiency and complexity issues were not considered in the proposal.

2.3.2 Deductive Conceptual Models

Antoni's techniques proposed in this area where mainly concerced with the validation of conceptual schemas and with the (semi)automatic generation of transactions from the specified schema. The main contributions of these techniques are summarized in the following.

2.3.2.1 Reasoning about Deductive Conceptual Models

Reasoning on a schema has always been concerned with determining whether the schema is correct or not (i.e. verification) and whether it satisfies the user needs and requirements (i.e. validation). This is one of the most important and crucial problems in information systems engineering since determining errors at the early stages of information systems development is directly related to improving the quality and the adequacy of the final system.

Antoni's main proposals for validating DCMs by means of reasoning were published in [8]. The proposed method uses SLDNF resolution as proof procedure and plan generation techniques developed in the Artificial Intelligence field to perform reasoning on the schema. Its main capabilities are the following:

- Given an initial and a target state of the IB, together with a sequence of external events, check whether the sequence is able to perform the transition between both states.
- Given an initial and a target state, obtain one or more sequences of external events (plans) able to perform the transition between both states.

This method was not only able to reason about DCMs but it had also the full power of the methods developed so far for the traditional operational approach to conceptual modelling. Moreover, the reasoning capabilities it provided were helpful and helped to improve the validation task of conceptual models of information systems.

2.3.2.2 Validating Conceptual Specifications

Work on validation was also carried out from a wider perspective to propose a method for explaining the behaviour of conceptual models of information systems [22]. This method assumed a conceptual model in terms of information base structure (with base and, optionally, derived information), integrity constraints and transactions. Therefore, the method could be adapted to most existing methodologies.

This method contributed to model validation by providing explanations about the results of model execution. Specifically, it can explain, in several complementary ways, why some facts hold (or do not hold) in the IB; why some facts have been inserted to (or deleted from) the IB when applying a transaction; how some intended effect on the IB can be achieved; and what would have happened if some other inputs were given (hypothetical explanations).

Answers to some of the above questions were given by some existing explanation systems, but properly extended by providing answers to questions about derived facts, to questions about how a fact can be made true or false, and to hypothetical questions. The method grows mainly on results in the field of deductive databases and this was useful to show how the procedures developed in that field for explaining the results of queries, or their failure, and for consistency-preserving updates may be useful for behaviour explanation of conceptual models. In this sense, this work linked the fields of DCMs and deductive databases.

2.3.2.3 Supporting Transaction Design

Grounded on his previous work on deductive databases, Antoni contributed also to deriving transaction specifications from deductive conceptual models of information systems [28]. This work used a logic-based language for the specification of conceptual models and applied logic-based techniques for the automatic generation of a system design from them. The idea was to build a a transaction for each external event that should be handled by the system. Preconditions of this transaction were then determined from the integrity constraints in the schema while the postconditions were drawn from an analysis of its deductive rules.

This work was extended in [24], with the goal of automatically deriving a transaction specification integrating in a uniform manner the updating of base and derived information and the checking and maintenance of integrity constraints within the IB of a DCM. In this way, the obtained transaction specifications may guarantee at definition time that the consistency of the IB is preserved. Therefore, no enforcement has to be performed at run time to ensure it. When there are several possible solutions, the method derives all of them and the designer has to intervene to select the most appropriate one to apply.

It is worth noting that it was not alway possible to derive the transaction specification satisfying a given update because of the well-known undecidability of integrity maintenance and view updating.

2.3.3 Object-Oriented Deductive Conceptual Models

Antoni's first attempt to evolve towards object-orientation was proposed in [26], where he provided a combination of the deductive and object-oriented approaches, by which the IB predicates were grouped using the concept of object. Therefore, one of the main goals of this work was to present the main components of an object-oriented deductive approach to conceptual modeling of information systems. This approach did not model object interaction explicitly. However, a method for deriving these interactions was outlined in the paper.

