A Database Design Methodology

krahim@umd.edu

N. Roussopoulos & R. Yeh
IEEE Computer 1984

1

A Complete Methodology

Area of application:

Design of database with its transactions.

Perspective:

The method assumes that the primary purpose of the future system is to automate current or planned activities of the enterprise. The method assumes (as do all database design methodologies) that different views on the enterprise, conflicts, and political differences will be resolved during the database design process.

Life-Cycle:

- **Project Progress Report: Phase I**
  - Environment & Requirement Analysis
  - System Analysis & Specification
- **Project Progress Report: Phase II**
  - Conceptual Modeling
  - Logical Modeling
  - Task Emulation
  - Optimization (NOT REQUIRED for the 424 project)
- **Project Progress Report: Phase III**
  - Implementation
  - 1 Convert Emulated tasks to code
  - 2 Bulk-Loading & Tuning (LIMITED for the 424 project)
  - 3 Testing
- **Limitation:**
  - The methodology does not cover implementation, testing, maintenance, and project management.

2

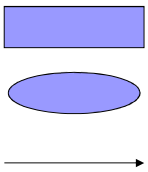
I.1. Environment & Requirements Analysis

- The purpose of this phase is to investigate the **information needs** of and thev**activities** within the enterprise and determine the **boundary** of the design problem (not necessarily identical to the boundary of the future computerized system, if any).
- **Input:**
  Information describing the current status of the enterprise, possible inefficiencies, plans for the future, and constraints that have to be satisfied in conducting business.
- **Output:**
  A **Top-Level Information Flow Diagram** describing the major documents and functions, and the boundary of the design problem. **The documents** include the major input, output, and internal documents. **The functions** model the major activities within the enterprise.
- **Function:**
  To collect the information about the enterprise and design the top-level information flow diagram.

3

Guidelines:

- **Techniques:** collect information by contacting interviews of people at all levels of the organization; analyze questionnaires; review short and long term plans, business anuals, files, forms, etc.
- **Tools:** express a top-level information flow diagram to capture the functions and important documents of the enterprise, and to start the design with the i/o documents and work from the outside in towards a "top-level" design.
- The tool we use for designing the top-level information flow diagram is the following **graphic formalism** for representing **structures and processes**:



structure

process

information flow

- Two structures are **never** directly connected.
- Two processes are **never** directly connected.

4

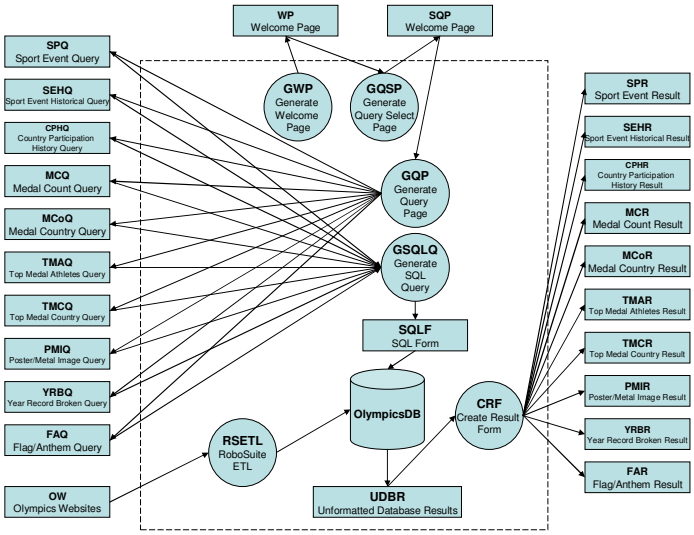
Example

Analysis, Design and Implementation of the OlympiChronicles DB System OLYMPICHRONICLES

Craig Shapiro
Steffanie Orellana

5

Top Level Information Flow



6

I.2. System Analysis & Specification

The purpose is to divide the functions from the Top-Level Information Flow Diagram hierarchically into tasks. The tasks should be reasonably independent to minimize the task-to-task interfaces (documents). During the division process, the documents used by each function are also broken down. The process is continued until each task is small enough to be clearly understood, and until each document can be conveniently expressed in terms of data elements that cannot be further divided. The result is a detailed Task Flow Diagram and a set of forms describing the documents and the tasks.

- **Input:**
  - The Top-Level Information Flow Diagram and information about the documents and functions from step 1.1
- **Output:**
  - Task Forms; Document Forms; Document and, the detailed Task Flow Diagram.
- **Function:**
  - Decompose functions and documents. Specify the resulting Task and component Document Forms. and Specify Document. Design detailed Task Flow Diagram.
- **Guidelines:**
  - **Technique:** top-down hierarchical decomposition.
  - **Tools:** Task Forms; Document Forms, and the graphical formalism for Task Flow Diagrams.

