Data management for the 300-Area IFC

Tim Johnson, Roelof Versteeg, Yuxin Wu, Sera White and Carson Fenimore

Idaho National Laboratory

Data Management Team members

- Roelof Versteeg (INL monitoring group lead)
- Tim Johnson (postdoctoral associate, hydrogeophysical modeling and inversion)
- Yuxin Wu (postdoctoral associate, experimental geophysics (linking geophysics with geochemical processes))
- Sera White – IT lead
- Carson Fenimore – IT support, hardware/software integration

Outline

- Background
- Data management objective for the 300 Area IFC
- Data management Implementation
  - Overview of Data management elements and general INL approach
  - 300 Area IFC implementation
- Next steps

Background

- **Data management motivation**
  - **Data driven motivation**
  - **Result driven motivation**
- **Formal definition of data management**
- **Components of data management**

The typical single PI research effort

- Data in
  - Original field files (dumps from dataloggers)
  - Notebooks
  - Electronic format (EDD from laboratories, pdf reports, excel files)
- Data used for single objective (project, publication)
- Applications used for data processing are local and “owned” by PI
- Results of data (graphs, summary conclusions, data synthesis) possibly used multiple times
- After project, data storage generally unorganized
  - Folder on harddisk with all files
  - burn to cd/dvd
- Data distribution typically as flat data files or reports
- No institutional memory

The multi PI effort

- **Each PI "owns" his/her own data**
- **Each PI maintains his own data**
- **Each PI has his/her own applications**
- **Sharing primarily at result level**
- **Data reuse requires reinventing the wheel**
- **Data confidence and progeny unclear outside of PI owner**

Result generation

- Result generation typically requires multiple, local, disassociated software applications
  - Excel
  - Modflow
  - Stomp
  - Mineql
  - Surfer
  - Matlab
  - ….
- Processing steps are generally not stored/documented
- Results are often
  - Not auditable
  - Not transparent
  - Not reproducible

Consequences

- Data reuse is effectively impossible
- Collaboration efforts are tenuous
- Project management complicated
- Scientific value is diminished
- → motivates data management

Data management definition

- **Data management comprises all the disciplines related to managing data as a valuable resource.**
- **(one) definition (wikipedia) is *Data Resource Management is the development and execution of architectures, policies, practices and procedures that properly manage the full data lifecycle needs of an enterprise***

Data management components

- Includes
  - Data modeling (design of a relational database model which fits the data)
  - Database administration
  - Data warehousing
  - Data mining
  - Data qa/qc
  - Data security
  - Meta data management

IFC data management objectives

- 1 - Capture all data and metadata associated with the IFC effort, and provide data management function (QA/QC, warehousing, security)
- 2 - Provide a comprehensive, web accessible environment which provides IFC and non IFC scientists
  - Access to IFC related data and results
  - Access to the computational tools used to generate these results (including visualization tools)

