Provenance for seismology

Alessandro Spinuso
University of Edinburgh (UK)
KNMI (Royal Institute of Meteorology - NL)
Outline

Seismological Use cases

Workflow system and provenance aware components

Roles of Provenance

Harvesting Challenges and Strategy in DISPEL
Seismological Pipelines

Processing Chains are common practice in
**Seismic Interferometry** [G. D. Bensen et al. 2007]
**Forward Modelling** [A. Fichter et al. 2008]
Seismological Pipelines

Processing Chains are common practice in Seismic Interferometry [G. D. Bensen et al. 2007] and Forward Modelling [A. Fichter et al. 2008].
Seismological Pipelines

Processing Chains are common practice in
**Seismic Interferometry** [G. D. Bensen et al. 2007]  
**Forward Modelling** [A. Fichter et al. 2008]

Cross Correlation  
[Zaccarelli et al. 2011]
Seismological Pipelines

Processing Chains are common practice in **Seismic Interferometry** [G. D. Bensen et al. 2007]

**Forward Modelling** [A. Fichter et al. 2008]
Seismological Pipelines

Processing Chains are common practice in Seismic Interferometry [G. D. Bensen et al. 2007] and Forward Modelling [A. Fichter et al. 2008]...
Computational Requirements

Seismic Interferometry:

Data Intensive: High IO/CPU rate.

Data processing distribution is the main concern.

Distributed **Streaming model** suitable for in-memory and time-series data analysis.

Forward Modeling:

Execution of established simulation software (Seissol, Specfem3D) on **HPC** resources

Large number of synthetics waveforms (time-series) produced can be **streamed** and processed on Data Intensive Systems
Scientific Workflows:
Unified framework for the execution and validation of scientific applications over scalable and distributed resources

DISPEL (WF Language and Architecture):

Abstraction on the Data Management Layer (Retrieval, Intermediate Results)

Abstraction on the enactment and the Computational resources

Modular and Monitorable:

User Analysis Code in Reusable Processing Elements

Provenance model for Validation and Monitoring of the Results
DISPEL: Abstracts on Distributed Components and Resources

User Machine

Compute Cluster
DISPEL: Abstracts on Distributed Components and Resources
Integration of Enactment and User Defined Scripts

Reusable Processing Elements
Integration of Enactment and User Defined Scripts
Reusable Processing Elements

User Code Integration framework:

- Easy for scientists to implement, hides most of the WFMS complexity.
- Multi-platform and command line testing (JSON I/O protocol).

$ cat < inputfile > | python <yourscript> <VERCE JSON> <PARAMETERS JSON (defined accordingly to the PE parameters)> > <outputfile>

Automated and extensible metadata extraction (as provenance artifacts):

- Artifact’s Metadata in JSON
- Support for automatic annotations
# Integration of Enactment and User Defined Scripts

## Reusable Processing Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization</td>
<td>INGV</td>
</tr>
<tr>
<td>DataTrimming</td>
<td>LMU</td>
</tr>
<tr>
<td>SplitStream</td>
<td>INGV</td>
</tr>
<tr>
<td>FillGaps</td>
<td>INGV</td>
</tr>
<tr>
<td>TemporalNormalization</td>
<td>LMU</td>
</tr>
<tr>
<td>ResamplingINGV</td>
<td>INGV</td>
</tr>
<tr>
<td>ResamplingWHISPER</td>
<td>CNRS</td>
</tr>
<tr>
<td>Whitening</td>
<td>INGV</td>
</tr>
<tr>
<td>InstrumentCorrection</td>
<td>KNMI</td>
</tr>
<tr>
<td>Detrend</td>
<td>LMU</td>
</tr>
<tr>
<td>Filter</td>
<td>LMU</td>
</tr>
<tr>
<td>MapSeissolOutputToStations</td>
<td>LMU</td>
</tr>
<tr>
<td>MisfitCalculation</td>
<td>LMU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreamToSeedFile</td>
<td>KNMI</td>
</tr>
<tr>
<td>CorrelateNoiseLMU</td>
<td>LMU</td>
</tr>
<tr>
<td>CorrelateNoiseWHISPER</td>
<td>CNRS</td>
</tr>
<tr>
<td>CorrelateNoiseINGV</td>
<td>INGV</td>
</tr>
<tr>
<td>WaveformPlot</td>
<td>INGV</td>
</tr>
<tr>
<td>Spectrogram</td>
<td>LMU</td>
</tr>
<tr>
<td>ArrayStacking</td>
<td>LMU</td>
</tr>
<tr>
<td>StackPlot</td>
<td>LMU</td>
</tr>
<tr>
<td>MonetDBFileStream</td>
<td>KNMI</td>
</tr>
<tr>
<td>ConvertSeissolOutput</td>
<td>LMU</td>
</tr>
<tr>
<td>Specfem3d2Stream</td>
<td>INGV</td>
</tr>
</tbody>
</table>

---

**Thursday, 20 June 2013**
Roles of Provenance

Provenance can describe the analysis at various granularities (Resources, WF Configuration, Lineage):

Provide intermediate data visualisation/access for validation at runtime

Process and store heterogeneous/user-defined metadata, processing parameters and errors

Capture and describe data interdependencies

Aware of Mutations in Software (PE), Machines and Data Sources (Registry)
Roles of Provenance

Preprocessing

Validation

Partial Cross Correlation

Validation

Full Run

Provenance

Intermediate data

Thursday, 20 June 2013
In Progress: Custom access within dedicated Science Gateways accessing external datasources
In Progress: Custom access within dedicated Science Gateways accessing external datasources
In Progress: Custom access within dedicated Science Gateways accessing external datasources
In Progress: Custom access within dedicated Science Gateways accessing external datasources

Thursday, 20 June 2013
In Progress: Custom access within dedicated Science Gateways accessing external datasources
Compound Model based on W3C PROV

Concepts Distribution

- Lineage / Retrospective Prov
- DISPEL/Registry Prospective Prov
- Enactment / Environmental Optimisation

Thursday, 20 June 2013
Usage Scenarios

1 - Components
Behaviours: Internal transformations within a group of PEs.

