The DBLP Case Study

Elena Planas, Antoni Olivé

ÍNDEX

1 INTRODUCTION

This document contains parts of the conceptual schema of the DBLP system, written in UML. **DBLP**, a computer science bibliography website (http://www.informatik.uni-trier.de/~ley/db/), was originally a database and logic programming bibliography site, homed at Universität Trier, in Germany, and has existed at least since the 1980s. As of January 2006, DBLP listed more than 710,000 articles on the computer science field, mirrored at five sites across the Internet. Some of the journals which are tracked on this site include VLDB, a journal for very large databases, and the ACM Transactions.

Nowadays, it's suggested that DBLP (once called **D**ata**B**ase systems and **L**ogic **P**rogramming) now stands for **Digital Bibliography & Library Project**.

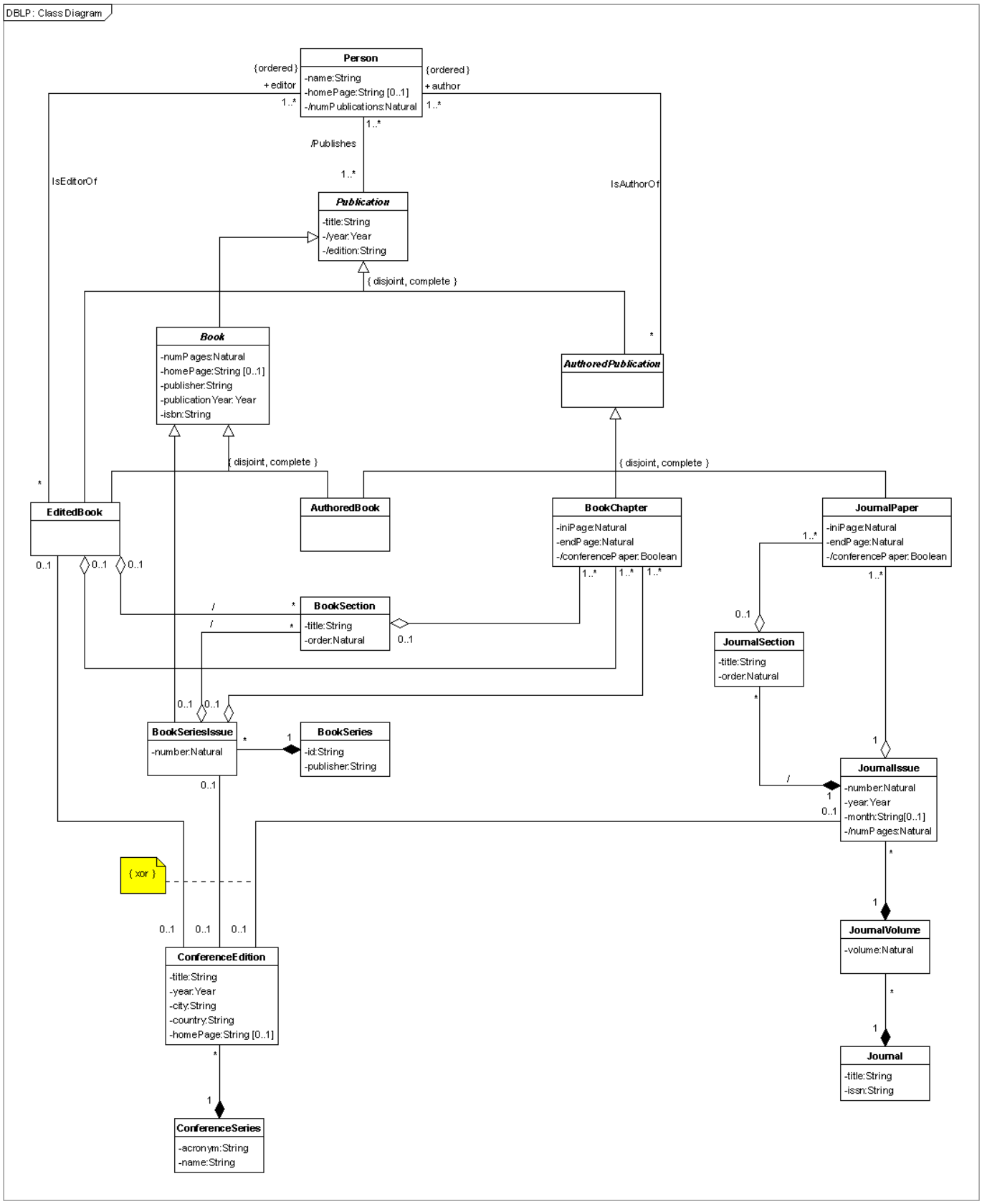
2 STRUCTURAL SCHEMA

The structural schema presented here deals with persons (authors and editors) and their publications, which may be edited books or authored publications such as authored books, book chapters and journal papers. Book chapters and journal papers may or may not be conference papers.