Idaho National Laboratory

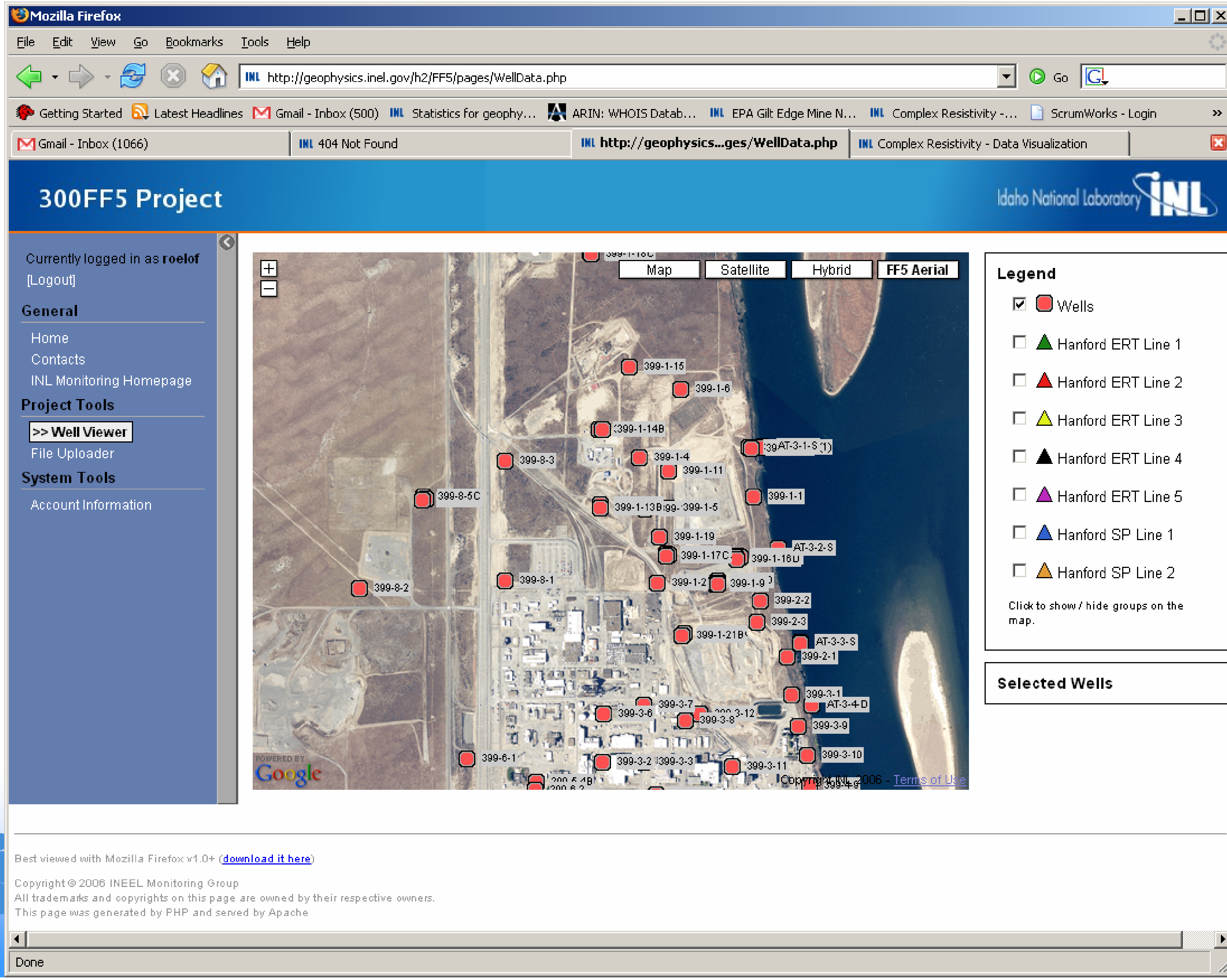
Note - 1

- IFC effort includes traditional data management component, but expands on it by including
  - Data capture effort
  - Web based Data access
  - Web based Tool access
- Approach driven by the fact that data is only part of the issue – other part is use of tools and data to generate information
- Novel IT development allows for implementation of this approach - parallels that by other US institutions (NOAA, NASA), as well as several international groups in Europe

Note - 2

- **Approach uses and implements existing tools and applications**
- **Approach uses well defined national and international standards**
- **Approach has been refined over last several years**
- **Approach provides a natural interface to GIS (Geographic Information System) technologies.**

Example 300 Area: use of Google Maps to show wells



Data management Implementation

- **Overview of Data management elements and general INL approach**

- **300 Area IFC implementation**

IT components

- Involves large number of acronyms, concepts and tools (XML, BPEL, Workflows, UDDI, webservices, relational databases, data normalization, C++, Javascript, Server/Client relations, transactions,…)
- Exact understanding not required – understanding of general concepts is beneficial

IFC data management effort: four <u>integrated</u> components

- **Data acquisition**
- **Data management**
- **Data processing and analysis**
- **Information access, distribution and use**

