Tutorial on Medical Image Retrieval
- System Aspects -

Medical Informatics Europe 2005
28.08.2005

Henning Müller
Thomas Deselaers
Topics

- search and processing
- parallel/distributed architectures
- component-based architectures
- integration into existing software
- efficient search structures

- research/development system
  - transparency
  - fast method exchange
  - reproducibility
System Design

- IRMA entities
  - physical
  - logical

- System architecture
  - central components
    (database, scheduler, web server)
  - distributed components
    (workstations, data storage)

- Transparency
  - method replication
  - method execution
IRMA System Architecture

- Sources of programs (.cpp)
- IRMA database
- Images, features, & tree data
- Methods, networks & experiments
- Administrative information (cluster)

Connections:
- IRMA scheduler
- IRMA daemons
- Web server (PHP)
- IRMA GUIs
- Jobs
- SQL
- Sockets
- HTTPS
- Server to clients

Further details include:
- Administrative information (cluster)
- Sources of programs (CPP files)
- Images, features, and tree data
- Methods, networks, and experiments
- SQL connections to IRMA scheduler and daemons
- HTTP for web server communication
- IRMA GUIs for user interfaces
Example: Method Replication

- IRMA daemon
  - method ID
  - name of executable
  - local copy ?
- IRMA scheduler
  - run program
  - notify ready
- IRMA database
  - get name
  - get FTP of host
  - get-ftp sources
- IRMA daemon
  - local copy of executable
  - make
  - local copy of sources
  - FTP address

Example: Method Replication
Example: Method Execution

- IRMA program
- ready state
- IRMA scheduler
- IRMA database
- output features
- store features
- get params
- ftp-get features
- parameters & input features
- execute
- terminate
- notify ready
- send command
- job ?
- no

©2005 Hôpitaux Universitaires de Genève / RWTH Aachen
PACS Integration - Data Interfaces

- PACS
- IRMA

- Imaging modalities
- IRMA daemon
- IRMA programs
- GUIs for image retrieval

- DICOM network
- IRMA scheduler & communicator
- SQL database
- PHP web server

- Viewing application
- IRMA handle
- GUIs for data entry

- PACS archive
- Communication interface
- DICOM
- HTTP

- Server level
- Physician's workstation
- Any client
- Server level

- Application programming interface
- API
- XML-RPC
- HTTP
Graphical User Interface

Web-based GUI with extended functionality

- Relevant facts
  - the system explains its response

- Relevance feedback
  - the user explains his question

- Query refinement
  - cycle of: relevant facts - relevant feedback - resubmission
  - not always improves results: UNDO / REDO
  - interactive sessions: COMPLETE HISTORY

- Query combination
  - of independently obtained results: AND / OR
Extended Functionality vs. Usability

- Non-contextual IRMA framework vs. effectiveness
- Application-specific GUI
- Application-specific GUI vs efficiency
- Module-based GUI
- Module-based GUI vs. user’s satisfaction
- Integrated workflow
- Integrated layout
Module-Based GUIs

- Output modules
  - image, text, table ...
- Parameter modules
  - sliding bars, buttons, ...
- Function modules
  - combinations, e.g. thumbnail
- Transaction modules
  - UNDO, REDO, HISTORY ...
- Process modules
  - AND, OR, NOT ...

✓ Relevant facts
✓ Relevance feedback
✓ Query refinement
✓ Query combination
GUI Workflow

1. **Start**
2. **Parameter Modules** [relevance feedback]
3. **Search**
4. **Output Modules**
5. **OK?**
   - **No**
   - **Yes**

**End**
GUI Workflow

1. Start
2. Parameter modules (relevance feedback)
3. Output modules (relevance facts)
4. Transaction modules (undo, redo, history)
5. Search
6. Query logging
7. OK?
   - Yes: Proceed
   - No: Step back
8. End

Flow directions:
- User to System
- System to User

Decision points:
- OK?
- Step back?
GUI Workflow

start

parameter modules
[relevance feedback]

query logging

search

output modules
[relevance facts]

transaction modules
(undo, redo, history)

OK ?

merge

yes

in loop ?

yes

start loop ?

yes

no

end

no

step back ?

yes

no

user

system
Integrated Layout

• Five components
  • header bar
  • navigation bar
  • parameter field
  • status bar
  • output field

• Construction rules
  • optional components
  • in fixed order
  • with adaptive navigation
Web-Based Implementation

- **PHP web server**
  - dynamically generated web pages
  - HTML form
    ```html
    <form action="index.php" name="main page" method="post">
    ```
- **Smart template class**
  - separation of data and its visualization
- **XLF: eXtensible Log Format**
  - general identifiers
    - user name, session ID, application ID, timestamp
  - PHP data
    - session variables
  - HTTP data
    - post variables, get variables
Example
IRMA Entities

- **Physical**
  - image data (images, global & local features)
  - tree data (hierarchical blob graphs)
  - source code (user-implemented programs)

- **Logical**
  - modules (programs, methods, networks, experiments)
  - input/output elements (image sets, feature sets)
  - control elements (jobs, workstations, cluster)
Different Axes to consider in designing a system for content-based access to medical images

System architecture has to account for:
- Fast processing → distributed system
- Component-based
- Connection to existing systems (e.g. PACS)
- User interface has to support flexible queries
- Ideally: use open standards as http, html, xml, ...

Conclusion