In order to simplify the schema, we have not considered:

- The links to the publications.
- Journal issues with more than one number.
- Journal issues that publish the proceedings of more than one conference edition.

2.1 CLASS DIAGRAM



2.2 PATTERN SENTENCES OF ASSOCIATIONS

Linguistically, a relationship is a fact that holds in the domain and that can be expressed by means of a grammatical sentence.

The pattern sentence of a relationship type is a declarative sentence with a placeholder for each participant. The sentence that linguistically expresses a relationship is obtained by filling in the placeholders with the names of the participants.

The pattern sentences that help us in understanding the meaning of relationship types of DBLP schema are:

1. Association *EditedBook – ConferenceEdition*:
   The edited book <*EditedBook*> publishes the proceedings of the conference edition <*ConferenceEdition*>

2. Association *BookSeriesIssue - ConferenceEdition*:
   The book series issue <*BookSeriesIssue*> publishes the proceedings of the conference edition <*ConferenceEdition*>

3. Association *JournalIssue - ConferenceEdition*:
   The journal issue <*JournalIssue*> publishes the proceedings of the conference edition <*ConferenceEdition*>

2.3 IDENTIFICATION CONSTRAINTS

1. *Person*: *name*
```
context Person inv nameIsKey:
   Person.allInstances() -> isUnique(name)
```

2. *Book*: *isbn*
```
context Book inv isbnIsKey:
   Book.allInstances() -> isUnique(isbn)
```

3. *BookSeries*: *id*
```
context BookSeries inv idIsKey:
   BookSeries.allInstances() -> isUnique(id)
```

4. *BookSeriesIssue*: *book series+ number*
```
context BookSeries inv BookSeriesAndNumberIdentifyBookSeriesIssue:
   self.bookSeriesIssue -> isUnique(number)
```

5. *Journal*: *issn*
```
context Journal inv issnIsKey:
   Journal.allInstances() -> isUnique(issn)
```

6. *Journal*: *title*
```
context Journal inv titleIsKey:
   Journal.allInstances() -> isUnique(title)
```

7. *JournalVolume*: *journal + volume*
```
context Journal inv journalAndVolumeIdentifyJournalVolume:
   self.journalVolume -> isUnique(volume)
```

8. *JournalIssue*: *journal volume + number*
```
context JournalVolume inv journalVolumeAndNumberIdentifyJournalIssue:
```

```
self.journalIssue -> isUnique(number)
```

9. *JournalSection*: *journal issue* + *title*

```
context JournalIssue inv journalIssueAndTitleIdentifyJournalSection:
   self.journalSection -> isUnique(title)
```

10. *ConferenceSeries*: *name*

```
context ConferenceSeries inv nameIsKey:
   ConferenceSeries.allInstances() -> isUnique(name)
```

11. *ConferenceEdition*: *title*

```
context ConferenceEdition inv titleIsKey:
   ConferenceEdition.allInstances() -> isUnique(title)
```

2.4 OTHER INTEGRITY CONSTRAINTS

1. The last page of a book chapter (*BookChapter*) must be equal or greater than the initial page.

```
context BookChapter inv correctPagination:
   self.iniPage ≤ self.endPage
```

2. The last page of a journal paper (*JournalPaper*) must be equal or greater than the initial page.

```
context JournalPaper inv correctPagination:
   self.iniPage ≤ self.endPage
```

3. The pages of the papers (*JournalPaper*) published in a journal issue (*JournalIssue*) do not overlap among them.

```
context JournalIssue inv correctPagination:
   self.journalPaper -> forAll(p1,p2 | p1<>p2 implies
         p1.iniPage > p2.endPage or p2.iniPage > p1.endPage)
```

4. The pages of the chapters (*BookChapter*) published in an edited book (*EditedBook*) do not overlap.

```
context EditedBook inv correctPagination:
   self.bookChapter -> forAll(c1,c2 | c1<>c2 implies
         c1.iniPage > c2.endPage or c2.iniPage > c1.endPage)
```

5. The pages of the chapters (*BookChapter*) that belong to an edition of book series (*BookSeriesIssue*) do not overlap.

```
context BookSeriesIssue inv correctPagination:
   self.bookChapter -> forAll(c1,c2 | c1<>c2 implies
         c1.iniPage > c2.endPage or c2.iniPage > c1.endPage)
```

6. The volumes of a *Journal* are consecutive starting from 1.

```
context Journal inv consecutiveVolumes:
   self.journalVolume -> sortedBy(volume).volume =
                         sequence{1..self.journalVolume -> size()}
```