2 - Data transformation snap-shots: Runtime view on the variation of a portion of the data stream

3 - Comparing Runs:
Comparing 1 & 2 over multiple runs

4 - Smart Reruns and Fault-tolerance:
Checkpoints could trigger the storage of the data
Usage Scenarios

1 - Components
Behaviours: Internal transformations within a group of PEs.

2 - Data transformation snap-shots: Runtime view on the variation of a portion of the data stream

3 - Comparing Runs:
Comparing 1 & 2 over multiple runs

4 - Smart Reruns and Fault-tolerance:
Checkpoints could trigger the storage of the data
DISPEL functions and Pipeline’s <SeismoStages>

Provenance Aware DISPEL Functions

Provenance

Seismo PE

Parameters and sub-pipelines

DISPEL Function PE that performs selective provenance/storage/visualisation of the processed datastream
DISPEL functions and Pipeline’s *<SeismoStages>*

**Provenance Aware DISPEL Functions**

**Processing Stage: CompositePE**

- **Provenance**
- **Visualize**
- **Store**

**Data Flow**

**DISPEL Function PE** that performs selective provenance/storage/visualisation of the processed datastream.

Thursday, 20 June 2013
DISPEL functions and Pipeline’s \textit{<SeismoStages>}

DISPEL Function PE that performs selective provenance/storage/visualisation of the processed datastream
Harvesting Challenges

- **Preparatory phases**: The evaluation of various processing strategies requires multiple runs adopting different components and configurations.

- Full coverage of retrospective provenance could present the following numbers:

  **Cross Correlation, preparatory phase, single run**:

  - **Preprocessing**: 20 stations, one year of data, 1 hour chunks size, 6 preprocessing stages = 1.051.200 artifacts
  
  - **Cross correlation**: \((20(20-1))/2\) x-correlation pairs plus data stacking = 3.328.800 artifacts.

  - **We are looking at a selective recoding strategies.** Decrease I/O rate and the storage of not interesting metadata.
Harvesting Challenges

Centralised vs Distributed Storage:

- Computation can be distributed across different models and institutional infrastructures.
- Computation can happen in locations with no persistent storage facilities, while provenance data has to be preserved.
Harvesting Challenges

Different enactment engines

Not all of them support provenance (OGSA-DAI, Twitter STORM, Apache S4)

Strategies in Evaluation:

In WF, programmatic provenance

DISPEL functions returning configurable and provenance aware CompositePEs

**Provenance PEs vs semantic annotations on functions**
DISPEL functions and Pipeline's `<SeismoStages>`

```java
PE<SeismoStage> plot(String plotLocation,
                      String provenanceResource,
                      String processingResource,
                      Boolean traceit) {

    /* Echo PEs are used to pass the lineage */
    /* data to the provenance PE if traceit is true */
    /* or to the next Stage if traceit is false */

    Echo echoin = new Echo;
    Echo echout = new Echo;
    ProvenancePE prov = new ProvenancePE;
    WaveformPlot plot = new WaveformPlot;

    /* Provenance recording enabled */
    if (traceit) {
        |-provenanceResource-| => prov.resource;
        plot.metadata => prov.metastring;
        echoin.output => prov.lineagein;
        prov.lineageout => echout.input;

        /* Provenance recording disabled */
    } else {
        plot.metadata => discard;
        echout.output => echoout.input;
    }

    |-processingResource-| => plot.resource;
    |-repeat enough of "filepath"+plotLocation-| => plot.parameters;

    /* Returns the composite PE */
    return SeismoStage(
        <Connection input=plot.input;
        Connection lineagein=echoin.input>
        =>
        <Connection lineageout=echout.output;
        Connection output=plot.output>);
}
```

DISPEL Function returning a `<SeismoStage>` PE that performs the plotting of the data stream. Provenance can be disabled by making `traceit = False`. 
Provenance Model for Streaming Systems

Keeping the Model implementation agnostic to the WFS internal codebase (general specs)

Stateful scenario: Capturing and storing relevant state and data dependencies for a stateful process $Q$ at step $i$. 

Thursday, 20 June 2013
In progress: Functional representation and lineage access via HOPM [Buneman, Cheney, Kostylev, 2012]

Introduction of HOPM like approach (Hierarchical Model of Provenance) for DISPEL

Nested hierarchy of function calls.

Expandable where the function is offering provenance recordings (P).

Users can define Checkpoints to activate provenance harvesting.
In progress: Functional representation and lineage access via HOPM [Buneman, Cheney, Kostylev, 2012]

Introduction of HOPM like approach (Hierarchical Model of Provenance) for DISPEL

Nested hierarchy of function calls.

Expandable where the function is offering provenance recordings \(P\).

Users can define **Checkpoints** to activate provenance harvesting.

\[\begin{align*}
\text{Let } Filt(x) &= \text{Vis}(C(x,\text{par})) \\
\text{Norm}(x) &= \text{Vis}(C'(x,\text{par}')) \\
\text{Vis}(x) &= C''(x,\text{par}'') \\
\text{In } \text{map}_{\text{Norm}}(\text{map}_{\text{Filt}}[S1,S2,...])
\end{align*}\]
Thank you!