FIT3003 final Notes

Business Intelligence And Data Warehousing (Monash University)

L1: Understanding BI and DW

Why study BI & DW?

- Broadly, two categories of information systems in enterprise IT
    1. **Transaction processing systems:** System for collecting and recording information about business activities (payroll systems, inventory management systems)
    2. **Decision support system:** Systems for collating and analysing information (typically collected from transaction processing systems) to support organisational decision-making
- Approaches that work well for developing transaction processing systems often don't work for decision support systems.

What is Business Intelligence?

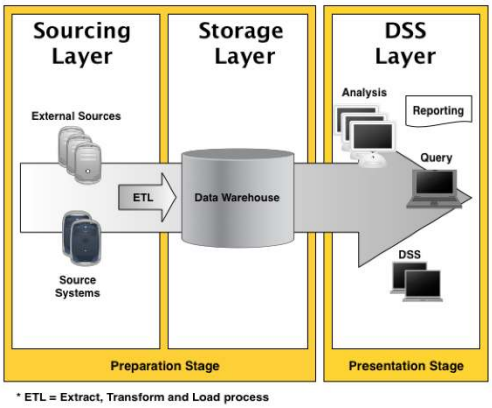
- Designed to support informed decision making in organisation through provision of functions to visualise reports and analyse organisational data
    - Huge amounts of data

What BI interfaces have in common:

- Data from lots of different sources → Internal, external, different database management approaches, structured and unstructured data
- Dashboards providing overview and monitoring of performance
- Ability to navigate from one report to another for investigation → 'Slice and Dice' and 'Drill Down'
- Visualisation of data → Graphical and Textual

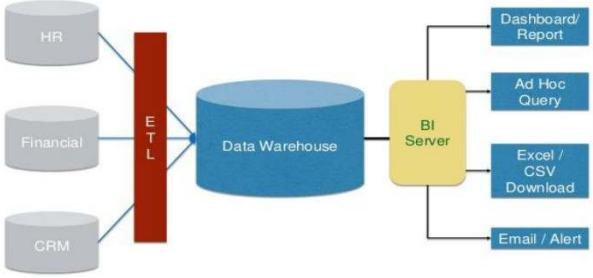
- **BI Systems have to:** 1. Acquire data 2. Present it
- **Data warehouse –** Large, centralised database for storing data from transaction processing systems
    - Provides data storage layer that provides: "**Subject-oriented, integrated, time-variant, non-volatile** collection of data in **support of management's decisions.**"

Business Intelligence System Architecture



* ETL = Extract, Transform and Load process

Business Intelligence System Architecture



Why Data Warehousing?

- Data warehouses store data from transaction processing systems

Motivation for data warehousing:

- Data in source systems is frequently inconsistent, of poor quality and stored in different formats
- Query processing on transaction-processing systems is not a good option
- DW is custom-designed for efficient data retrieval
- DW reduces complexity and costs
    - BI systems can source their data from single system with a known, consistent data structure and format

DW – subject oriented:

- Data warehouse is organised by "data subjects" that are relevant to organisation (Customer, claim, shipment, product)
- May be contrasted with the *process orientation* of many transaction-processing systems

DW – integrated:

- Data in the warehouse is structured based on a corporate-wide model, spanning the functional boundaries of source systems (the naming standards, units of measurement and periodicity)
- Compare with transaction processing system:
    - Highly normalised
    - Lack of consistency between systems

DW – time variant:

- Data in the data warehouse is characterised by the time-series nature of historical data
- Data consists of **series** of "snapshots" which are time-stamped and record values at a moment in time
- This *supports trend analysis* of data

DW – non-volatile:

- Data in a data warehouse is periodically up-loaded at a scheduled time interval (say daily) – "upserting"
- When updated → earlier versions of data is maintained
- Data warehouse is **not** continuously updated (inserts, deletes, changes) like data in transaction-processing systems (don't maintain history – edits, deletions overwrite data)

Why learn about BI/DW? Key aspects of BI/DW systems make them different to other information systems: **The task** supported, **The users**, The **development process** required

The Task – Decision Making

- Task is typically well-defined business workflow:
  - Clear responsibilities
  - Explicit information requirements
  - Explicit rules for using information to perform and action
- This allows us to develop a requirements specification for that information system design
- BI systems deal with a much more complex (design too) and ambiguously defined task than most transaction processing systems.