7. The year of publication of a book (*EditedBook*) that publishes the proceedings of a conference must be equal or greater than the year of edition of the conference (*ConferenceEdition*) that it publishes.

```
context EditedBook inv compatibleYear:
   (self.conferenceEdition -> notEmpty()) implies
    self.publicationYear ≥ self.conferenceEdition.year
```

8. The year of publication of a book series issue (*BookSeriesIssue*) must be equal or greater

than the year of edition of the conference (*ConferenceEdition*) that it publishes.

```
context BookSeriesIssue inv compatibleYear:
   (self.conferenceEdition -> notEmpty()) implies
    self.publicationYear ≥ self.conferenceEdition.year
```

9. The year of publication of a journal issue (*JournalIssue*) that publishes the proceedings of a conference must be equal or greater than the year of edition of the conference (*ConferenceEdition*) that it publishes.

```
context JournalIssue inv compatibleYear:
   (self.conferenceEdition -> notEmpty()) implies
    self.year ≥ self.conferenceEdition.year
```

10. An edited book (EditedBook) cannot have more than one section (BookSection) with the same title.

```
context EditedBook inv editedBookWithoutRepetitions:
   self.bookSection -> isUnique(title)
```

11. A book series issue (*BookSeriesIssue*) cannot have more than one section (*BookSection*) with the same title.

```
context BookSeriesIssue inv bookSeriesIssueWithoutRepetitions:
   self.bookSection -> isUnique(title)
```

12. A journal section (*JournalSection*) cannot have more than one paper (*JournalPaper*) with the same title.

```
context JournalSection inv journalSectionWithoutRepetitions:
   self.journalPaper -> isUnique(title)
```

13. A book section (*BookSection*) cannot have more than one chapter with the same title.

```
context BookSection inv bookSectionWithoutRepetitions:
   self.bookChapter -> isUnique(title)
```

14. An edition of a conference (*ConferenceEdition*) must be published in an edited book (*EditedBook*), in a book series issue (*BookSeriesIssue*) or in a journal issue (*JournalIssue*).

```
context ConferenceEdition inv conferenceIsPublished:
   self.editedBook -> notEmpty() or
   self.bookSeriesIssue -> notEmpty() or
   self.journalIssue -> notEmpty()
```

15. The publisher of a book series issue (*BookSeriesIssue*) is the same publisher of its book series (*BookSeries*).

```
context Book inv theSamePublisher:
   if self.oclIsTypeOf(BookSeriesIssue)
   then self.publisher = self.oclAsType(BookSeriesIssue).bookSeries.publisher
```

2.5 DERIVATION RULES

Most derivation rules of attributes and associations are defined as proposed in the OCL specification. A few of them however have been defined using "defining operations" as explained in:

Antoni Olivé: Derivation Rules in Object-Oriented Conceptual Modeling Languages. CAiSE 2003: 404-420

We use "defining operations" only when the standard specification is not possible. This happens when we want to redefine a derivation rule in subclasses.

2.5.1 Derived attributes

1. Attribute *numPublications* of *Person*: The number of publications of a person is the cardinality of the set of his or her publications.

```
context Person::numPublications:Natural
   derive: self.publication -> size()
```

2. Attribute *numPages* of *JournalIssue*: The number of pages of a journal issue (*JournalIssue*) is the number of the last page of the last paper it contains:

```
context JournalIssue::numPages:Natural
   derive: self.journalPaper -> sortedBy(endPage) -> last().endPage
```

3. Attribute *year* of *Publication*: The year of a publication is the publication year of the book (*Book*) or journal issue (*JournalIssue*) that publishes it.

```
context Publication::year():Year
   body: (abstract)

context EditedBook::year():Year
   body: self.publicationYear

context AuthoredBook::year():Year
   body: self.publicationYear

context BookChapter::year():Year
   body: if self.editedBook -> notEmpty()
         then self.editedBook.publicationYear
         else self.bookSeriesIssue.publicationYear
         endif

context JournalPaper::year():Year
   body: self.journalIssue.year
```

4. Attribute *edition* of *Publication*: The edition of a publication consists of the concatenation of several pieces of information related to the publication.

```
context Publication::edition():String
   body: (abstract)
```

The edition of an edited book (EditedBook) consist of: *publisher* + *publicationYear*
(We assume that there exists the operation *toString()*, that converts the simple types Year and Natural into a String type.)

```
context EditedBook::edition():String
   body: self.publisher.concat(self.publicationYear.toString())
```

The edition of an authored book (*AuthoredBook*) consist of: *publisher* + *publicationYear*

```
context AuthoredBook::edition():String
   body: self.publisher.concat(self.publicationYear.toString())
```