Based on these results, the paper discussed whether explicit object interaction is a desirable feature of conceptual models and it ended up by showing that most difficulties in the modeling of the dynamic aspect with object-oriented methods existing at that time arose because they tried to model explicitly the interaction among objects, which was shown in the paper not to be necessary from a conceptual point of view.

2.4 Object-Oriented Conceptual Schemas: definition and evolution

During the period between 1999 and 2007 Antoni Olivé focused his research on the essential aspects and principles of conceptual modeling, on the formal basis of conceptual schemas and on the evolution of conceptual schemas. The result of this period of intense research was the publication of his book "Conceptual Modeling of Information Systems" [19] in 2007.

His research group continued growing during this period, which gave as a result the doctoral thesis of two of his PhD students: Juan Ramón López (2001) and Cristina Gómez (2003).

2.4.1 Definition

At the end of the nineties Antoni Olivé began to deepen in the study of object-oriented conceptual modeling. Conceptual modeling is defined in [19] as the activity to elicit and describe the general knowledge a particular information system needs to know. The main objective of conceptual modeling is to obtain the description of the conceptual schema, formed by the structural schema and the behavioral schema. Antoni's passion and rigor in the study and analysis of conceptual modeling constructs used in conceptual schemas resulted in a considerable number of publications in top international conferences and journals.

2.4.1.1 Entity types.

Entity and relationship types, the most important constructs in structural conceptual schemas, attracted Antoni's attention in several works. An entity type is a concept whose instances at a given time are identifiable individual objects that are considered to exist in the domain at that time [19]. In [5], Antoni reviewed the definition of entity types derived by symbol-generating rules. These types appear frequently in conceptual schemas but most conceptual modeling languages, like the UML and ORM, did not allow their formal definition. He proposed a method for the definition of entity types derived by symbol-generating rules based on the fact that these types can always be expressed as the result of the reification of a derived relationship type.

2.4.1.2 Relationship types.

A relationship type is a concept whose instances at a given time are identifiable individual relationships between objects that are considered to exist in the domain at that time [19]. There exist some relationship types, called generic relationship types, that appear in many structural conceptual schemas and even several times in the same schema that have a particular meaning. Typical examples are IsPartOf or IsMemberOf. Antoni studied generic relationship types in [15] and proposed two alternatives methods for their representation. Moreover, he described the contexts in which one or the other representation is more appropriate, showed their advantatges and the described the adaptation of the methods to the UML.

2.4.1.3 Temporal Aspects of Entity and Relationship Types.

Temporal aspects of structural conceptual schemas were investigated by Antoni in [6] and [14]. In [6] he proposed a standard extension of the UML that allows the designer to define a set of temporal features of entity and relationship types appearing in a conceptual schema. Moreover, he also defined several temporal operations to refer to any past state of the information base that may be used to deal with UML/OCL as if it were a temporal conceptual modeling language. He also presented a method for the transformation of a conceptual schema in this extended language into a conventional one. The temporal view of relationship reification was presented in [14]. Refying a relationship consists in viewing it as an entity [19]. Antoni generalized previous work on reification, and proposed three temporal reification kinds.

He defined the characteristics of the entity types, and of their intrinsic relationship types, produced by each reification kind. The result of his work may be applicable to any temporal conceptual model.

2.4.1.4 Derived Types.

An entity type or relationship type is derived when its instances need not to be explicitly represented in the information base, because the information system may derive (i.e. infer or calculate) them at any time. For each derived type, there is a derivation rule, which is an expression that defines the necessary and sufficient conditions for an entity or relationship to be an instance of a given type [19]. In [16], Antoni proposed three methods for the definition of derivation rules in object-oriented conceptual modeling languages. The first method proposed applies to static rules, and associates each derived element with a defining operation. The specification of this operation is then the definition of the corresponding derivation rule. The second method applies to constant relationship types whose instances can be derived when the instances of one of its participant entity types are created and the third one deals with hybrid types, and defines their partial derivation rules. The three methods are adapted to the UML.