The Users

- Organisational decision makers: Knowledge workers, managers, executives, directors
- Impossible to specify what the above people do from moment to moment (Sales clerk: repetitive task vs. Manager: many different and unique one-off tasks)
- Organisationally powerful: If they don't want to use the BI system, they won't

The Development Process

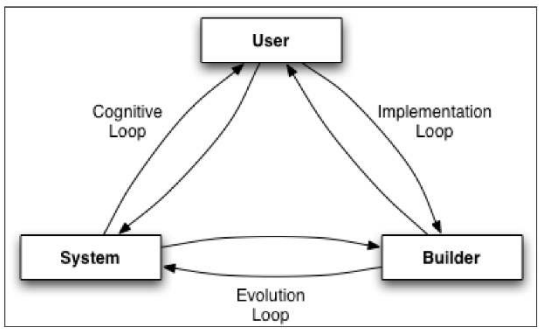
- Traditional approach to Software Development Life Cycle:
  1. Gather information requirements
  2. Produce a requirements specification document, with sign-off from client
  3. Develop the system – Database, software, hardware
  4. Test the system to make sure the system matches the requirements specification
  5. Client signs-off and takes acceptance of the system
  6. Project is closed

But for BI/DW:

- Purpose of the system is to help users understand the decision problem so they can make a better decision
- Information needed to make a good decision depends on the decision problem
- **Catch 22:**
  You don't know what information you'll need until you start using the system → You can't use system *until it's designed* → Can't be designed *until designer has gathered the information requirements*

Decision Support Systems Development

- Framework for any system that supports decision-making:



BI/DW Development:

- Any system designed to support decision making must:
  - Be closely developed with co-operation between users and developers

    - *Developed in adaptive manner* – User understanding and requirements change? System design has too as well
- Evolutionary/Adaptive development
    - Significantly different to traditional project-based development where there is clear end-date and specific deliverable (and budget)
- High profile projects
    - Significant organisational impact, affects multiple business areas
- As a result → High risk

Summary:

- BI is current approach to organisational decision support
- DW provides data infrastructure to develop various decision support and business intelligence initiatives in an organisation
- BI/DW are fundamentally different from more typical information systems:
    - Different nature of *task support*
    - Different, discretionary and powerful *users*
    - Different, 'chaotic' *development process*

L2: BI Interfaces 1

Why is the BI user interface important:

- *BI system developers* care more about the technology, *BI system users* care most about the interface

General BI UI Requirements:

- Interface must not get in the way
  - Allow "analysis at speed of thought" and "transparent interaction"
- Information presented must be easy to understand
- Must be clear where information comes from
- Interface must be easy to learn and easy to remember
- Focus must be on the design rather than engineering

Mobile BI UI Requirements:

- General BI UI requirements still apply
- Mobile BI applications tend to support very specific tasks
  - Understand decision task and environment very well
  - Work closely with users to develop the application
  - Don't limit development and testing to emulators
- Design BI application interface to conform with UI conventions of the target platform
- Choose font sizes and menu targets carefully
- Mobile application users expect "beautiful" apps

Business Intelligence UI Components:

Report -

- Document that presents information in an organised format for a specific audience and purpose
- Information organised in → Graphic, tabular, narrative formats
- **Reports output:** Generated on recurring schedule, when certain conditions met or as required
- Can be *static or interactive* (users can enter parameters, sort columns, filter content)

Tables -

- Values are arranged in grid format (rows, columns)
- Values are *encoded in text* (words, numbers)
- Rows and columns can be defined by grid lines or white space

- **Most appropriate method to:**
  - Display information when user has need to look up individual values
  - Display simple relationship between numeric and categorical values

Graphs (Charts) –

- Values are displayed within an area bordered by 1/+ axes
- Values are *encoded as visual objects* (shapes and lines) positioned in relation to the axes
- Axes provide scales that are used to label and assign values to the visual objects
- **Most appropriate method when:**
    - Message is contained in shape of the values
    - Objective is used to reveal relationships among multiple values

