Pervasive Business Intelligence

Elisa Turricchia
elisa.turricchia2@unibo.it

ALMA MATER STUDIORUM - Università di Bologna
Doctorate in Electronics, Computer Science and
Telecommunications – XXV Cycle

TUTOR: Prof. Dario Maio
CO-TUTORS: Prof. Stefano Rizzi, Prof. Matteo Golfarelli

Second Year Ending Seminar
Bologna, Italy, 21 October 2011
Outline

- Introduction
- Research activity: Pervasive Business Intelligence (BI)
  - Collaborative BI
  - Lean BI
  - OLAP query personalization
- Results
- Future works
Introduction (1)

- **Business Intelligence (BI):** a set of tools and activities to transform business data in useful information for the decision making.

- **Data Warehouse (DW):** a collection of multidimensional data supporting the decision making. It represents the temporal evolution of the data.

Which is the product that maximizes the profit?

Mining the relationship between revenues of two different products.
On-Line Analytical process (OLAP) Systems: organize data in a multidimensional structure and allow the user to interactively navigate the data to support the business analysis.

Dimensional Fact Model (DFM): graphical specification to represent multidimensional data.
Introduction (3)

**OLAP operations:**
- **Drill-down:** disaggregates data  
  (e.g., category → subcategory)
- **Roll-up:** aggregates data  
  (e.g., customer → city)
- **Slice:** selection predicates on data  
  (e.g., category=“Drinks”)

---

<table>
<thead>
<tr>
<th>category</th>
<th>subcategory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
<td><strong>Measures</strong></td>
</tr>
<tr>
<td>Beer and Wine</td>
<td>14.029,08</td>
</tr>
<tr>
<td>Carbonated Beverages</td>
<td>6.236,35</td>
</tr>
<tr>
<td>Drinks</td>
<td>5.642,29</td>
</tr>
<tr>
<td>Hot Beverages</td>
<td>9.261,74</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Research Activity (1)

- Lean BI
- Collaborative BI
- OLAP query personalization

Pervasive BI
Pervasive in terms of:

**Collaborative BI**
- Supporting collaborative contexts to favour new business opportunities.

**Lean BI**
- Promoting the BI tools distribution even in small and medium companies.

**OLAP query personalization**
- Fostering the use of BI tools by different users and in different contexts.
Collaborative BI - Goal

- Extend the decision-making process beyond the boundaries of a single company.
- Improve collaborative and inter-business activities (e.g., multi-national companies).
- Support companies to organize and coordinate themselves to share opportunities, respecting their own autonomy and heterogeneity.

Example: *Health-care context*

Monitor clinical data of different hospital units to mine global phenomena (e.g., epidemic diseases)
Collaborative BI – State of the art

Data warehouse integration and federation

- Integration by *Extraction Transformation and Loading* (ETL) process
  - Hardly feasible in a dynamic network, with autonomous and independent nodes

- Centralized architecture
  - Non-scalable architecture, security issues

- Summarizability for non-distributive aggregate functions in distributed data warehouse has not been deeply investigated in the literature
  - Limited analysis on aggregate data
Collaborative BI – Contribution (1)

The framework:

Business Intelligence Network (BIN): a dynamic, collaborative network of peers, each hosting a local, autonomous BI system.
The framework features:

- Each peer relies on a local multidimensional schema that represents the peer's view of the business.
- Users transparently access business information distributed over the network in a pervasive and personalized fashion.
- Access is secure, depending on the access control and privacy policies adopted by each peer.
- A BIN is decentralized and scalable because the number of participants, the complexity of business models, and the workload can change.
Collaborative BI – Contribution (3)

The framework functionalities:

1. A user formulates an OLAP query $q$ by accessing the local multidimensional schema exposed by his peer.

2. Query $q$ is processed locally on the data warehouse of peer (target peer).

3. At the same time $q$ is forwarded to the network (source peer).

4. Each involved peer locally processes the query on its data warehouse and returns its results to the original peer (target peer).

5. The results are integrated and returned to the user.
Collaborative BI – Contribution (4)

Query Language

\[ q_1 = \text{HOSPITALIZATION}, \{ \text{region, year} \}, \text{\{gender = 'Female'\}}, \text{cost, \{\{cost, sum\}\}} \]
Collaborative BI – Contribution (4)

Mapping Language

The peers are characterized by different multidimensional schemata; a mapping language solve the heterogeneity between different schemata.
Collaborative BI – Contribution (5)

Reformulation algorithm

Adapt a query to the multidimensional schema of each peer; a reformulation algorithm to transform a query formulated on a peer to a query executable in a different peer.

Considerations:
- The query language is closed under reformulation
- We demonstrated the correctness of the reformulation algorithm.