2.4.1.5 Integrity Constraints.

Integrity constraints are conditions that might not be satisfied in some states of the information base or by some events, but it is understood that the information system will include mechanisms to guarantee its satisfaction at any time [19]. A method that eases the definition of integrity constraints in object-oriented conceptual modeling languages was introduced by Olivé in [18]. The method propose to represent constraints by special operations called constraint operations. The formal specification of these operations is the definition of the corresponding constraints. The method allows the specialization of constraints and the definition of exceptions. The main application of the method is for static constraints. However, a variant of it can also be applied for creation-time and deletion-time constraints, two particular classes of temporal constraints. The method can be adapted to any object-oriented language.

2.4.1.6 Taxonomies.

In some cases instances of an entity type must also necessarily be instances of another entity type. This type of relationship between entity types is called IsA relationships. Entity types and their IsA relationships form a network structure called a taxonomy [19]. Antoni's extensive study of taxonomies produced two relevant publications in this area. The first one [9] deal with relationship type refinements, tha is, the specification of additional constraints when some of the participant entities are also instances of other entity types. In this paper, he characterized relationship type refinements in conceptual models with multiple classification, provided a graphical and textual notation for their specification, and gave their formal definition in logical terms. Moreover, he presented a set of necessary conditions to guarantee that a given set of refinements is valid. The second publication [23] analyzes the relationships between derived types and taxonomic constraints to see which taxonomic constraints are entailed by derivation rules and to analyze how taxonomic constraints can be satisfied in presence of derived types.

2.4.1.7 Events.

Antoni also focused his attention on the behavioral conceptual modeling. A method for modeling events as entities was proposed in [21]. The method makes extensive use of language constructs such as constraints, derived types, derivation rules, type specializations and operations, which are present in all complete object-oriented conceptual modeling languages. The method can be adapted to most object-oriented languages, including the UML.

2.4.2 Evolution

Another Antoni's main line of research during this period was the evolution of conceptual schemas. As he argued in [11], changes in the requirements of information systems should be defined and managed at the conceptual schema level, with an automatic propagation down to the logical database schema(s) and application programs. He proposed a framework for the evolution of temporal conceptual schemas of information systems. The framework uses a reflective architecture with two levels: a meta schema and schema, and two loosely coupled information processors, one for each level and it can be used to specify schema changes. The evolution of the partitions modeling construct in conceptual models with multiple specialization and classification, and considering base and derived entity types was analyzed in [10]. He provided a list of possible schema changes and, for each of them, he gave its formal specification.

In the same field, Antoni characterized in [20] the set of valid type configurations, taking into account the constraints defined by specializations and generalizations and considering multiple specialization, generalization and classification, as well as dynamic classification.He also analyzed the problem of determining the valid evolution of the type configuration of entities in the context of IsA hierarchies.

2.5 Conceptual Schema-Centric Development

In his most recent works, Antoni Olivé has expanded his research interests to study the role of conceptual modeling in the broader field of systems and software engineering and the additional challenges conceptual modelers would face in such context, specially regarding the quality and scalability aspects that models should comply to answer real-life industrial problems.

This vision was first outlined in his CAiSE keynote "Conceptual Schema-Centric Development: A Grand Challenge for Information Systems Research" [17]. He named the challenge "conceptual schema-centric development" (CSCD) in order to emphasize that the conceptual schema should be the center of the development of information systems.

Indeed, the goal of automating information systems building was already stated in the sixties but forty years later it is clear that the goal has not been achieved in a satisfactory degree. Antoni revisits this goal by emphasizing the key role conceptual schemas can play in it, now that standard modeling languages and platforms are available. He shows that to develop an information system it is necessary to define its conceptual schema and that, therefore, the CSCD approach does not place an extra burden on developers. In CSCD, conceptual schemas are explicit, executable in the production environment and the basis for the system evolution.

Obviously, this is an ambitious goal that involves solving as well a number of related research problems. Some of them were the focus on his work in the subsequent years as we will see below. This work was done in collaboration with a final generation of PhD students that completed their thesis during this period: Jordi Conesa (2008), Ruth Raventós (2009), Albert Tort (2012), Antonio Villegas (2013) and David Aguilera (2014).