The edition of a book chapter (BookChapter) consist of:
- *acronym* + *year* + *iniPage* + *endPage*, if it refers a conference chapter
- *title* + *year* + *iniPage* + *endPage*, if it refers a conventional chapter

```
context BookChapter::edition():String
   body: if (self.conferencePaper)
         then -- BookChapter of ConferencePaper
             if self.editedBook -> notEmpty()
             then -- BookChapter of EditedBook
                 self.editedBook.conferenceEdition.conferenceSeries.acronym
                 .concat(self.editedBook.conferenceEdition.year.toString()
```

```
                .concat(self.iniPage.toString()
                .concat(self.endPage.toString())))
            else -- BookChapter of BookSeriesIssue
                self.bookSeriesIssue.conferenceEdition.conferenceSeries
                .acronym
                .concat(self.bookSeriesIssue.conferenceEdition.year
                 .toString()
                .concat(self.iniPage.toString()
                .concat(self.endPage.toString())))
            endif
        else -- conventional BookChapter
            if self.editedBook -> notEmpty()
            then -- BookChapter of EditedBook
                self.editedBook.title
                .concat(self.editedBook.publicationYear.toString()
                .concat(self.iniPage.toString()
                .concat(self.endPage.toString())))
            else - BookChapter of BookSeriesIssue
                self.bookSeriesIssue.title
                .concat(self.bookSeriesIssue.publicationYear.toString()
                .concat(self.iniPage.toString()
                .concat(self.endPage.toString())))
            endif
        endif
```

The edition of a journal paper (JournalPaper) consist of:

- *acronym* + *year* + *iniPage* + *endPage*, if it refers a conference paper
- *title* + *volume* + *issue* + *iniPage* + *endPage* + *year*, if it refers a conventional paper

```
context JournalPaper::edition():String
   body: if (self.conferencePaper)
         then -- JournalPaper of ConferencePaper
              self.journalIssue.conferenceEdition.conferenceSeries.acronym
              .concat(self.journalIssue.conferenceEdition.year.toString()
              .concat(self.iniPage.toString()
              .concat(self.endPage.toString())))
         else -- conventional JournalPaper
             self.journalIssue.journalVolume.journal.title
             .concat(self.journalIssue.journalVolume.volume.toString()
             .concat(self.journalIssue.number.toString()
             .concat(self.iniPage.toString()
             .concat(self.endPage.toString()
             .concat(self.journalIssue.year.toString())))))
         endif
```

5. Attribute *conferencePaper* of *JournalPaper:* We assume that a journal paper is a conference paper if it is included in a journal issue that publishes the proceedings of a conference edition.

```
context JournalPaper::conferencePaper:Boolean
   derive: self.journalIssue.conferenceEdition -> notEmpty()
```

6. Attribute *conferencePaper* of *BookChapter:* We assume that a book chapter is a conference paper if it is included in a book that publishes the proceedings of a conference edition.

```
context BookChapter::conferencePaper:Boolean
   derive: self.editedBook.conferenceEdition -> notEmpty() or
           self.bookSeriesIssue.conferenceEdition -> notEmpty()
```

2.5.2 Derived associations

1. Association *Publishes* between *Person* and *Publication*: The set of publications of a person are that person's edited books and authored publications.

```
context Person::publication:Publication
   derive: self.editedBook -> union(self.authoredPublication)
```

2. Association between *JournalIssue* and *JournalSection*: The sections of a journal issue (*JournalIssue*) are all those that contain their papers (*JournalPapers*).

```
context JournalIssue::journalSection:JournalSection
   derive: self.journalPaper.journalSection
```

3. Association between *EditedBook* and *BookSection*: The sections of an edited book (EditedBook) are all those that contain their chapters (*BookChapter*).

```
context EditedBook::bookSection:BookSection
   derive: self.bookChapter.bookSection
```

4. Association between *BookSeriesIssue* and *BookSection*: The sections of a book series issue (*BookSeriesIssue*) are all those that contain their chapters (*BookChapter*).

```
context BookSeriesIssue::bookSection:BookSection
   derive: self.bookChapter.bookSection
```

