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 DataBase systems and Logic Programming) 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:

2. Association BookSeriesIssue - 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:
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
   ```

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

   ```
   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 \( (\text{ConferenceEdition}) \) that it publishes.

\[ \text{context BookSeriesIssue} \; \text{inv} \; \text{compatibleYear}: \\
   \left( \text{self.conferenceEdition} \rightarrow \text{notEmpty()} \right) \implies \\
   \text{self.publicationYear} \geq \text{self.conferenceEdition.year} \]

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

\[ \text{context JournalIssue} \; \text{inv} \; \text{compatibleYear}: \\
   \left( \text{self.conferenceEdition} \rightarrow \text{notEmpty()} \right) \implies \\
   \text{self.year} \geq \text{self.conferenceEdition.year} \]

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

\[ \text{context EditedBook} \; \text{inv} \; \text{editedBookWithoutRepetitions}: \\
   \text{self.bookSection} \rightarrow \text{isUnique(title)} \]

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

\[ \text{context BookSeriesIssue} \; \text{inv} \; \text{bookSeriesIssueWithoutRepetitions}: \\
   \text{self.bookSection} \rightarrow \text{isUnique(title)} \]

12. A journal section \( (\text{JournalSection}) \) cannot have more than one paper \( (\text{JournalPaper}) \) with the same title.

\[ \text{context JournalSection} \; \text{inv} \; \text{journalSectionWithoutRepetitions}: \\
   \text{self.journalPaper} \rightarrow \text{isUnique(title)} \]

13. A book section \( (\text{BookSection}) \) cannot have more than one chapter with the same title.

\[ \text{context BookSection} \; \text{inv} \; \text{bookSectionWithoutRepetitions}: \\
   \text{self.bookChapter} \rightarrow \text{isUnique(title)} \]


\[ \text{context ConferenceEdition} \; \text{inv} \; \text{conferenceIsPublished}: \\
   \text{self.editedBook} \rightarrow \text{notEmpty()} \text{ or } \\
   \text{self.bookSeriesIssue} \rightarrow \text{notEmpty()} \text{ or } \\
   \text{self.journalIssue} \rightarrow \text{notEmpty()} \]

15. The publisher of a book series issue \( (\text{BookSeriesIssue}) \) is the same publisher of its book series \( (\text{BookSeries}) \).

\[ \text{context Book} \; \text{inv} \; \text{theSamePublisher}: \\
   \text{if self.oclIsTypeOf(BookSeriesIssue)} \text{ then } \text{self.publisher} = \text{self.oclAsType(BookSeriesIssue).bookSeries.publisher} \]

\[2.5 \; \text{DERIVATION RULES}\]

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:

\[ \text{Antoni Olivé: Derivation Rules in Object-Oriented Conceptual Modeling Languages. CAiSE 2003: 404-420} \]