2.5.1 Very large conceptual schemas

One of the major challenges is to be able to specify, understand and transform large conceptual schemas, required to model with enough level detail complex systems.

2.5.1.1 Specification of large schemas

In the past, most conceptual schemas of information systems have been developed essentially from scratch. However, Antoni explored an alternative approach that tries to reuse as much as possible the knowledge included in existing ontologies. Using this approach, conceptual schemas would be developed as refinements of (more general) ontologies. However, when the refined ontology is large, a new problem that arises using this approach is the need of pruning the concepts in that ontology that are superfluous in the final conceptual schema. He developed new automatic method for pruning ontologies [7] that can be adapted to most conceptual modeling languages and ontologies, though his approach takes as example the Cyc ontology.

Recently, he has also shown that the same approach can also be adapted to a more technological problem, which is the modeling of the microdata tagging a website content [33]. Similar in philosophy to the ontology pruning problem, here the goal is to prune schema.org vocabularies to derive the tags relevant for an individual website. Indeed, for large websites, implementing microdata can take a lot of time. In general, it is necessary to perform two main activities: (i) designing what he calls the website schema.org, which is the fragment of schema.org that is relevant to the website and (ii) adding the corresponding microdata tags to the web pages. Antoni's approach consists in using a human-computer task-oriented dialogue, whose purpose is to arrive at that design.

2.5.1.2 Understanding large schemas

Once you have built a very large schema, the next problem is how to effectively visualize and understand it. This problem appears in many information systems development activities in which modelers need to cooperate. While they need to have access to the global schema, most times their main focus and role in the collaboration is to take care of a small piece of the knowledge contained in that schema.

Therefore, Antoni proposed a method for filtering a fragment of the knowledge contained in a large conceptual schema [37]. In his method, once a user focuses on one or more entity types of interest for her task at hand, the method automatically filters the schema in order to obtain a set of entity and relationship types (and other knowledge) relevant to that task, taking into account the interest of each entity type with respect to the focus, computed from the measures of importance and closeness of entity types.

Later this method was extended to cover also the constraint expressions (also called schema rules) in the schema. Understanding such expressions is complex since the types they refer to may be located in very different places in the schema, possibly distant from each other and embedded in an intricate web of irrelevant elements. In [38], he described a method that, given a set of constraint expressions and a large conceptual schema, automatically filters the conceptual schema, obtaining a smaller one that contains the elements of interest for the understanding of the expressions.

Another factor that plays a role in this readability problem is the naming conventions for schema elements. The problem is significant because in general there are many elements that require a name, and the names given have a strong influence on the understandability of that schema. Following the same naming conventions across a large schema clearly helps modelers to identify the type and role of elements they are looking at. In [2], Antoni proposed a guideline for every kind of element to which a conceptual modeler may give a name in UML. The guideline comprises the grammar form of the name and a pattern sentence. A name complies with the guideline if it has that form and the sentence generated from the pattern sentence is grammatically well-formed and semantically meaningful.

2.5.1.3 Transforming large schemas

Many model manipulation operations involve a schema translation that reexpresses the input model in a new language / metamodel. Nowadays, more metamodels are defiend as instances of the OMG's MOF standard but the specification of the mappings and translations remains a difficult and error-prone task.

Antoni proposes a new approach to the schema translation problem. to make an extensive use of object-oriented concepts in the definition of translation mappings, particularly the use of operations (and their refinements) and invariants, both of which are formalized in OCL [27]. Then, these translation mappings can be used to check that two schemas are translations of each other, and to translate one into the other, in both directions. The translation mappings are declaratively defined by means of pre- and postconditions and invariants, and they can be implemented in any suitable language. From an implementation point of view, by taking a MOF-based approach, you have immediately a wide set of tools available, including tools that execute OCL.

2.5.2 Quality in modeling

In the CSCD approach, the final software implementation is (at least partially) derived from the initial conceptual schema. Therefore, the quality of the schema has a direct impact on the quality of the final system. As a result, another major focus of Antoni's work on the CSCD challenge has been devoted to ensure the quality of schemas.

