

Conceptual Graphs Based Modeling of Semi-Structured Data

Andrea Eva Molnar, Viorica Varga, and [Christian Sacarea](#)

Babeş-Bolyai University, Cluj-Napoca, Romania

ICCS 2018, June 20-22, 2018, Edinburgh



INTRODUCTION

- Visual query systems (VQSs) use visual representation of the structure of databases and queries.
- VQSs are designed to improve the effectiveness of the human-computer communication.
- Conceptual Graphs: express meaning in a logically precise form which is computer tractable and human readable.
- Previous work focused on:
 - using FCA to investigate the structure of FD in XML data sets
 - CG based representation and querying of relational databases



IN THIS PAPER

- CG based representation of semi-structured data model
- CG based XML structure representation
- Visual Interface XSD Builder: intuitive graphical web application which represents the XML data structure in form of Conceptual Graphs and gives the possibility of constructing queries also in form of Conceptual Graphs on the selected data structure.

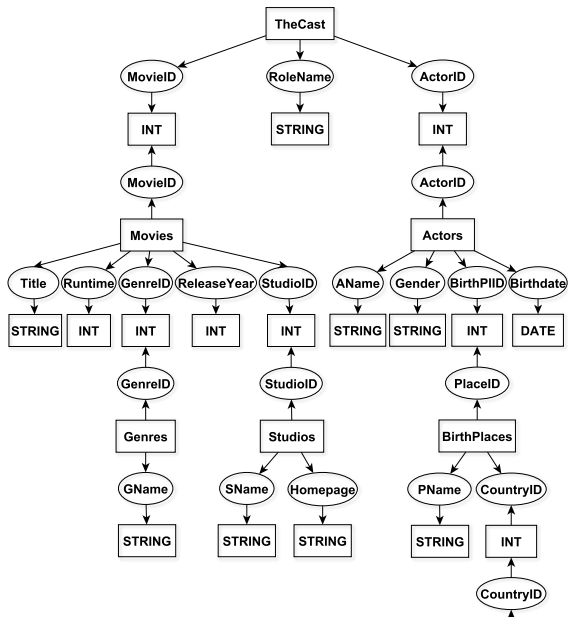


CONCEPTUAL GRAPHS

- Conceptual graphs express meaning in a form that is logically precise, humanly readable, and computationally tractable;
- Conceptual graphs are bipartite directed graphs with two kinds of nodes: concepts and relations;
- The goal of conceptual graphs is to provide a graphical representation for logic which is able to support human reasoning;
- There are a lot of possible applications of such a logical representation system
- One interesting application is a consistent, graphical interface for database interaction, which has not been completely developed until today.



CG FOR RELATIONAL DATABASES SCHEME



GRAPHIC REPRESENTATION OF SEMI-STRUCTURED DATA MODEL BY CONCEPTUAL GRAPHS

Three steps procedure:

- 1 Identify all complex objects. Every object will be modeled as a CG **concept** and graphically represented as a rectangle.
- 2 Define the relationships between the objects. The relationship between two or more **concepts** will be represented by means of a **relation** and graphically represented as an oval. A directed arc from the first concept node to the relation node and another arc from the relation to the second concept node will represent this relationship.
- 3 Identify all descendant objects of the complex objects, defined at Step 1.



EXAMPLE

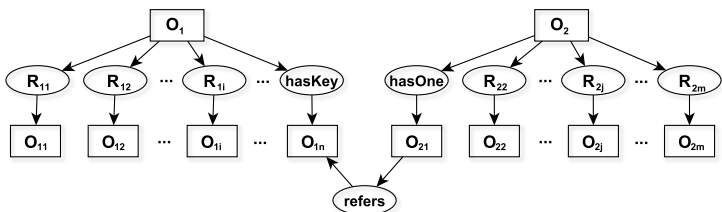


Figure 2: CG describing a connection between the related objects O_1 and O_2 .

CONCEPTUAL GRAPHS BASED REPRESENTATION OF XML DATA STRUCTURE

Step 1. Identify all complex elements.

- The **concept type** can represent: a complex or simple *element (E)*; an *attribute (A)*; a *root element (R)*; the *type of the data (T)*; or an *enumeration constraint (N)*.
- The **concept referent** can be: the *name of the correspondent node (element or attribute)*; the *data type of the corresponding simple element or attribute*; or the *set of acceptable values of the corresponding element in the case of enumeration*.



CONCEPTUAL GRAPHS BASED REPRESENTATION OF XML DATA STRUCTURE

Step 2. Define the relationships between the nodes.

- (hasKey): specifies the key node of a complex element;
- (hasOne): models a one-to-one relationship, which may exists between two elements or element and attribute;
- (hasMore): represents a one-to-many relationship.
- (refers): models a reference between two or more complex elements.
- (isOptional): designs the case when the minimum number of occurrences of an element is 0.
- (hasType): defines the data type of a simple element or an attribute.
- (hasChoice) and (isPossibility): specify the child elements of a choice compositor.
- (isEnum): models the enumeration constraint, which limits the content of an element to a set of acceptable values.