7

Examples of Task Forms

3.2.2.2 ETL Task

TASK NUMBER:	ETLT
TASK NAME:	Extract, Transform, and Load Task
PERFORMER:	Kapow RoboSuite 5.5
PURPOSE:	To extract data, transform or reformat it and load it into the OlympicsDB
ENABLING COND:	The creation of the OlympicsDB and any addition of data or updates to the OlympicsDB.
DESCRIPTION:	This tool (Kapow RoboSuite 5.5) extracts specific data from a web page, and load it into a predefined data relation or table.
FREQUENCY:	Once for the creation of the OlympicsDB and during any updates.
DURATION:	Varies
IMPORTANCE:	Critical
MAXIMUM DELAY:	N/A
INPUT:	A selected web page
OUTPUT:	Data into a relation in the OlympicsDB
DOCUMENT USE:	HTML documents
OPS PERFORMED:	Data extraction, data transformation, and data loading.
SUBTASKS:	Web pages Research
ERROR COND:	None

8

Another Task

3.2.2.8 Create Query Result Form Task

TASK NUMBER:	CRFT
TASK NAME:	Create Result Form
PERFORMER:	Server side script
PURPOSE:	Provide a formatted result from the OlympicsDB.
ENABLING COND:	Database completing operations.
DESCRIPTION:	Formats output of the extracted data from the OlympicsDB to a form that can be interpreted by a web browser.
FREQUENCY:	Once per user query submission.
DURATION:	Depends on the complexity of the query result.
IMPORTANCE:	Critical
MAXIMUM DELAY:	5-10 seconds
INPUT:	OlympicsDB data
OUTPUT:	(SPR) Sport Event Result; (SEHR) Sport Event Historical Result; (CPHR) Country Participation History Result; (MCR) Medal Count Result; (MCoR) Medal Country Result; (TMAR) Top Medal Athletes Result; (TMCR) Top Medal Country Result; (PMIR) Poster/Medal Image Result; (YRBR) Year Record Broken Result, or (FAR) Flag/Anthem Result.
DOCUMENT USE:	None
OPS PERFORMED:	Transform data from the OlympicsDB output format to a web browser compatible format.
SUBTASKS:	None
ERROR COND:	If OlympicsDB_output=unknown, then produce error message and stop.

9

Rule of Thumb for Task Decomposition

- Many performers are required to carry out the task and each performer has different skills, or each can carry out a part independently.
- Different levels of authorization exist for carrying out different parts of the task.
- Different enabling conditions activate parts of the task.
- Different frequencies and durations apply to different parts of the task.
- Input documents are not used uniformly within the task.
- Different documents are used for different parts of the task.
- Many diversified operations are carried out within the task.
- Many subtasks are controlled by the task.

10

Examples of Document Forms

SPQ: Sport Event Query
Sport
Event
Year
Site
GM
GMCountry
GMResult
SM
SMCountry
SMResult
BM
BMCountry
BMResult
OR
WhenORBroken
WR
WhenWRBroken

SEHQ: Sport Event History Query
Sport
Event
Year
Site
GM
GMCountry
GMResult
SM
SMCountry
SMResult
BM
BMCountry
BMResult

MCoQ: Medal Country Query
Country
Year
Site
SumGM
SportGM
EventGM
SumSM
SportGM
EventGM
SumBM
SportBM
EventBM

TMAQ: Top Medal Athletes Query
Athlete
Year
Site
Sport
Event
GM or SM or BM
SumNumMedals >= 3

CPHQ: Country Participation History Query
Country
YearFirstParticipated
Year
Site
Sport
Event
SumNumGames

TMCQ: Top Medal Country Query
Sport or Event
Country
SumNumMedals >= 1

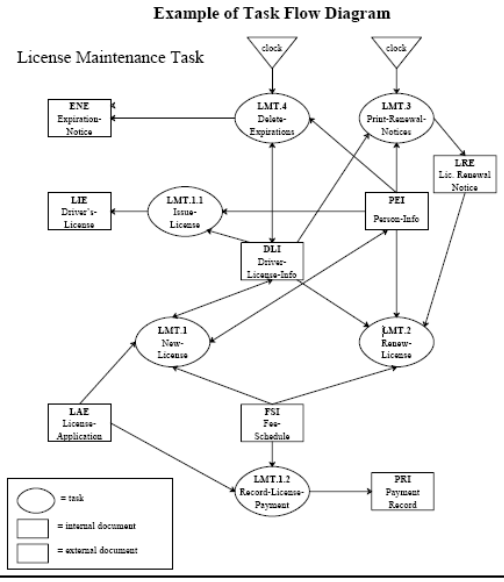
PMQ: Poster Medal Query
Year
Site
Poster
Medal