A first step was to come up with a precise definition of quality properties for conceptual schemas. As he reported, in the literature, there are many proposals of quality properties of conceptual schemas, but only a few of them (mainly those related to syntax) have been integrated into the development environments used by professionals and students. A possible explanation of this unfortunate fact may be that the proposals have been defined in disparate ways, which makes it difficult to integrate them into those environments.

As a reaction to this situation, Antoni proposed a list of quality issues, which essentially are conditions that should not happen in a schema, and a unified method for their definition and treatment [1]. In this same work, he also showed that his method could be used to successfully describe most of the quality properties already described in the literature in an homogeneous way, which should facilitate building a catalogue of quality properties to be enforced in integrated development environments (IDEs). This is specially relevant since the support provided by current IDEs wrt the enforcement of quality criteria is still very limited [3] and, clearly, one of the most effective ways of increasing the quality of conceptual schemas in practice is by using an IDE that assists the designers in this matter.

An alternative to enforce quality properties on a conceptual schema is to test if the schema satisfies them. Testing is sometimes regarded as "poor man's verification" but still a very reasonable trade-off when the size or complexity of the schema hampers its verification in a reasonable amount of time. Antoni proposed a list of five kinds of tests that can be applied to conceptual schemas [32]. Two of them require schemas comprising both the structural and the behavioral parts, but he showed that it was possible and useful to test incomplete schema fragments, even if they consist of only a few entity and relationship types, integrity constraints and derivation rules. Tests are specified in CSTL, a language for writing automated tests of executable schemas written in UML/OCL. CSTL includes language primitives for each of the above kinds of tests. CSTL follows the style of the modern xUnit testing frameworks.

When testing, a more ambitious approach is to go from testing an existing model to using the tests to build the model itself, which is known as Test-Driven Development (TDD). TDD is an extreme programming development method in which a software system is developed in short iterations. Antoni proposed the Test-Driven Conceptual Modeling (TDCM) method [34], which is an application of TDD for conceptual modeling, and showed how to develop a conceptual schema using it. In TDCM, a system's conceptual schema is incrementally obtained by performing three kinds of tasks: (1) Write a test the system should pass; (2) Change the schema to pass the test; and (3) Refactor the schema to improve its qualities.

2.6 Conclusions

We have summarized in this chapter thirty years of contributions of Antoni Olivé to conceptual modeling. They began with fostering the deductive approach to conceptual modeling of information systems and by proposing techniques aimed at dealing with some of the most relevant problems in this field. The *internal events rules* were probably the most significant contribution at that time, and they were also successfully applied for handling several problems in deductive databases. This first period of research lasted for almost fifteen years, from 1986 to 1999.

Then, he progressively moved to object-oriented conceptuals schemas, field were he was contributing for about ten years, devoting a lot of efforts to set up the basis of the main components of conceptual schemas and analyzing how to facilitate its evolution over time. The leading landmark of this period was his seminal book *"Conceptual Modeling of Information Systems"*, published by Springer in 2007, which is still a frame of reference at many Universities in the world.

After that, Antoni's research interests moved to analyze the role conceptual modeling should play in software engineering and the challenges that conceptual modelers should face in this context. His outmost contribution from this period is probably his proposal for *conceptual schema-centric development*, emphasizing that the conceptual schema should be the driving force in information systems development.

As a conclusion, we would like to say that it is impossible to review all this huge amount of work just in a few pages, although we expect that we have been able to provide a fair summary of its significance to the conceptual modeling discipline. All of us are indebted to this long journey and we will modestly do our best to further pursue in the direction that Antoni has shown us during all these years.

Acknowledgements As coauthors of Antoni's papers, several people from the Universitat Politècnica de Catalunya have contributed to the field of conceptual modeling, and are still doing it with new, younger, researchers. We are grateful to all of them since they also made this chapter possible. So, we wish to thank David Aguilera, Jordi Conesa, Dolors Costal, Juan Ramón López, Joan Antoni Pastor, Carme Quer, Ruth Raventós, Jaume Sistac†, Albert Tort, Toni Urpí and Antonio Villegas for all the good work they did while being PhD students and collaborators of Antoni.

