How Do Developers Document Database Usages in Source Code?

Mario Linares-Vasquez, Boyang Li, Christopher Vendome, and Denys Poshyvanyk
Database-centric application (DCA)

DCAs are software systems that rely on databases to persist records using database objects.
DCAs are software systems that rely on databases to persist records using database objects.
Challenges
### Challenges

**STUDENT**

<table>
<thead>
<tr>
<th>ID</th>
<th>Num</th>
<th>PWD</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

DBManager.getAllInfoByStudentID ()
Challenges

DBManager.getAllInfoByStudentID ()

<table>
<thead>
<tr>
<th>ID</th>
<th>Num</th>
<th>PWD</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Num</th>
<th>PWD</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STUDENT

ST_LOGIN

ST_DETAILS
Challenges

STUDENT

<table>
<thead>
<tr>
<th>ID</th>
<th>Num</th>
<th>PWD</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

ST_LOGIN

<table>
<thead>
<tr>
<th>ID</th>
<th>Num</th>
<th>PWD</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

ST_DETAILS

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

DBManager.getAllInfoByStudentID ()

getSTLogin()

getSTDetails()
Challenges

UI.student.buttonClickShowAllInfo()

UI.student.queryAllInfoByID()

DBManager.getAllInfoByStudentID()

getSTLogin()

getSTDetails()
Challenges

• How the model is described by a schema

• How the database is used in the source code
Related works
Extracting the extended entity-relationship model from a legacy relational database

Rita Abu
Department of Computer Science, Advanced Analytical Systems and Applications Laboratory, University of Calgary, Calgary, Alberta, Canada T2N 1N4
Received 2 May 2001; revised and accepted 28 May 2001

Abstract

The maintenance of an existing database depends on the depth of understanding of its characteristics. Such understanding usually lasts when the developer departs. The situation becomes worse when the related documentation is missing. This paper explores the issue by extracting the extended entity-relationship model from the relational schema. We developed algorithms that extract a documentation of an existing legacy database in order to identify existential classes of all attributes in the relational schema, to locate foreign keys, and to examine the relationships between attributes. The extended entity-relationship model is a set of rules that describe the relationship between existences and binary relations between the class entries. The existence and existence of each class is related to the main table and the table attribute. Thus, the model helps to organize the data and make the data more accessible. Finally, the last step is to process each foreign key with appropriate updating the original conceptual data model. In addition, the extended entity-relationship model will be used for future applications. This model provides a tool for the identification of the entities and the relationships between the attributes. It is a valuable tool for the identification of the entities and the relationships between the attributes.

1. Introduction

Organizations are turning to system re-engineering as a means of upgrading that existing information system to situations where it appears to be a less expensive alternative to system replacement. Reverse engineering is viewed as a critical part of the whole systems re-engineering process because successful systems re-engineering highly depends on effective reverse engineering. In general, reverse engineering can be classified into two main steps: (1) understanding the existing system and (2) designing a new system. This paper describes the second step.
Related works
No previous work has been done to understand database documentation practices at source code level.
Goal

How Do Developers Document Database Usages in Source Code?
Methodology

GitHub

381,161 projects
Methodology

GitHub

Identified the projects using SQL

381,161 projects

18,828 projects
Methodology

GitHub

Identified the projects using SQL

381,161 projects

≥ 1 projects

18,828 projects

≥ 1 projects

3,113 projects
Methodology

GitHub

3,113 projects

A survey

A mining-based analysis
Methodology

GitHub
3,113 projects

A survey
147 developers

A mining-based analysis
33,045 methods
Methodology

GitHub
3,113 projects

A survey
A mining-based analysis

with
on

147 developers
33,045 methods

Results
Results
Research Questions

RQ1. Do developers document database-related methods

RQ2. Do developers update comments for database-related methods

RQ3. How difficult is to understand the database schema constraints along call-chains
RQ1. Do developers comment methods in source code that locally execute SQL queries and statements?

SQ1. Do you add/write documentation comments to methods in the source code?

Yes 122 (82.99%)
No 25 (17.01%)

SQ2. Do you write source code comments detailing database schema constraints?

Yes 32 (21.77%)
No 115 (78.23%)
RQ1. Do developers comment methods in source code that locally execute SQL queries and statements?
RQ1. Do developers comment methods in source code that locally execute SQL queries and statements?

“The database schema and documentation takes care of that. I can always look at the table definition very easily.”