MCQ: Medal Count Query
Year
Site
SumGM
SumSM
SumBM

YRBQ: Year Record Broken Query
Sport
Event
Year
Site
OR

FAQ: Flag Anthem Query
Country
Flag
Anthem

Task Flow Diagram



Phase II.1 Conceptual Modeling

The purpose of this phase is to design a **conceptual schema** of the database. We will use the E-R data model.

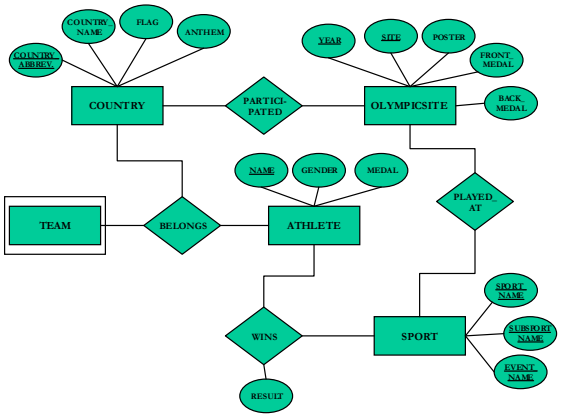
- **Input:**
  - The Document Forms
- **Output:**
  - A Conceptual Schema described in terms of the E-R data model
- **Function:**
  - To design the Conceptual Schema from the Document Forms
- **Guidelines:**
  - **Techniques** for conceptual schema design. E.g. **semantic data modeling and normalization!**

13

The Conceptual Model

Map Data Documents into E-R:

- Find Entities, their keys, and attributes
- Find Relationships, their keys, and attributes
- Discover FD's



14

Phase II.2 Logical Modeling

The purpose of this phase is to convert the **conceptual schema to a logical data model** of the database. We will map the E-R schema to a Relational schema.

- **Input:**
  - The E-R diagrams
  - FDs discovered in II.1
  - 1-1, 1-many, and many-many constraints of the relationships
- **Output:**
  - A Relational Schema (Logical Model) corresponding to the E-R model
- **Function:**
  - Map the E-R model to tables, their keys, and FDs
  - Normalize the relations to obtain BCNF or at least 3NF relations.
- **Guidelines:**
  - Algorithm for mapping E-R to relations and normalization

15

Relational (Logical) Model

COUNTRY

COUNTRY_ABBREV	COUNTRY_NAME	FLAG	ANTHEM

OLYMPIC_SITE

YEAR	SITE	POSTER	FRONT_MEDAL	BACK_MEDAL

SPORT

SPORT_NAME	SUBSPORT	EVENT_NAME

ATHLETE

NAME	GENDER	MEDAL

TEAM

NAME	GENDER	MEDAL

PLAYED_AT

YEAR	SPORT_NAME	SUBSPORT_NAME	EVENT_NAME

PARTICIPATED

COUNTRY_ABBREV	COUNTRY_NAME	FLAG	ANTHEM
YEAR	SITE	POSTER	FRONT_MEDAL	BACK_MEDAL

BELONGS

NAME	GENDER	MEDAL

WINS

NAME	GENDER	MEDAL
SPORT_NAME	SUBSPORT	EVENT_NAME

Functional Dependencies

For Country entity:

Country_Abbreviation → Country_Name

Country_Abbreviation → Flag

Country_Abbreviation → Anther

Country_Abbreviation → First_Year_Participated

For OlympicSite entity:

Year → Site

Year → Poster

Year → Medal

16

Phase II.3 Task Emulation

The purpose of this phase is to map the database design and the software that performs the tasks **before** any database implementation starts. In other words, before creating a schema in the DBMS and writing the application programs. This gives the opportunity to correct the logical schema flaws such as incomplete, superfluous, or even dead wrong. Doing the design of both the database schema and the applications simultaneously, complements these two orthogonal specifications and catches most of the errors before the beginning of the implementation.

- **Input:**
  - The Logical Schema from the previous phase
  - The Task Forms
- **Output:**
  - The set of design specifications of the pieces of software that performs the tasks described in the task forms. The design specifications can be given in terms of abstract programs with embedded sequences of DML statements,

17

Phase II.3 Task Emulation
(cont)

- **Function:**
  - Use the Task Forms describing the tasks. Formulate for each task an abstract program including embedded sequences of DML statements that perform the task using the conceptual schema. (During this phase small corrections of the conceptual schema may be needed to support the tasks: *validation*).
- **Guidelines:**
  - **Techniques:** those that apply to the use of the particular DML.