References

1. Aguilera, D., Gómez, C., Olivé, A.: A method for the definition and treatment of conceptual schema quality issues. In: Atzeni et al. [4], pp. 501–514
2. Aguilera, D., Gómez, C., Olivé, A.: A complete set of guidelines for naming UML conceptual schema elements. Data Knowl. Eng. 88, 60–74 (2013)
3. Aguilera, D., Gómez, C., Olivé, A.: Enforcement of conceptual schema quality issues in current integrated development environments. In: Salinesi, C., Norrie, M.C., Pastor, O. (eds.) Advanced Information Systems Engineering - 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7908, pp. 626–640. Springer (2013)
4. Atzeni, P., Cheung, D.W., Ram, S. (eds.): Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7532. Springer (2012)
5. Cabot, J., Olivé, A., Teniente, E.: Entity types derived by symbol-generating rules. In: Song, I., Liddle, S.W., Ling, T.W., Scheuermann, P. (eds.) Conceptual Modeling - ER 2003, 22nd International Conference on Conceptual Modeling, Chicago, IL, USA, October 13-16, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2813, pp. 376–389. Springer (2003)
6. Cabot, J., Olivé, A., Teniente, E.: Representing temporal information in UML. In: Stevens, P., Whittle, J., Booch, G. (eds.) UML 2003 - The Unified Modeling Language, Modeling Languages and Applications, 6th International Conference, San Francisco, CA, USA, October 20-24, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2863, pp. 44–59. Springer (2003)
7. Conesa, J., Olivé, A.: A method for pruning ontologies in the development of conceptual schemas of information systems pp. 64–90 (2006)
8. Costal, D., Olivé, A.: A method for reasoning about deductive conceptual models of information systems. In: CAiSE. pp. 612–631 (1992), `https://doi.org/10.1007/BFb0035156`
9. Costal, D., Olivé, A., Teniente, E.: Relationship type refinement in conceptual models with multiple classification. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) Conceptual Modeling - ER 2001, 20th International Conference on Conceptual Modeling, Yokohama, Japan, November 27-30, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2224, pp. 397–411. Springer (2001)
10. Gómez, C., Olivé, A.: Evolving partitions in conceptual schemas in the UML. In: Pidduck et al. [25], pp. 467–483