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>${a_1^i, \ldots, a_j^i}$ drill-down $f_1{a_1^i, \ldots, a_k^i}$</td>
<td>$\forall \nu(a_1^i), \ldots, \nu(a_j^i), \nu(a_{j+1}^i), \ldots, \nu(DIM_k^i)$</td>
</tr>
<tr>
<td></td>
<td>$(S.dt_1^i(x_1^i), \ldots, S.dt_k^i(x_k^i), S.ft(x^i)),$</td>
</tr>
<tr>
<td></td>
<td>$\nu(a_1^i) = f(\nu(a_1^i), \ldots, \nu(a_j^i)), 1, \ldots, \nu(a_k^i) = f(\nu(a_1^i), \ldots, \nu(a_j^i)).$</td>
</tr>
<tr>
<td></td>
<td>$\exists \nu(DIM_{i+1}^i), \ldots, \nu(DIM_k^i)(T.dt_1^i(x_1^i), \ldots, T.dt_k^i(x_k^i), T.ft(x^i))$</td>
</tr>
<tr>
<td>${a_1^j, \ldots, a_j^j}$ equi-level ${a_1^j, \ldots, a_k^j}$</td>
<td>$\forall \nu(a_1^j), \ldots, \nu(a_k^j), \nu(DIM_{i+1}^j), \ldots, \nu(DIM_k^j)$</td>
</tr>
<tr>
<td>${a_1^j, \ldots, a_j^j}$ roll up ${a_1^j, \ldots, a_k^j}$</td>
<td>$(S.dt_1^j(x_1^j), \ldots, S.dt_k^j(x_k^j), S.ft(x^j)),$</td>
</tr>
<tr>
<td>${a_1^j, \ldots, a_j^j}$ drill-down ${a_1^j, \ldots, a_k^j}$</td>
<td>$\exists \nu(a_1^j), \ldots, \nu(a_j^j), \nu(DIM_{i+1}^j), \ldots, \nu(DIM_k^j)$</td>
</tr>
<tr>
<td>${a_1^j, \ldots, a_j^j}$ related ${a_1^j, \ldots, a_k^j}$</td>
<td>$(T.dt_1^j(x_1^j), \ldots, T.dt_k^j(x_k^j), T.ft(x^j))$</td>
</tr>
</tbody>
</table>
Lean BI - Goal

- To make faster and nimbler the data warehouse design, improving the customer satisfaction
- Improve the flexibility of the data warehouse development, to better comply to the changing requirement of customers and market
- Facilitate the penetration of BI in small and medium companies
Lean BI – State of the art

Data warehouse systems are characterized by a long and expensive development process that hardly meets the ambitious requirements of today's market.

- Long delay in delivering a working system
- Missing and inadequate requirements

Further investigations on the data warehouse methodological aspects to make efficient and nimbler the development process.

A lot of methodologies, typically context-oriented
Lean BI – Contribution (1)

The methodology: Four-Wheel-Drive (4WD)

Couples traditional data warehouse methodologies (e.g., waterfall methods) to Agile approaches.
Lean BI – Contribution (2)

Principles

- **Incrementally and risk-based iteration**: developing and releasing the system in increments leads to a better management of the project risks.

- **Prototyping**: complex projects are conveniently split into smaller units or increments corresponding to sub-problems that can be more easily solved and released to users.

- **User involvement**: continuous communication and user participation increases customer satisfaction and promotes a high level of trust between the parties.
Lean BI – Contribution (3)

Principles

- **Component-reuse**: the reuse of predefined and tested components speeds up product releases and promotes cost reduction as well as software reliability.

- **Formal and light documentation**: a well-defined documentation is a key feature to comply with user requirements.

- **Automated schema transformation**: formal and automated transformations between schemata representing different software perspectives (e.g., between conceptual and logical schemata) accelerates the software development and promotes standard processes.
OLAP query personalization – Goal

Based on the past OLAP analysis of a single user, the aim is to add OLAP preferences to the current query, to reduce the result to the most relevant information for the user.
OLAP query personalization – State of the art

- Personalized visualization of OLAP results based on explicit user information
  - Past actions of the user and context have not been taken into account

- Recommendation on past logs
  - Prescriptive approach
  - Not on user profile
OLAP query personalization – Contribution (1)

1. The user formulates an OLAP query

```
SELECT AvgIncome ON COLUMNS,
    Crossjoin(OCCUPATION, members,
    Crossjoin(Descendants(RACE, AllRaces, RACE Mm),
    Descendants(RESIDENCE, AllCities, RESIDENCE Region))) ON ROWS
FROM CENSUS WHERE TIME.Year, [2009]
```

2. From the log, we extract relevant rules for the current query

3. We add best rules to the current query

```
SELECT AvgIncome ON COLUMNS,
    Crossjoin(OCCUPATION, members,
    Crossjoin(Descendants(RACE, AllRaces, RACE Mm),
    Descendants(RESIDENCE, AllCities, RESIDENCE Region))) ON ROWS
FROM CENSUS WHERE TIME.Year, [2009]
PREFERRING AvgIncome BETWEEN 500 AND 1000
AND Occ POS 'Engineer' AND RESIDENCE CONTAIN Region
```
OLAP query personalization – Contribution (2)

1. **Log mining**: data mining algorithm on the user’s query log is used to extract the set $R$ of association rules whose **support** and **confidence** are above a given threshold.

2. **Rule selection**: when that user formulates an MDX query $q$, a subset $S \subseteq R$ of rules is selected that are **pertinent** and **effective** for the current query.

3. **Fragment translation**: each rule is translated into an OLAP constructor, then coalesced and composed.

4. **Querying**: the current query is annotated with the resulting OLAP preference and then executed on the OLAP tool.
Results (1)

Collaborative BI


Lean BI

Results (2)

OLAP Query personalization


During this year I spent a **2-month** research period in France at **Université François-Rabelais Tours**, working on OLAP query personalization and recommendation.
Future works

Collaborative BI

• Rooting strategies to select the most promising neighbouring nodes during the reformulation step
• Techniques to reconcile multidimensional data returned by different peers through object fusion techniques
• Rank the peer results depending on how compliant they are with the original local query
• Deal with security depending on the degree of trust between the BIN participants

Lean BI

• Optimization model to support the scheduling of Agile data warehouse projects

OLAP Query personalization

• OLAP query recommendation