Dashboards –

- Visual display of the most important information needed to achieve one or more objectives which fits entirely on a single computer screen so it can be monitored at a glance
- Small, concise, clear, intuitive, and **often interactive** display
- Customised for specific person, group, or function
- Focus → Monitoring and measuring
- Consists of → Charts and tables
- **Types:**
    - **Strategic**
        - Used by managers to monitor business health & progress towards meeting strategic objectives
        - High-level measure of past and forecast performances (last week/month)
    - **Analytic**
        - Used by analysts as a way to gain access to broad range of more detailed information → Find out why things happen
    - **Operation**
        - Used to *monitor operation of a factory or service*
        - Display of information is real-time or near-real-time (NOW)
        - Displays measures, warnings, alerts
- **Purpose:** Let end-users understand their business intuitively without too much thinking, meaning they can receive critical information from dashboard without even interacting with it.

Dashboard vs Report

- Minimal interactivity and analysis needed
- Requires minimal cognitive effort
- Focus on quick communication of key data
- **Individual visualisations should work together to communicate effectively:**
    - Consider placement for comparisons
    - Avoid multiple 'screens' (tabs – horizontal, scrolling – vertical). No navigation.
- Click through to reports for substantial analysis
- Dashboards provide summary while reports let you detailed analysis

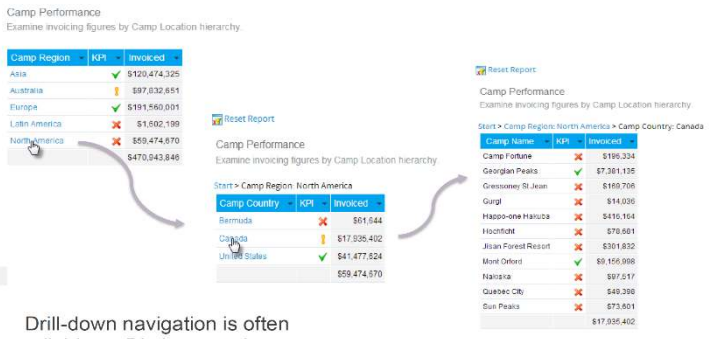
BI UI Navigation Methods:

Drill-Down

Navigation to a report with **more detail** following the levels of a pre-defined dimension hierarchy.

– Example

→ Yearly sales
→Quarterly sales
→Monthly sales
→Daily sales



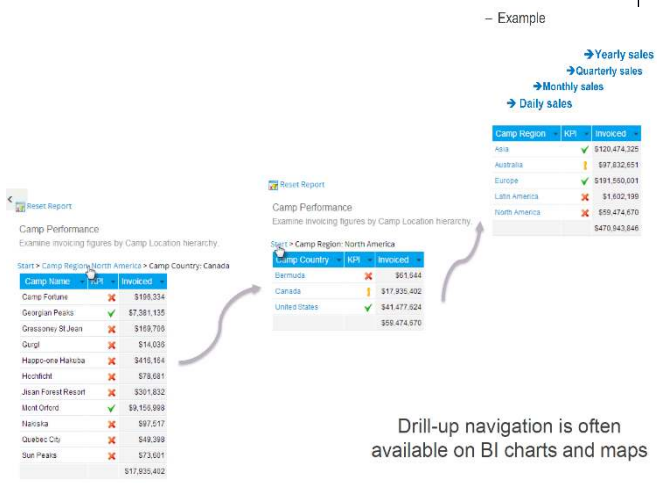
Drill-down navigation is often available on BI charts and maps

Drill-Through

Navigating from a summary report to report that shows the **detailed transaction that made up the summary.**

– Example

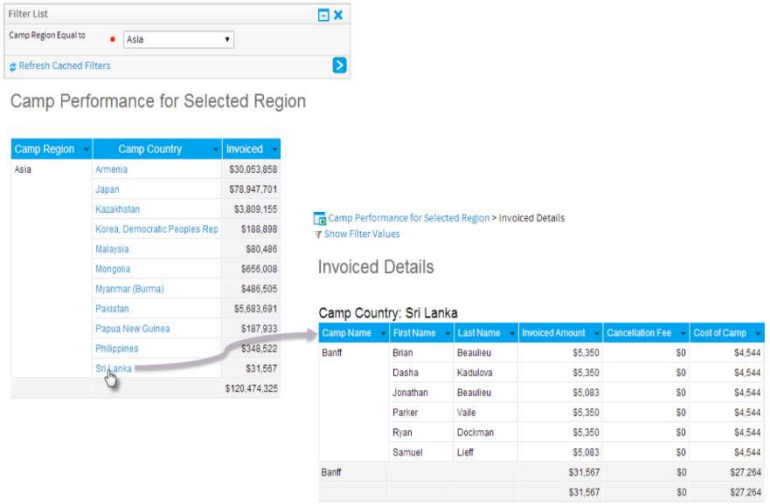
→Yearly sales
→Quarterly sales
→Monthly sales
→ Daily sales