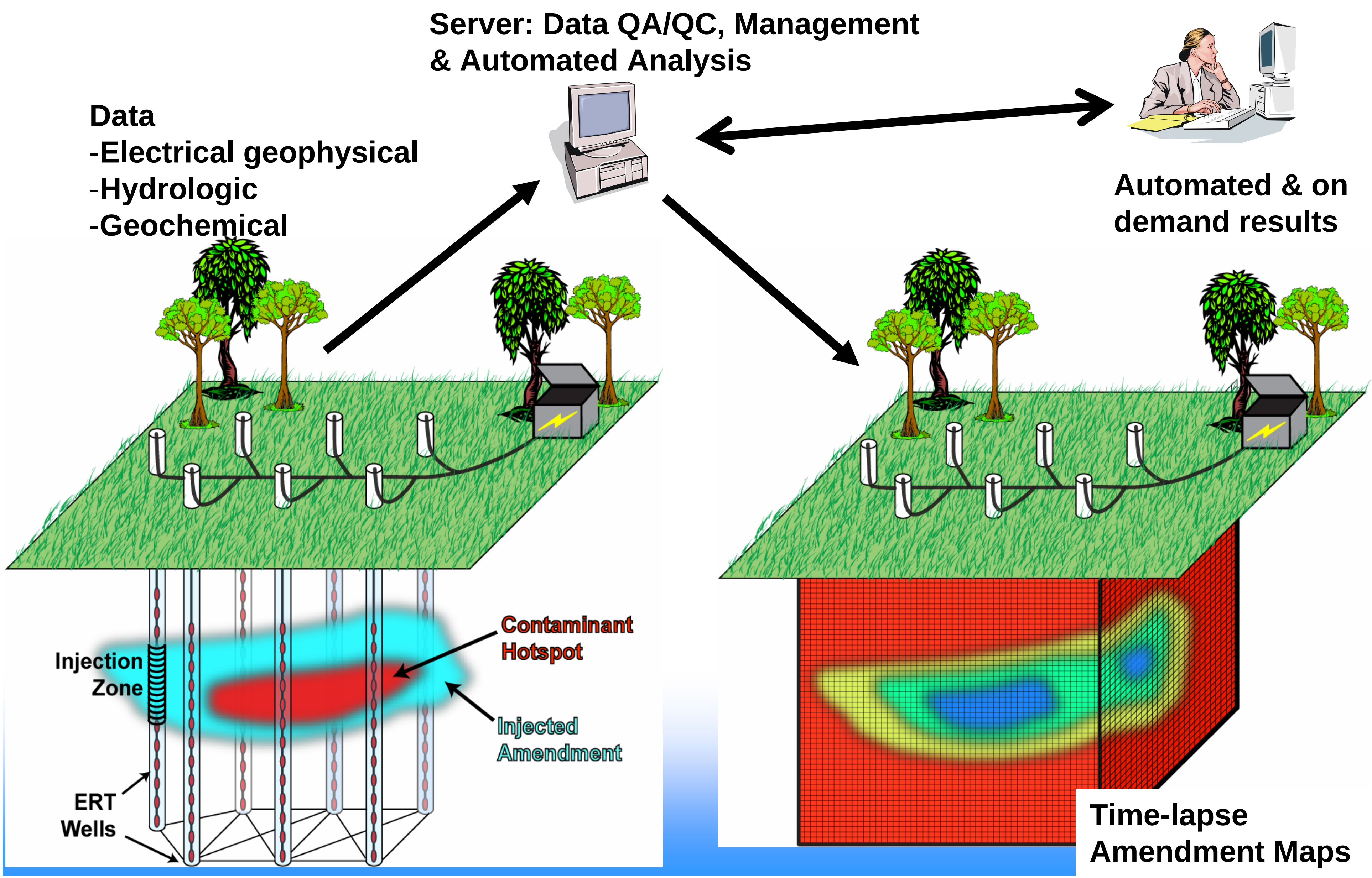
Data Acquisition
-Automated
-Manual

Data Management
-Storage of data
-Linking of data

Data Processing
-Internal (local)
-External (3rd party applications)

Information Access
-Web User Interfaces
-Automated Alerts
-Export to 3rd party software

Example: possible setup and data flow for 300 area amendment injection experiment