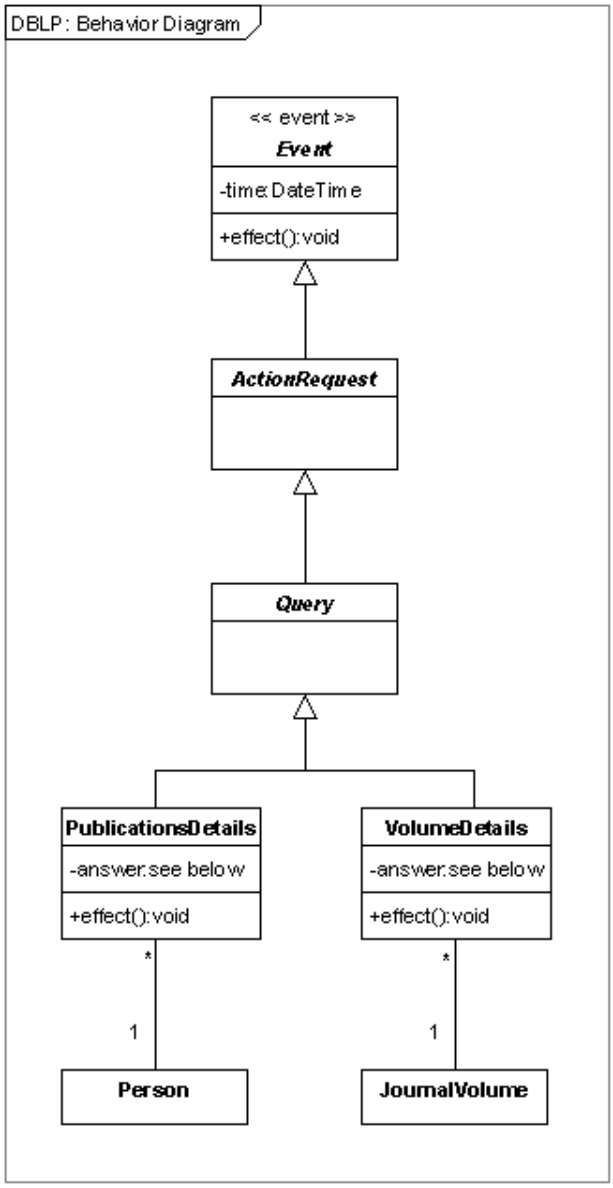
3 BEHAVIORAL SCHEMA

In this document, we deal only with two important queries of the behavioral schema of DLBLP:

- **PublicationDetails**: Given a person, the query provides the available information of that person's publications.
- **VolumeDetails**: Given a journal volume, the query provides the information of the volume issues and the papers published in each of them.

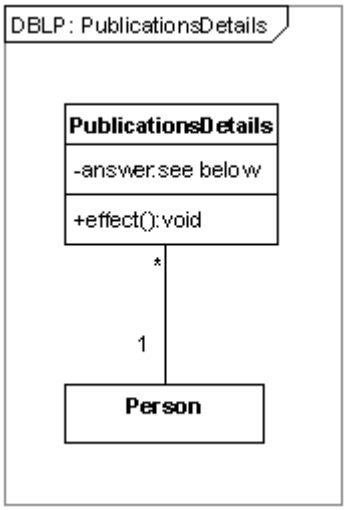
The queries are specified in the style described in:
Antoni Olivé: Definition of Events and Their Effects in Object-Oriented Conceptual Modeling Languages. ER 2004: 136-149

3.1 BEHAVIOR DIAGRAM



3.2 QUERY SPECIFICATION

3.2.1 PublicationsDetails



Class: PublicationsDetails

Attributes:

```
      answer: Set (TupleType
                     (year: Year,
                      yearPublications:
                        Set (TupleType
                                authorsOrEditors: Set(String),
                                title: String,
                                edition: String)))
```

Operations:

```
      effect()
```

```
context PublicationDetails::effect()
post:
  answer =
   self.person.publication.year -> asSet() -> collect(y |
     Tuple
       {year = y,
        yearPublications =
          self.person.publication -> select (p | p.year = y) -> collect (p2 |
          Tuple
            {authorsOrEditors = p2.person.name,
             title = p2.title,
             edition = p2.edition})}) -> sortedBy(year)
```

For example, if we ask the publication details of the person named:

```
Person.name = "Peter P. Chen"
```

we get the following answer:

Peter P. Chen

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL - ACM Guide - CiteSeer - CSB - Google