Drill-up navigation is often available on BI charts and maps

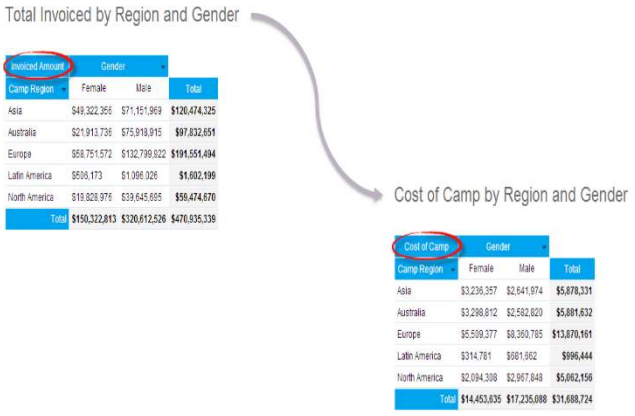
Drill-Up

Navigating to a report with **less detail** following the levels of a pre-defined dimension hierarchy.



Drill-Across

Navigating from a report based on one set of dimension attributes and measure(s) to a new report with the same set of dimension attributes but with **different measurer(s).**



Designing the BI UI:

- **Usability**
  - "Extent to which a product can be used by specified users to achieved specified goals with **effectiveness, efficiency and satisfaction** in a specified context of use"

- **Affordance**
  - A capability, offered by an artefact, to a user. Can be different for different users.
  - What the object means to you/Use of the object ("To sit on")
- **Visibility**
  - Features or properties of an object that tells or shows a user how to interact with, or the status of, an object ("Button → Push", "Switch → Flip")

UI Design Guidelines:

1. **Visibility of System Status** → System gives you feedback on what's going on (Progress bar, error message, "Loading", Spinning wheel)
2. **Match Between System and Real-World** → System should use language (words, phrases) that are familiar to user
3. **User Control and Freedom** → Support undo and redo where user could make "emergency exit" if needed
4. **Consistency and Standard** → User shouldn't have to worry if different words/actions mean the same thing
5. **Error Prevention** → Eliminate error-prone conditions or check for them and present users with confirmation option before committing to action
6. **Recognition Rather than Recall** → Interface should be easy to recognise what it's about (Easily understandable in second)
7. **Flexibility and Efficiency of Use** → Allows users, whether inexperienced/experienced to tailor frequent actions
8. **Aesthetic and Minimalist Design** → Dialogue shouldn't contain info which is irrelevant or rarely needed
9. **Help users Recognise and Recover from Errors** → Error messages should be expressed in plain languages (no codes), precisely indicate the problem and continuously suggest solution
10. **Help and Documentation** → Better if system can be used without documentation but may be necessary to provide help/documentation

Usability Testing:

- **What is it?** → Test the application
- **Why do it?** → To make sure you're on the right track
- **When should it be done?** → Draft application ready

Usability Testing - Structure:

- **Identify the scope** → The thing(s) you actually test
- **What is the purpose** → To see if it's easy to understand/use
- **Schedule and location** → Define where you run it

- **Who will participate** → Define target audience
- **How many participants?**
- **Choose scenarios**
- **Identify test metrics** → # of steps/clicks to do task, time used (too much = not good)
- **Conduct the test**
- **Compile the results**
- **Fix the usability problems**

Summary:

- BI UI consists of → *Reports, tables, graphs and dashboards* and can be *navigated* in different ways
- Usable BI UI → *Satisfied users & confident in information* provided
- Apply *usability principles* and conduct *usability test*

L3: BI Interfaces 2

Financial Markets and Financial Intermediaries:

- **Security -** A claim on the issuer's future income or assets