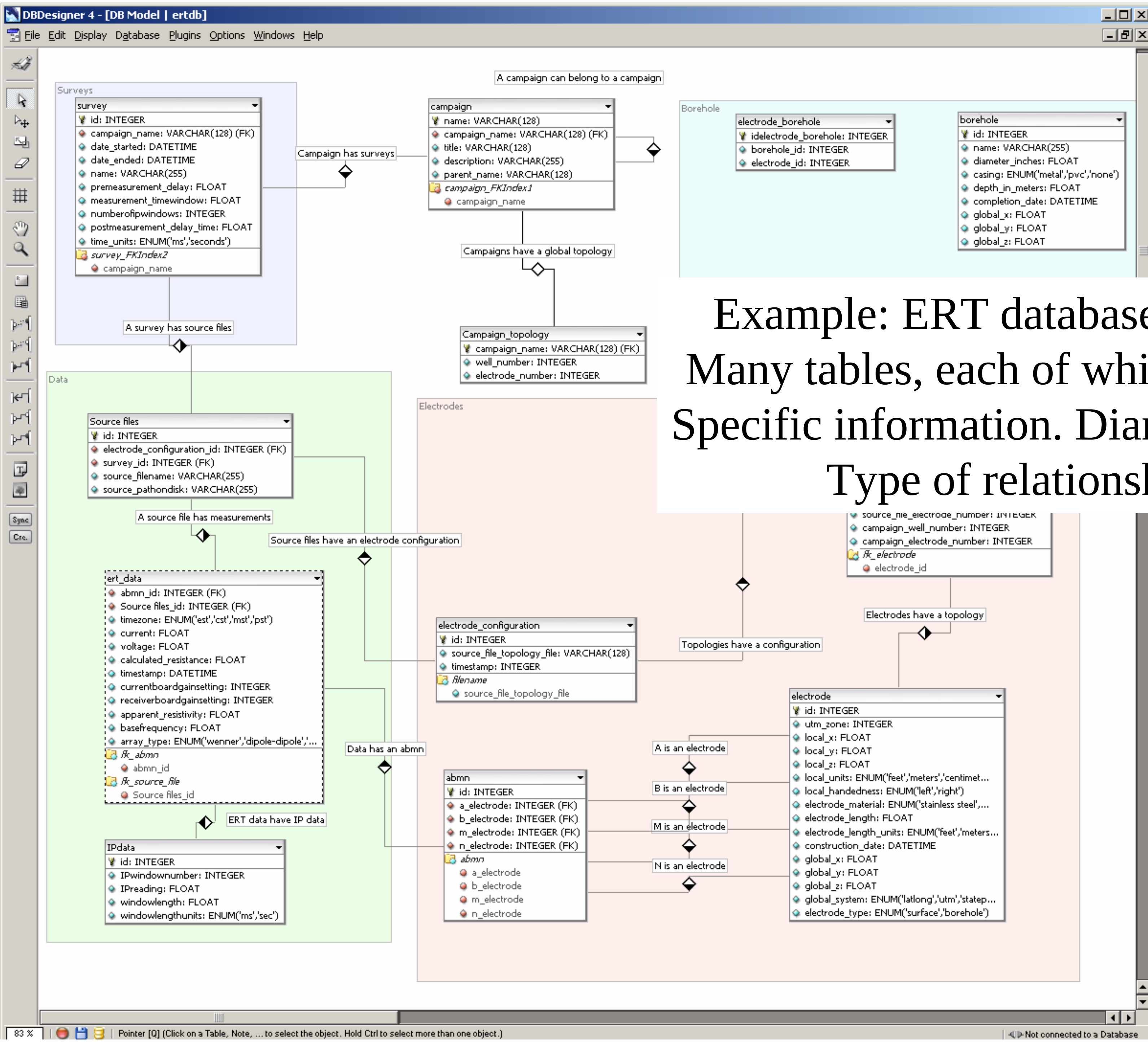


Effective Data management

- Requires
  - Well defined plan for capture for data (need to know what will be collected, by whom, when)
  - Use of relational databases for data storage
  - Collection of appropriate metadata
  - Plans in place BEFORE data is collected (from hard experience)

Relational databases

- Core part is relational model, and use of schemas
  - Schema: structure of how data is arranged
  - The fundamental assumption of the relational model is that all data are represented as mathematical *n*-ary relations (1 to 1, 1 to many, many to many)
- The relational model of data permits the database designer to create a consistent, logical representation of information.
- Includes a process of database normalization whereby a design with certain desirable properties can be selected from a set of logically equivalent alternatives.
- Data access and operation are handled by the DBMS engine, and are not reflected in the logical model.



Example: ERT database structure. Many tables, each of which contains Specific information. Diamonds show Type of relationships

Relational Databases : An Example

- Structure consists of linked tables
- Simple model would have two tables: well table, and sample table
- Well location would have
  - Well ID (unique number)
  - Well information (diameter, casing, possible screening depths)
  - Location (both Lat/Long and Washington State)
  - Construction information (completion date, driller id) [Note: this would link to a "driller table")]
- Sample table would have wellid, and sample information (sample date, sampling results)
- One well can have many samples
- One sample only collected in one well
- One to many relationship

Advantages of relational databases

- Only store information in one location
- All information is linked

Main Database task

- Database modeling: define an appropriate structure and relationship between all data
- Requires an in depth understanding of data and relationships
- Should be as comprehensive as possible (it is hard to go back and gather data later on (for instance sensor calibration information, environmental conditions, sensor number, …)