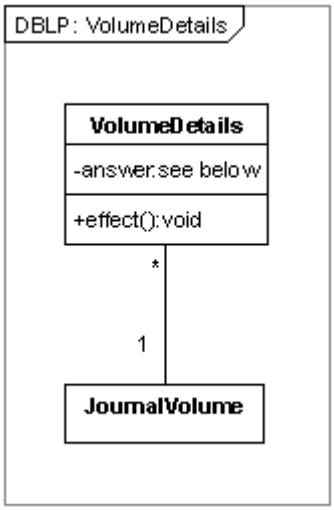
Home Page

		2005
53		Peter P. Chen, Leah Y. Wong: A Proposed Preliminary Framework for Conceptual Modeling of Learning from Surprises. IC-AI 2005: 905-910
52	EE	R. F. Lax, Guoli Ding, Peter P. Chen, Jianhua Chen: Approximating Pseudo-Boolean Functions on Non-Uniform Domains. IJCAI 2005: 1754-1755
51	EE	Guoli Ding, Jianhua Chen, Robert Lax, Peter P. Chen: Efficient Learning of Pseudo-Boolean Functions from Limited Training Data. ISMIS 2005: 323-331
50	EE	Steven S. Seiden, Peter P. Chen, R. F. Lax, Jianhua Chen, Guoli Ding: New bounds for randomized busing. Theor. Comput. Sci. 332(1-3): 63-81 (2005)
		2004
49	EE	Colleen Cunningham, Il-Yeol Song, Peter P. Chen: Data warehouse design to support customer relationship management analyses. DOLAP 2004: 14-22
48	EE	John Horner, Il-Yeol Song, Peter P. Chen: An analysis of additivity in OLAP systems. DOLAP 2004: 83-91
47	EE	Peter P. Chen: Editorial introduction by the Editor-in-Chief. Data Knowl. Eng. 50(3): 241-246 (2004)
46	EE	Guoli Ding, Peter P. Chen: Unavoidable doubly connected large graphs. Discrete Mathematics 280(1-3): 1-12 (2004)
45	EE	Min Song, Il-Yeol Song, Peter P. Chen: Design and Development of a Cross Search Engine for Multiple Heterogeneous Databases Using UML and Design Patterns. Information Systems Frontiers 6(1): 77-90 (2004)
44	EE	Peter P. Chen, Guoli Ding: The best expert versus the smartest algorithm. Theor. Comput. Sci. 324(2-3): 361-380 (2004)
		2003
43	EE	Peter P. Chen: XML and the Semantic Web: What is the future? HICSS 2003: 122
42	EE	Guoli Ding, Peter P. Chen: Generating r-regular graphs. Discrete Applied Mathematics 129(2-3): 329-343 (2003)
		2002
41	EE	Peter P. Chen, Reind P. van de Riet: Editorial introduction. Data Knowl. Eng. 41(2-3): 125-132 (2002)
40	EE	Peter P. Chen: From Goto-less to Structured Programming: The Legacy of Edsger W. Dijkstra. IEEE Software 19(5): 21 (2002)
		1999
39		Peter P. Chen, Jacky Akoka, Hannu Kangassalo, Bernhard Thalheim: Conceptual Modeling, Current Issues and Future Directions, Selected Papers from the Symposium on Conceptual Modeling, Los Angeles, California, USA, held before ER'97 Springer 1999
38		Peter P. Chen, David W. Embley, Jacques Kouloumdjian, Stephen W. Liddle, John F. Roddick: Advances in Conceptual Modeling: ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling, Paris, France, November 15-18, 1999, Proceedings Springer 1999
37	EE	Peter P. Chen: ER Model, XML and the Web. ER 1999: 538

1998		
36	EE	Peter P. Chen, Reind P. van de Riet: Introduction to the Special Issue Celebrating the 25th Volume of Data & Knowledge Engineering: DKE. Data Knowl. Eng. 25(1-2): 1-9 (1998)
1997		
35	EE	Peter P. Chen, Bernhard Thalheim, Leah Y. Wong: Future Directions of Conceptual Modeling. Conceptual Modeling 1997: 287-301
34	EE	Peter P. Chen: From Ancient Egyptian Language to Future Conceptual Modeling. Conceptual Modeling 1997: 56-64
33	EE	Peter P. Chen: Current Issues of Conceptual Modeling: A Summary of Selective Active Research Topics. Conceptual Modeling 1997: ix-xxiv
32	EE	Peter P. Chen: English, Chinese and ER Diagrams. Data Knowl. Eng. 23(1): 5-16 (1997)
1996		
31	EE	Anyuan Yang, Peter P. Chen: Efficient Data Retrieval and Manipulation Using Boolean Entity Lattice. Data Knowl. Eng. 20(2): 211-226 (1996)
1992		
30	EE	Peter P. Chen: ER vs. OO. ER 1992: 1-2
1989		
29	EE	Asuman Dogac, Esen A. Ozkarahan, Peter P. Chen: An Integrity System for a Relational Database Architecture. ER 1989: 287-301
1987		
28	EE	Sreerama K. Karukonda, Edward T. Lee, Peter P. Chen: Design of a pictorial knowledgebase. ACM Conference on Computer Science 1987: 114-119
27	EE	Peter P. Chen: Products from Chen & Associates. ER 1987: 15-16
1986		
26	EE	Peter P. Chen, Ming-rui Li: The Lattice Structure of Entity Set. ER 1986: 217-229
25	EE	Arie Zvieli, Peter P. Chen: Entity-Relationship Modeling and Fuzzy Databases. ICDE 1986: 320-327
24		Peter P. Chen: The Time Dimension in the Entity-Relationship Model (Invited Paper). IFIP Congress 1986: 387-390
1985		
23		Peter P. Chen: Entity-Relationship Approach: The Use of ER Concept in Knowledge Representation, Proceedings of the Fourth International Conference on Entity-Relationship Approach, Chicago, Illinois, USA, 29-30 October 1985 IEEE Computer Society and North-Holland 1985
22	EE	John F. Sowa, Peter P. Chen, Peter Freeman, Sharon C. Salveter, Roger C. Schank: Mapping Specifications to Formalisms - Panel Session. ER 1985: 100-101
21	EE	Asuman Dogac, Peter P. Chen, N. Erol: The Design and Implementation of an Integrity Subsystem for the Relational DBMS RAP. ER 1985: 295-302
20		Peter P. Chen: Database Design Based on Entity and Realtionship. Principles of Database Design (I) 1985: 174-210
1984		
19	EE	Peter P. Chen: An Algebra for a Directional Binary Entity-Relationship Model. ICDE 1984: 37-40
1983		
18		Peter P. Chen: Entity-Relationship Approach to Information Modeling and Analysis, Proceedings of the Second International Conference on the Entity-Relationship Approach (ER'81), Washington, DC, USA, October 12-14, 1981 North-Holland 1983
17	EE	Peter P. Chen: ER - A Historical Perspective and Future Directions. ER 1983: 71-77
16	EE	Peter P. Chen: English Sentence Structure and Entity-Relationship Diagrams. Inf. Sci. 29(2-3): 127-149 (1983)
1981		

