The Electronic Tool Integration Platform

Bernhard Steffen
University of Dortmund

Volker Braun
Tiziana Margaria
Roadmap

• Motivation
• Goals
• The ToolZone Software
• The ETI Meta Model
• Executing Coordination Sequences
• Conclusion
Motivation

Tools

1. Searching for candidate tools
2. Getting and installing the tools
3. Comparing the tools

End Users

adequate communication channel is missing
The **ETI** Project is intended to support people who want to set up a **software tool experimentation site** by providing

- a Web-based, open platform for the interactive experimentation with and coordination of heterogeneous software tools
  
  accessible by the **ToolZone** Software

- an infrastructure organizing the platform development, extension and ETI site hosting

  the **ETI Community Online Service**
  at [www.eti-service.org](http://www.eti-service.org)
Goals

The ETI Sites:

• application-domain-specific instantiation of the platform

• tool functionalities are located in the tool repository

• tool providers can publish their tools

• end users can experiment online with the tools and heterogeneous combinations of tool functionalities

• access to common case studies as well as private data space
The ToolZone Software

Tool Access

- **structured access** to the tool functionalities located in the tool repository with the ability to
  - get **detailed information** on each available tool feature
  - **execute single** tool features
  - **combine** heterogeneous tool functionalities **to programs**
  - **run the programs** via the Internet

**tool coordination** and **Internet-based execution facilities** are the conceptual key features
The ToolZone Software

Tool Coordination

- **full coordination** facility by means of the HLL
  - access to every available tool feature
  - mainly for experienced users

- automated **coordination support** by ETI’s **synthesis component**
  - tool sequences are generated out of **goal-oriented** abstract
    - descriptions specifying **what** should be done instead of **how**
    - designed for unexperienced users
    - access to specifically structured functionalities
The ETI Meta Model

Activities

• ETI-specific component model models a tool feature as “transformational” entity

\[ T_1 \rightarrow T_2 \]

• specified by two aspects:
  - interface aspect
  - functional aspect

• can be executed in **stand-alone** or **tool-coordination** mode
## The ETI Meta Model

### Simple Text Processing Activities

<table>
<thead>
<tr>
<th>Activity Name</th>
<th>Input Type</th>
<th>Output Type</th>
<th>OS Cmd.</th>
</tr>
</thead>
<tbody>
<tr>
<td>latex</td>
<td>TEXFile</td>
<td>DVIFile</td>
<td>latex</td>
</tr>
<tr>
<td>dvips</td>
<td>DVIFile</td>
<td>PSFile</td>
<td>dvips</td>
</tr>
<tr>
<td>gv</td>
<td>PSFile</td>
<td>Display</td>
<td>gv</td>
</tr>
</tbody>
</table>
The ETI Meta Model

Taxonomies

- **classification** of types and activities
- represented as directed acyclic graphs
  - leaves represent **atomic entities**
  - intermediate nodes denote the **set of reachable entities**
  - edges model "is-a" relation between their source and target nodes
The ETI Meta Model

A Simple Type Taxonomy Example
GUI Impressions: Taxonomies

Activity Group tt

Description:

This is the root of the ETI activity taxonomy graph. This group represents all goup and atomic activities within the ETI activity taxonomy.

Please send suggestions and corrections to eti-team@eti-service.org

Last modified on 08-Mar-2001, 16:19
The ETI Meta Model

Coordination Sequences

- **sequential programs** built on the basis of the activities
  \[ T_1 \xrightarrow{a_1} T_2 \xrightarrow{a_2} T_3 \rightarrow \cdots \rightarrow T_{n-1} \xrightarrow{a_n} T_{n-1} \]

- can be **executed via the Internet** using the ToolZone software

- simple **example**:
  
  \[
  \text{TEXFile} \xrightarrow{\text{latex}} \text{DVIFile} \xrightarrow{\text{dvips}} \text{PSFile} \xrightarrow{\text{gv}} \text{Display}
  \]
Loose Specifications