Automated data acquisition

- Automated – all data which does not involve manual intervention at any place during the acquisition process
  - Will include most geophysical data, pressure sensors, in well chemistry sensors, weather stations and so on.
  - Data is stored in well defined, fixed formats
  - Data can be transmitted automatically to server, or retrieved from data logger by dialup

Manual data acquisition

- Typically will include soil and water samples, and chemical, biological and soil analysis
- Requires combination of electronic sampling information (e.g. sampling plan and procedures) with sample number, and analytical results for good management
- Protocols for data scheduling and data management exist

Data processing

- **Collecting data is easy (and will get easier and cheaper)**
- **Managing and distributing data is harder**
- **Allowing other people to make effective use of your data is really hard**
- **Making use of OPD (Other People's Data) will have to become a way of life (requires confidence in data and collectors)**

- **→ <u>Core challenge in data processing</u>: How do we effectively process data and generate information – especially for distributed systems?**

Core challenge: information generation

- **Information generation from earth science data currently done through the sequential use of disjoint diverse applications by technical experts**
- **Information generation is typically a one way street– generating different views requires substantial manual efforts**
- **Example: generation of predictive model for typical DOE site is a customized, artisanal effort: documenting these models is hard because each model is "unique"**

Consequences

- **Hard and expensive for diverse users to generate new but similar results**
- **Poorly/Not auditable**
- **Little/No transparency**
- **No reproducibility**

- **<u>Workflows to the rescue!</u>**

Workflow definition

- ***The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules***
- **Scientific workflows: “*The automation of scientific data analysis according to a set of procedural rules*”.**
- **Workflow concept historically well known to most scientists, but typically within application (e.g. Excel macros, Seismic Unix and Promax scripts, Matlab .m files)**

Workflow examples in practice

- Timesheets
- Walmart ordering process
- Web purchases
- …..

- Key is
  - The existence of a process which can be formalized
  - Automation of this process

Scientific Workflows

- Historically developed within specific desktop application or specific computational environment (e.g. PNL Frames)
- Works well, but
  - Hard to share workflows (requires similar computational environment)
  - Hard to extend and distribute
  - Scientific workflows typically designed for high skill level users (different from business workflows)

- Following business workflows, evolving to workflows on the web (using webservices)

Web service

- **Web service:** ***a software component that is described via WSDL (Web Service Definition Language) and is capable of being accessed via standard network protocols such as but not limited to SOAP over HTTP***
- **Laymen terms: a web service is a self describing piece of software (which performs a specific action on well defined inputs and outputs) which can be invoked over the web using a standard calling protocol**
- **A web service has specific functionality. Underlying implementation is shielded from the user.**
- **A web service can be thought of as a subroutine or a function in traditional programming languages**

Specifics

- A web service is associated with a specific URI (Uniform Resource Identifier – similar to URL)
- Web service takes and returns arguments in standard formats (typically XML)
- A web service resides on a server
- A web service operates using well defined standards and protocols

Web Service Protocol Stack

- Service Transport: This layer is responsible for transporting messages between applications. Currently, this includes HTTP, SMTP, FTP, and newer protocols, such as Blocks Extensible Exchange Protocol
- XML Messaging: This layer is responsible for encoding messages in a common XML format so that messages can be understood at either end. Currently, this includes XML-RPC and SOAP.
- Service Description: This layer is responsible for describing the public interface to a specific Web service. Currently, service description is handled via the WSDL.
- Service Discovery: This layer is responsible for centralizing services into a common registry, and providing easy publish/find functionality. Currently, service discovery is handled via the UDDI.

Workflows on the web

- Wrap applications into a webservice
- Describe processing flow as a complex sequence of webservices
- Have workflow engine invoke webservices

- Some Advantages
  - Webservices are self describing
  - Webservices can "live" anywhere
  - Workflows can be self documenting
  - Exposure of application functionality can be tiered
  - User empowerment
  - Webservices can use existing applications – no need to reinvent the wheel