15	EE	Paolo Atzeni, Peter P. Chen: Completeness of Query Languages for the Entity-Relationship Model. ER 1981: 109-122
14	EE	Ilchoo Chung, Fumio Nakamura, Peter P. Chen: A Decomposition of Relations Using the Entity-Relationship Approach. ER 1981: 149-171
13	EE	Peter P. Chen: A Preliminary Framework for Entity-Relationship Models. ER 1981: 19-28
12	EE	Asuman Dogac, Peter P. Chen: Entity-Relationship Model in the ANSI/SPARC Framework. ER 1981: 357-374
		1980
11		Peter P. Chen, R. Clay Sprowls: Proceedings of the 1980 ACM SIGMOD International Conference on Management of Data, Santa Monica, California, May 14-16, 1980. ACM Press 1980
10		Peter P. Chen: Entity-Relationship Approach to Systems Analysis and Design. Proc. 1st International Conference on the Entity-Relationship Approach North-Holland 1980
9		Peter P. Chen, Jacky Akoka: Optimal Design of Distributed Information Systems. IEEE Trans. Computers 29(12): 1068-1080 (1980)
		1979
8	EE	Peter P. Chen: Entity-Relationship Diagrams and English Sentence Structure. ER 1979: 13-14
7	EE	Peter P. Chen: Recent Literature on the Entity-Relationship Approach. ER 1979: 3-12
		1978
6		Peter P. Chen: Applications of the Entity-Relationship Model. Data Base Design Techniques I 1978: 87-113
		1977
5		Peter P. Chen: The Entity-Relationship Model - A basis for the Enterprise View of Data. AFIPS National Computer Conference 1977: 77-84
4	EE	Peter P. Chen, S. Bing Yao: Design and Performance Tools for Data Base Systems. VLDB 1977: 3-15
		1976
3	EE	Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. ACM Trans. Database Syst. 1(1): 9-36 (1976)
		1975
2	EE	Peter P. Chen: The Enity-Relationship Model: Toward a Unified View of Data. VLDB 1975: 173
		1974
1		Jeffrey P. Buzen, Peter P. Chen: Optimal Load Balancing in Memory Hierarchies. IFIP Congress 1974: 271-275

3.2.2 VolumeDetails



Class: VolumeDetails

Attributes:

```
      answer: TupleType
       (journal: String,
        volume: Natural,
        issues: Set (TupleType
                      (number: Natural,
                       month: String,
                       year: Year,
                       sections: Set (TupleType
                                       (titleSection: String,
                                        papers: Set (TupleType
                                                      (authors: Set(String),
                                                      title: String,
                                                      iniPage: Natural,
                                                      endPage: Natural)))))))
```

Operations:

```
      effect()
```

```
context VolumeDetails::effect()
post:
  answer =
   Tuple
     {journal = self.journalVolume.journal.title,
      volume = self.journalVolume.volume,
      issues = self.journalVolume.journalIssue -> sortedBy(number)-> collect(i|
        Tuple
          {number = i.number,
           month = i.month,
           year = i.year,
           sections = i.journalSection -> sortedBy(order)-> collect(s |
             Tuple
               {titleSection = s.title,
                papers = s.journalPaper -> sortedBy(iniPage)-> collect(p |
                  Tuple
                    {authors = p.person.name,
                     title = p.title,
                     iniPage = p.iniPage,
                     endPage = p.endPage })})})}
```