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

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)
   ```
   ```
   context EditedBook::edition():String
   body: self.publisher.concat(self.publicationYear.toString())
   ```
   ```
   context AuthoredBook::edition():String
   body: self.publisher.concat(self.publicationYear.toString())
   ```
   ```
   context BookChapter::edition():String
   body: if (self.conferencePaper) then -- BookChapter of ConferencePaper
   else -- BookChapter of EditedBook
   self.editedBook.conferenceEdition.conferenceSeries.acronym
   .concat(self.editedBook.conferenceEdition.year.toString())
   ```
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
  .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
end

derive: self.journalIssue.conferenceEdition -> notEmpty()

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
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 PublicationsDetails::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

<table>
<thead>
<tr>
<th>Year</th>
<th>Volume</th>
<th>Authors</th>
<th>Title</th>
<th>Journal/Conference</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EE</td>
<td>R. F. Lax, Guoli Ding, Peter P. Chen, Jianhua Chen</td>
<td>Approximating Pseudo-Boolean Functions on Non-Uniform Domains.</td>
<td>IJCAI 2005</td>
<td>1754-1755</td>
</tr>
<tr>
<td></td>
<td>EE</td>
<td>Guoli Ding, Jianhua Chen, Robert Lax, Peter P. Chen</td>
<td>Efficient Learning of Pseudo-Boolean Functions from Limited Training Data.</td>
<td>ISMIS 2005</td>
<td>323-331</td>
</tr>
<tr>
<td>2004</td>
<td></td>
<td>Peter P. Chen, Il-Yeol Song</td>
<td>Data warehouse design to support customer relationship management analyses.</td>
<td>DOLAP 2004</td>
<td>14-22</td>
</tr>
<tr>
<td></td>
<td>EE</td>
<td>John Horner, Il-Yeol Song</td>
<td>An analysis of additivity in OLAP systems.</td>
<td>DOLAP 2004</td>
<td>83-91</td>
</tr>
<tr>
<td></td>
<td>EE</td>
<td>Guoli Ding, Peter P. Chen</td>
<td>Unavoidable doubly connected large graphs.</td>
<td>Discrete Mathematics 280(1-3)</td>
<td>1-12 (2004)</td>
</tr>
<tr>
<td>2003</td>
<td></td>
<td>Peter P. Chen</td>
<td>XML and the Semantic Web: What is the future?</td>
<td>HICSS 2003</td>
<td>122</td>
</tr>
<tr>
<td>1999</td>
<td></td>
<td>Peter P. Chen, Jacky Akoka, Hannu Kangassalo, Bernhard Thalheim</td>
<td>Conceptual Modeling, Current Issues and Future Directions, Selected Papers from the Symposium on Conceptual Modeling, Los Angeles, California, USA, held before ER'97</td>
<td>Springer 1999</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EE</td>
<td>Peter P. Chen</td>
<td>ER Model, XML and the Web.</td>
<td>ER 1999</td>
<td>538</td>
</tr>
<tr>
<td>Year</td>
<td>EE</td>
<td>Authors</td>
<td>Title</td>
<td>Journal/Conference</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>----------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>Peter P. Chen</td>
<td>From Ancient Egyptian Language to Future Conceptual Modeling.</td>
<td>Conceptual Modeling 1997: 56-64</td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>30</td>
<td>Peter P. Chen</td>
<td>ER vs. OO.</td>
<td>ER 1992: 1-2</td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1991</td>
<td>29</td>
<td>Asuman Dogac, Esen A. Ozkarahan, Peter P. Chen</td>
<td>An Integrity System for a Relational Database Architecture.</td>
<td>ER 1989: 287-301</td>
<td></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Peter P. Chen</td>
<td>Products from Chen &amp; Associates.</td>
<td>ER 1987: 15-16</td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>26</td>
<td>Peter P. Chen</td>
<td>The Lattice Structure of Entity Set.</td>
<td>ER 1986: 217-229</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>John F. Sowa, Peter P. Chen, Peter Freeman, Sharon C. Salveter, Roger C. Schank</td>
<td>Mapping Specifications to Formalisms - Panel Session.</td>
<td>ER 1985: 100-101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>Asuman Dogac, Peter P. Chen, N. Erol</td>
<td>The Design and Implementation of an Integrity Subsystem for the Relational DBMS RAP.</td>
<td>ER 1985: 295-302</td>
<td></td>
</tr>
<tr>
<td>1987</td>
<td>19</td>
<td>Peter P. Chen</td>
<td>An Algebra for a Directional Binary Entity-Relationship Model.</td>
<td>ICDE 1984: 37-40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>Peter P. Chen</td>
<td>ER - A Historical Perspective and Future Directions.</td>
<td>ER 1983: 71-77</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Authors</td>
<td>Title and Publication Details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1978</td>
<td>Peter P. Chen</td>
<td>Applications of the Entity-Relationship Model. Data Base Design Techniques I 1978: 87-113</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1977</td>
<td>Peter P. Chen</td>
<td>The Entity-Relationship Model - A basis for the Enterprise View of Data. AFIPS National Computer Conference 1977: 77-84</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 ))))))}
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
  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

  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