11. López, J., Olivé, A.: A framework for the evolution of temporal conceptual schemas of information systems. In: Wangler, B., Bergman, L. (eds.) Advanced Information Systems Engineering, 12th International Conference CAiSE 2000, Stockholm, Sweden, June 5-9, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1789, pp. 369–386. Springer (2000)
12. Olivé, A.: A comparison of the operational and deductive approaches to conceptual information systems modelling. In: IFIP Congress. pp. 91–96 (1986)
13. Olivé, A.: A formal approach to timing analysis and design of information systems. Inf. Syst. 12(1), 1–10 (1987), https://doi.org/10.1016/0306-4379(87)90013-5
14. Olivé, A.: Relationship reification: A temporal view. In: Jarke, M., Oberweis, A. (eds.) Advanced Information Systems Engineering, 11th International Conference CAiSE'99, Heidelberg, Germany, June 14-18, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1626, pp. 396–410. Springer (1999)
15. Olivé, A.: Representation of generic relationship types in conceptual modeling. In: Pidduck et al. [25], pp. 675–691
16. Olivé, A.: Derivation rules in object-oriented conceptual modeling languages. In: Eder, J., Missikoff, M. (eds.) Advanced Information Systems Engineering, 15th International Conference, CAiSE 2003, Klagenfurt, Austria, June 16-18, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2681, pp. 404–420. Springer (2003)
17. Olivé, A.: Conceptual schema-centric development: A grand challenge for information systems research. In: Pastor, O., e Cunha, J.F. (eds.) Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3520, pp. 1–15. Springer (2005)
18. Olivé, A.: A method for the definition of integrity constraints in object-oriented conceptual modeling languages. Data Knowl. Eng. 59(3), 559–575 (2006)
19. Olivé, A.: Conceptual modeling of information systems. Springer (2007)
20. Olivé, A., Costal, D., Sancho, M.: Entity evolution in ISA hierarchies. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) Conceptual Modeling - ER '99, 18th International Conference on Conceptual Modeling, Paris, France, November, 15-18, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1728, pp. 62–80. Springer (1999)
21. Olivé, A., Raventós, R.: Modeling events as entities in object-oriented conceptual modeling languages. Data Knowl. Eng. 58(3), 243–262 (2006)
22. Olivé, A., Sancho, M.: Validating conceptual specifications through model execution. Inf. Syst. 21(2), 167–186 (1996), https://doi.org/10.1016/0306-4379(96)00010-5
23. Olivé, A., Teniente, E.: Derived types and taxonomic constraints in conceptual modeling. Inf. Syst. 27(6), 391–409 (2002)
24. Pastor-Collado, J.A., Olivé, A.: Supporting transaction design in conceptual modelling of information systems. In: International Conference on Advanced Information Systems Engineering. pp. 40–53. Springer (1995)
25. Pidduck, A.B., Mylopoulos, J., Woo, C.C., Özsu, M.T. (eds.): Advanced Information Systems Engineering, 14th International Conference, CAiSE 2002, Toronto, Canada, May 27-31, 2002, Proceedings, Lecture Notes in Computer Science, vol. 2348. Springer (2002)
26. Quer, C., Olivé, A.: Deteriming object interaction in object-oriented deductive conceptual models. Inf. Syst. 19(3), 211–227 (1994), https://doi.org/10.1016/0306-4379(94)90042-6
27. Raventós, R., Olivé, A.: An object-oriented operation-based approach to translation between MOF metaschemas. Data Knowl. Eng. 67(3), 444–462 (2008)
28. Sancho, M.R., Olivé, A.: Deriving transaction specifications from deductive conceptual models of information systems. In: International Conference on Advanced Information Systems Engineering. pp. 311–324. Springer (1994)
29. Sistac, J.: The DADES/GP approach to automatic generation of information system prototypes from a deductive conceptual model. Inf. Syst. 17(3), 195–208 (1992), https://doi.org/10.1016/0306-4379(92)90013-D
30. Teniente, E., Olivé, A.: The events method for view updating in deductive databases. In: Advances in Database Technology - EDBT'92, 3rd International Conference on Extending

Database Technology, Vienna, Austria, March 23-27, 1992, Proceedings. pp. 245–260 (1992), https://doi.org/10.1007/BFb0032435
31. Teniente, E., Olivé, A.: Updating knowledge bases while maintaining their consistency. VLDB J. 4(2), 193–241 (1995), http://www.vldb.org/journal/VLDBJ4/P193.pdf
32. Tort, A., Olivé, A.: An approach to testing conceptual schemas. Data Knowl. Eng. 69(6), 598–618 (2010)
33. Tort, A., Olivé, A.: An approach to website schema.org design. Data Knowl. Eng. 99, 3–16 (2015)
34. Tort, A., Olivé, A., Sancho, M.: An approach to test-driven development of conceptual schemas. Data Knowl. Eng. 70(12), 1088–1111 (2011)
35. Urpí, T., Olivé, A.: Evants and events rules in active databases. IEEE Data Eng. Bull. 15(1-4), 56–59 (1992), http://sites.computer.org/debull/92DEC-CD.pdf
36. Urpí, T., Olivé, A.: A method for change computation in deductive databases. In: 18th International Conference on Very Large Data Bases, August 23-27, 1992, Vancouver, Canada, Proceedings. pp. 225–237 (1992), http://www.vldb.org/conf/1992/P225.PDF
37. Villegas, A., Olivé, A.: A method for filtering large conceptual schemas. Conceptual Modeling–ER 2010 pp. 247–260 (2010)
38. Villegas, A., Olivé, A., Sancho, M.: Understanding constraint expressions in large conceptual schemas by automatic filtering. In: Atzeni et al. [4], pp. 50–63