18

Task Emulation

Extract, Transform, and Load
Start RoboSuite 5.5
Configure RoboSuite 5.5
for each website bookmarked
for each webpage on website [query results]
RoboSuite.url = webpage.url
set values to look for
extract information to a predefined table.

Web Pages Research
{Google query to find Summer Olympic Games sites}
For each website found in Google
if website has relevant data and if website has complete data to be used by the OlympicsDB
Bookmark
else
skip

19

Task Emulation

Generate SQL
If query == Sport_Event_Query
SELECT year, site, sport_name, subsport_name, event_name, subevent_name, medal
FROM Sports, OlympicSites, Medal, Wins, Played_At
WHERE year=year_chosen and site=site_chosen and sport_name=sport_name_chosen and subsport_name=subsport_name_chosen and event_name=event_name_chosen and subevent_name=subevent_name_chosen and medal=medal_chosen
Else if query == Sport_Event_Historical_Query
SELECT year, site, sport_name, subsport_name, event_name subevent_name, medal
FROM Sports, OlympicSites, Medal
Else if query == Country_Participation_History_Query
SELECT C.year, site, country_abbreviation, C.country_name, year_first_participated, count(country_name)
FROM Country C, OlympicSite O, Participated P
GROUP BY P.year
Else if query == Medal_Count_Query
SELECT year, site, country_name, count(medal)
FROM OlympicSite, Country, Medal, Participated, Wins, Belongs
WHERE year=year_chosen and site=site_chosen and country_name=country_name_chosen
GROUP BY medal
Else if query == Medal Country History Query
SELECT year, site, country_name, medal
FROM OlympicSite, Country, Medal, Participated, Wins Belongs

Generate SQL (cont…)
Else if query == Top_Medal_Athletes_Query
SELECT year, site, first_name, last_name, medal
FROM OlympicSite, Athlete, Belongs, Participated, Medal, Sport, Wins, Played_At
HAVING count (medal) > 3
Else if query == Top_Medal_Country_Query
SELECT year, site, country_name, event_name, count(medal)
FROM OlympicSite, Country, Sport, Medal, Win, Participated Played_At
Else if query == Poster/Medal_Image_Query
SELECT year, site, poster, front_medal, back_medal
FROM OlympicSite
WHERE year=year_chosen and site=site_chosen
Else if query == Year_Record_Broken_Query
SELECT
FROM
WHERE
Else if query == Flag/Anthem_Query
SELECT year, site, country_name, flag, anthem
FROM OlympicSite, Country
WHERE year=year_chosen and site=site_chosen and country_name=country_name_chosen

20

Phase III.1 Implementation

The purpose of this phase is to translate the conceptual schema and the task design specifications into actual schema definitions and application program modules.

- **Input:**
  - The relational (logical) schema
  - the task specifications
- **Output:**
  - The DDL statements for the DBMS
  - The tasks programmed in terms of the host-language with embedded SQL statements.
- **Function:**
  - Map the database schema into the relational (logical) schema.
  - Translate the task designs into the host-language modules.
- **Guidelines:**
  - Use of automated tools like the SSDB for the schema and debuggers like the SQL Developers

21

Phase III.2 Bulk Loading & Testing

The purpose of this phase is to load the real stuff and fine tune its performance.

- **Input:**
  - The schema definitions and the application programs from the previous step
  - A set of test data.
- **Output:**
  - The database system.
- **Function:**
  - Almost always this is very painful step which can take several weeks or even months. The biggest problem is data errors that need to be cleaned before entered. Bulk loading implies high volume of data (unlike your CMSC 424 project).
- **Guidelines:**
  - **Technique:** patience!
  - **Tool:** bulk loaders and scripting languages.

22

DON'T FORGET

- The secret behind successful Database Design is careful analysis, specification, and design. These are done in the phases I and II of the methodology. Having done a careful analysis on these, the development is certain to succeed.
- There are always bugs in large databases. Careful testing eliminates only the most obvious. Testing requires a systematic methodology different than the one used by Microsoft!
- Large databases are used for many years. Maintaining a database throughout its life-time typically takes several times more than its development. It is impossible to maintain a database with an undocumented design. The documents produced by this methodology is the design specification and will be the heart of the database and must be properly maintained. Without the methodology, there is no common language to exchange design specifications.

23