• building coordination sequences manually may be a non-trivial task

• synthesis component generates coordination sequences out of loose descriptions

• two orthogonal dimensions of looseness
  – local looseness
  – global/temporal looseness
The ETI Meta Model

Loose Specification: Example

- "TEXFile < Output" to query all coordination sequences being able to display a TEXFile on an Output device

- local looseness: loose specification of the output device by using the type group Output instead of a concrete type, like Printer or Display

- global looseness: by using the before operator "<"
Generating Coordination Sequences

- ETI’s synthesis component generates a **synthesis solution graph**
- based on a **coordination universe**
- **synthesis strategies** influence the resulting solution set
  - all solutions
  - all **minimal** solutions
  - all **shortest** solutions
  - one **shortest** solution
The Example

• The Alternating Bit

• Visual Verification

• Model Checking
Synthesizing Coordination Graphs
Executing Coordination Sequences
The LOTOS Specification
GUI Impressions: Graph Systems
The Minimized Automaton
**Program Execution**

- From the initial state, there is a strict alternation between 'PUT' and 'GET' actions, starting with a 'PUT'. Moreover, after every 'PUT' (resp. 'GET') a 'GET' (resp. 'PUT') is always potentially reachable.

```plaintext
nu ExpectPUT .
  < true -> "PUT" > true
data
  [ "PUT" ] nu ExpectGET .
  < true -> "GET" > true
data
  [ "PUT" ] false
data
  [ "GET" ] ExpectPUT

and
```

Bernhard Steffen, University of Dortmund

18.9.2001
Model Checking
The ETI Community Service

This site organizes the effort to enhance the Electronic Tool Integration (ETI) platform, an environment designed for the interactive experimentation with and coordination of heterogeneous software tools via the Web (see Goals section for more details).

The ETI online community functions as an informal meeting point for the exchange of opinions; end users, tool providers, tool integrators, platform developers and ETI site managers are invited to comment, to query, and to provide feedback on the platform.

The focus of this site is the general platform which can be instantiated using tools from a chosen application domain. This application specific instantiation is typically done by an ETI Site. ETI users can access the software tools provided by an ETI site using the ToolZone client.

If you have further questions on the ETI platform, subscribe to one of the mailing lists or feel free to contact a member of the ETI coordination team.

The ETI project is proudly sponsored by

METAFrame
The Personalized Online Service House

Sun Microsystems
Conclusion

- a Web-based, open platform
- interactive experimentation with heterogeneous tools
- elaborate coordination support
- coordination programs can be run via the Net
Conclusion

- The ETI Community Service: support for people, who want to set up a Web-based communication channel between tool providers and end user. See: www.eti-service.org

- a concrete instance: International Journal on Software Tools for Technology Transfer (STTT)

- We are in the course of setting up Network of Excellence
The Electronic Tool Integration Platform
Logical Layers

Client Layer

ToolZone Client

Internet Access Layer

Internet Access Server

Feature Layer

Tool Management Application

Persistency Layer

Tool Repository
Software Architecture

Client Layer

Internet Access Layer

Feature Layer

Persistency Layer

Client Host
ToolZone Client
HTTP RMI

ToolZone Host
ToolZone Server
HTTP Server

Application Server
Tool Management Application
Local Tools

Application Host

Remote Tool Host
Remote Tools

CORBA RMI

Database Host
File Server
DBMS

Bernhard Steffen, University of Dortmund
The Tool Management Application

- fully controls the access to the tool functionalities located in the tool repository
- based on the METAFrame environment
- application-independent kernel managing the application-specific components (here tools features)
  - the kernel provides:
    - the hypertext system
    - the synthesis component
    - the HLL interpreter
- central for tool integration
The High-Level Language

- **procedural** programming language
- can *dynamically be extended* by new basic types and procedures
- types and procedures are provided by **METAFrame Modules**
Tasks for Tool Integration

- **split up the tool** to be integrated into set of activities
- **classify** activities and types within ETI’s taxonomies
- **implement METAFrame Modules** which HLL-enable the chosen tool functionalities and types
- **write HLL program fragments** specifying the operational view of the activities