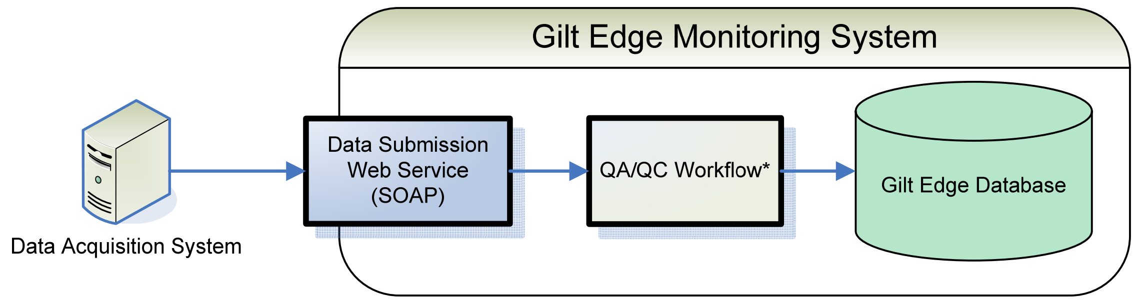
Note 1: Effort is standard based

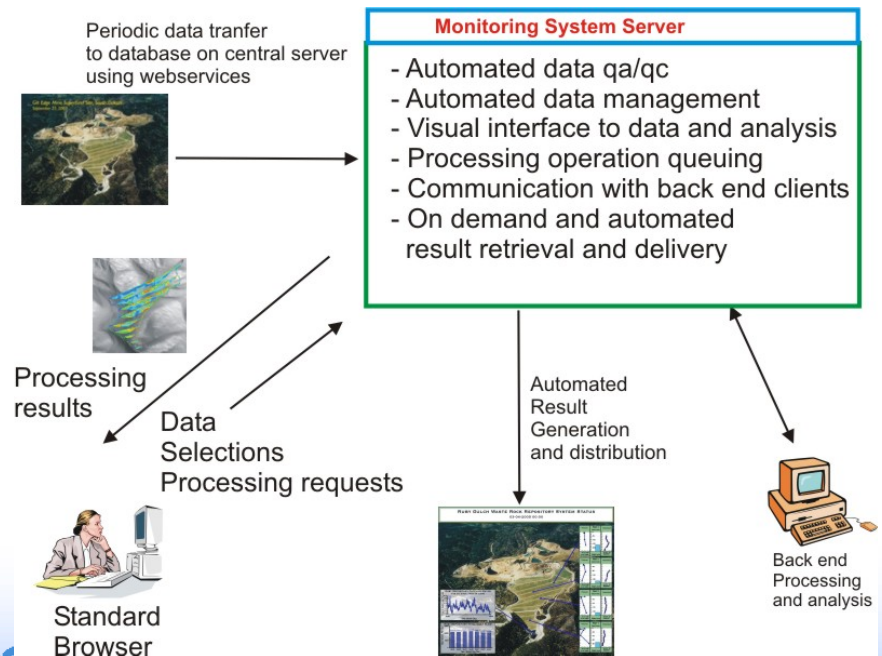
- **Approach is based on well published, well documented, industry wide standards**
- **Allows for easy integration with other efforts (e.g. Google, NASA, ESRI,…..)**
- **Makes for application which has built in longevity**

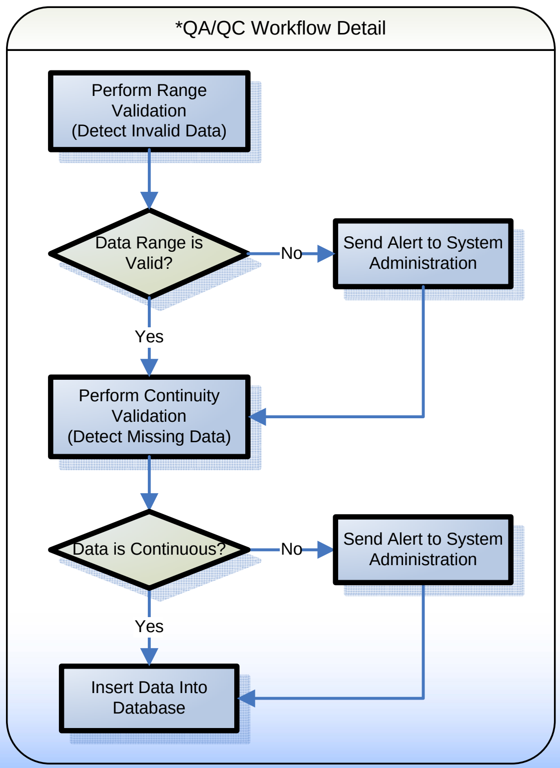
Note 2: Architecture is a SOA

- SOA stands for Service Oriented Architecture
- Operations are implemented as services
- Operations are loosely/lightly coupled
- Services can be accessed without knowledge of underlying implementation
- (think pizza delivery)