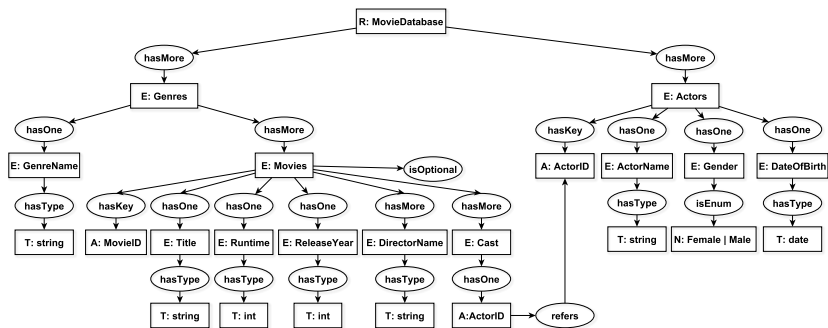
CONCEPTUAL GRAPHS BASED REPRESENTATION OF XML DATA STRUCTURE

Step 3.

- *Identify all descendant nodes* (elements or attributes) of the complex elements, defined at *Step 1*,
- give the *data type* for each one
- specify also the potential *restrictions* of the corresponding nodes (elements or attributes).



THE *MoviesDB* XML SCHEMA CG

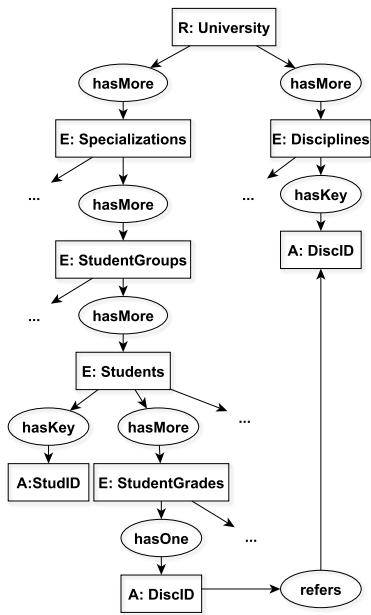


ANOTHER EXAMPLE: UNIVERSITY

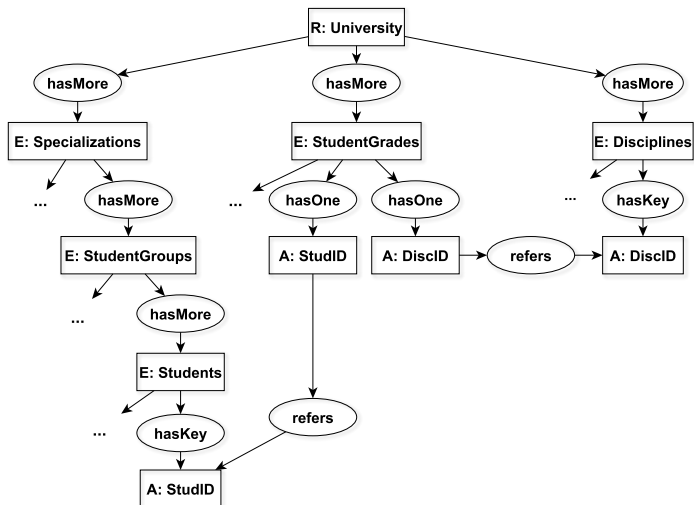
- 1 Concept: **University** - root element of the XML Schema; Specializations, Disciplines - complex elements;
- 2 Students are organized in groups. **Specialization** becomes root element, etc.



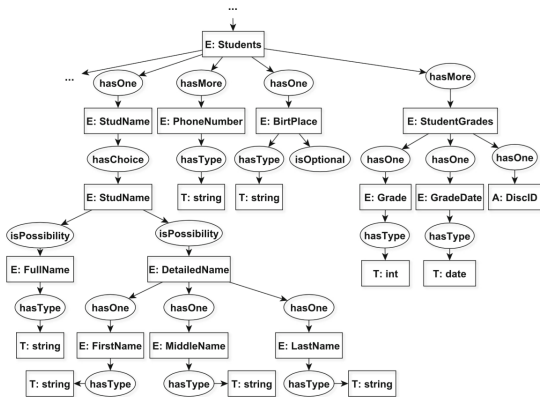
WITH EMBEDDING



WITHOUT EMBEDDING



THE SNIPPET OF THE UNIVERSITY XML SCHEMA CG ILLUSTRATING THE STRUCTURE OF THE STUDENTS AND STUDENTGRADES ELEMENTS.



VISUAL INTERFACE

- XSD Builder: provides a user-friendly interface in form of CGs and gives the user the possibility to build the graphs of personalized XML Scheme.



Thank You!