For example, if we ask the volume details of the volume:

```
JournalVolume.Journal.title = "ACM Transactions on Database Systems(TODS)"
JournalVolume.volume = 1
```

we get the following answer:

ACM Transactions on Database Systems (TODS) , Volume 1

Volume 1, Number 1, March 1976

- David K. Hsiao:
  ACM Transactions on Database Systems - Aim and Scope. 1-2
  Electronic Edition (ACM DL) BibTeX
- R. Stockton Gaines, David K. Hsiao:
  Papers from the International Conference on Very Large Data Bases, September 22-24, 1975, Framingham, Massachusetts. 3-8
  Electronic Edition BibTeX
- Peter P. Chen:
  The Entity-Relationship Model - Toward a Unified View of Data. 9-36
  Electronic Edition (ACM DL) BibTeX
- Rudolf Bayer, J. K. Metzger:
  On the Encipherment of Search Trees and Random Access Files. 37-52
  Electronic Edition (ACM DL) BibTeX
- Chyuan Shiun Lin, Diane C. P. Smith, John Miles Smith:
  The Design of a Rotating Associative Array Memory for a Relational Database Management Application. 53-65
  Electronic Edition (ACM DL) BibTeX
- Samy A. Mahmoud, J. Spruce Riordon:
  Optimal Allocation of Resources in Distributed Information Networks. 66-78
  Electronic Edition (ACM DL) BibTeX
- David W. Stemple:
  A Data Base Management Facility for Automatic Generation of Data Base Managers. 79-94
  Electronic Edition (ACM DL) BibTeX

Volume 1, Number 2, June 1976

- Morton M. Astrahan, Mike W. Blasgen, Donald D. Chamberlin, Kapali P. Eswaran, Jim Gray, Patricia P. Griffiths, W. Frank King III, Raymond A. Lorie, Paul R. McJones, James W. Mehl, Gianfranco R. Putzolu, Irving L. Traiger, Bradford W. Wade, Vera Watson:
  System R: Relational Approach to Database Management. 97-137
  Electronic Edition (ACM DL) BibTeX
- Shamkant B. Navathe, James P. Fry:
  Restructuring for Large Data Bases: Three Levels of Abstraction. 138-158
  Electronic Edition (ACM DL) BibTeX
- S. Bing Yao, K. Sundar Das, Toby J. Teorey:
  A Dynamic Database Reorganization Algorithm. 159-174
  Electronic Edition (ACM DL) BibTeX
- Walter A. Burkhard:
  Hashing and Trie Algorithms for Partial Match Retrieval. 175-187
  Electronic Edition (ACM DL) BibTeX

Volume 1, Number 3, September 1976

- Michael Stonebraker, Eugene Wong, Peter Kreps, Gerald Held:
  The Design and Implementation of INGRES. 189-222
  Electronic Edition (ACM DL) BibTeX

- Eugene Wong, Karel Youssefi:
  Decomposition - A Strategy for Query Processing. 223-241
  Electronic Edition (ACM DL) BibTeX
- Patricia P. Griffiths, Bradford W. Wade:
  An Authorization Mechanism for a Relational Database System. 242-255
  Electronic Edition (ACM DL) BibTeX
- Dennis G. Severance, Guy M. Lohman:
  Differential Files: Their Application to the Maintenance of Large Databases. 256-267
  Electronic Edition (ACM DL) BibTeX
- Ben Shneiderman, Victor Goodman:
  Batched Searching of Sequential and Tree Structured Files. 268-275
  Electronic Edition (ACM DL) BibTeX

Volume 1, Number 4, December 1976

- Philip A. Bernstein:
  Synthesizing Third Normal Form Relations from Functional Dependencies. 277-298
  Electronic Edition (ACM DL) BibTeX
- Jane W.-S. Liu:
  Algorithms for Parsing Search Queries in Systems with Inverted File Organization. 299-316
  Electronic Edition (ACM DL) BibTeX
- Stephen W. Sherman, Richard S. Brice:
  Performance of a Database Manager in a Virtual Memory System. 317-343
  Electronic Edition (ACM DL) BibTeX
- John J. Donovan:
  Database System Approach to Management Decision Support. 344-369
  Electronic Edition (ACM DL) BibTeX
- William C. McGee:
  On User Criteria for Data Model Evaluation. 370-387
  Electronic Edition (ACM DL) BibTeX