Technical implementation example







Approach description

- Current model couples
  - Web interface for workflow composition, configuration and invocation
  - Distributed web services providing functionality
  - Workflow engine performing execution
- Model uses industry standards for communications (XML/SOAP) , description (WSDL) and service discovery (UDDI)

Note: Some required elements for running workflows

- Workflow orchestration language (BPEL - Business Process Execution Language)
- Libraries of webservices and associated servers
- Yellow pages for webservices for data and applications (UDDI)
- Interfaces

Some other aspects

- Structure integrates seamlessly with Grid Computing/ASCR efforts
- Automatic collaborative research environment
- Implicit compatibility with open source model (not only what was done, but also specific configurations and underlying models)
- Focus on web service functionality (as opposed to implementation) allows for user transparent enhancements
- Provides long term structure for keeping track of data and results at little effort for original PI

300 Area IFC data management implementation

- General objectives
- High level technical description
- Year 1 Objectives
- Scope/actions in year 1

IFC data management objective

- 1 - Capture all data and metadata associated with the IFC effort, and provide data management function (QA/QC, warehousing, security)
- 2 - Provide a comprehensive, web accessible environment which provides IFC and non IFC scientists
  - Access to IFC related data and results
  - Access to the computational tools used to generate these results (including visualization tools)

IFC implementation – technical summary

- Service Oriented Architecture model for data access, data processing and result delivery
- Common components approach (reuse components developed by other groups)
- Utilize structures and standards developed in other fields
- Build on existing 300 Area efforts
- Adapt and refine existing INL model implementations to IFC needs (for instance, ability to integrate data and models)

Data management Implementation – Year 1

- Objective #1 is capture and store all IFC relevant data
- Done through initial (first 6 months) focus on
  - Data
  - Historic data inventory and collection
  - Basic web based collaborative environment implementation
- Should result in operational system in October 07
- Objective #2 is to understand tools.
  - Done through parallel inventory effort

Science implementation – data (1)

- Obtain from each IFC participant detailed information on all predicted types of data and metadata collected and needed by participant
- Discuss
  - sample planning/scheduling
  - Sample naming conventions
  - Sampling procedures
  - QA/QC rules (formal/informal)
  - Metadata collection
  - Sample analysis steps (analysis tools, calibration procedures, laboratory use)
  - Analysis results
  - Formats
  - Delivery mechanism
  - Current storage approaches

Science implementation – data (2)

- **Will result in a sampling type specific data model, as well as clear rules on how data are supposed to be collected, as well as qa/qc rules**
- **Will also result in proposed mechanisms for data transfer to central repository**
- **Will result in data models and relational database structures**
- **Will be tested with actual data**
- **Will result in a formal "IFC data management plan"**

- **<u>Will result in a basic web accessible system for data access in 10/07</u>**

Historic inventory: collect and assemble in one model all historic data

- Currently in hand
  - Well locations
  - Topography
  - Geophysical data
  - Data from EM monitoring effort
  - High resolution aerial topography
- Planned for integration:
  - Automated well data
  - River stage data
  - Weather data

Objective #2 – tool inventory

- Obtain from each IFC participant information on what they do with the data
- Software packages currently used + typical steps used in packages

Objective by October 2007: Basic collaborative environment

- Implement website (password protected) where users can
  - Access historic data (graph, zoom, download)
  - Access project data (if present)
  - Upload project data
  - Examine project documents (e.g. sampling protocols and plans)
  - Have access to wiki related to data management

Out year efforts

- Commodification of common tasks
  - Graphing
  - 3D visualization
  - Map display
  - Statistics
  - Linear algebra operations
- Models accessible through web interface
- Capture and central storage of user specific parameterizations

Next steps

- Understand IFC and non IFC 300 area data [March/April 2007]
- Project website setup [Early April 2007]
- Data modeling effort [April/May 2007]
- Formalization/Implementation/testing [May/July] of
  - Data acquisition protocols
  - Qa/qc rules
  - Data import mechanisms
- Integration of historic/existing data in one project website [April-July 2007]
- Tool inventory [March/June 2007]
- Start bringing in project data to system [July/October 2007]

Questions?