“Comments related to the database schema and its constraints I consider to be irrelevant to the code using it. The schema, its details, and any quirks about it should be outlined in a separate document.”

Yes 32 (21.77%)  
No 115 (78.23%)
RQ1. Do developers comment methods in source code that locally execute SQL queries and statements?

In the 3,113 projects, we identified a total of 33,045 methods invoking SQL queries/statements.
RQ2. Do developers update comments of database-related methods during the evolution of a system?

SQ3. How often do you find outdated comments in source code?

SQ4. When you make changes to database related methods, how often do you update comments?
RQ2. Do developers update comments of database-related methods during the evolution of a system?

SQ3. How often do you find outdated comments in source code?

SQ4. When you make changes to database related methods, how often do you update comments?

Never: 118 (80.27%)
Rarely: 29 (19.73%)
Sometimes: 
Fairly Often: 
Always: 

Never: 31 (21.08%)
Rarely: 
Sometimes: 
Fairly Often: 
Always: 116 (78.92%)
RQ2. Do developers update comments of database-related methods during the evolution of a system?

3,113 projects

8.5% Had explicit releases

264 projects
RQ2. Do developers update comments of database-related methods during the evolution of a system?
RQ2. Do developers update comments of database-related methods during the evolution of a system?
RQ2. Do developers update comments of database-related methods during the evolution of a system?
RQ2. Do developers update comments of database-related methods during the evolution of a system?
RQ2. Do developers update comments of database-related methods during the evolution of a system?

106 methods

frequency that the comments were updated when the method was modified
RQ2. Do developers update comments of database-related methods during the evolution of a system?

106 methods

frequency that the comments were updated when the method was modified
RQ3. How difficult is it for developers to understand propagated schema constraints along call-chains?

SQ5. How difficult is it to trace the schema constraints (e.g., foreign key violations) from the methods with SQL statements to top-level method callers.
Lessons learnt

(i) Documenting database usages and constraints is not a common practice in source code methods

(ii) Developers do not update comments when changes are done to database-related methods

(iii) Tracing schema constraints through call-chains in the call graph is not an easy task in most of the cases
Lessons learnt

(i) Documenting database usages and constraints is not a common practice in source code methods

(ii) Developers do not update comments when changes are done to database-related methods

(iii) Tracing schema constraints through call-chains in the call graph is not an easy task in most of the cases
Summary

Challenges

UI.student.buttonClickShowAllInfo()  
UI.student.queryAllInfoByID()  
DBManager.getAllInfoByStudentID()  
getSTLogin()  
getSTDetails()  

STUDENT

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Fname</th>
<th>Gender</th>
<th>Address</th>
<th>Yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

ST_LOGIN

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Fname</th>
<th>Gender</th>
<th>Address</th>
<th>Yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

ST_DETAILS

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Fname</th>
<th>Gender</th>
<th>Address</th>
<th>Yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Summary

Challenges

Methodology

GitHub
3,113 projects
A mining-based analysis
33,045 methods
Results

A survey
wth
147 developers
Results
Summary

Challenges

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>PWID</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

STLOGIN

STDETAILS

STUDENT

UI.student.buttonClickShowAllInfo()

UI.student.queryAllInfoByID()

DBManager.getAllInfoByStudentID()

getSTLogin()

getSTDetails()

Methodology

GitHub

3,113 projects

A survey with 147 developers

A mining-based analysis on 33,045 methods

Results

Research Questions

RQ1. Do developers document database-related methods

RQ2. Do developers update database-related methods

RQ3. How difficult is it to understand database schema along call-chains
Summary

Challenges

STUDENT

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Phone</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Phone</th>
<th>Gender</th>
<th>Address</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ST_LOGIN

ST_DETAILS

UI.student.buttonClickShowAllInfo()

UI.student.queryAllInfoByID()

DBManager.getAllInfoByStudentID()

getSTLogin()

getSTDetails()

Methodology

GitHub

3,113 projects

A survey

147 developers

A mining-based analysis

33,045 methods

Results

Research Questions

RQ1. Do developers document database-related methods

RQ2. Do developers update database-related methods

RQ3. How difficult is to understand database schema along call-chains

Lessons learnt

(i) Documenting database usages and constraints is not a common practice in source code methods

(ii) Developers do not update comments when changes are done to database-related methods

(iii) Tracing schema constraints through call-chains in the call graph is not an easy task in most of the cases

Documentation

Automation

Calling context