Audio n-Genie

Domain Specific Language for Audio Processing

Presenter: Tiziano Leidi, ICIMSI, Switzerland
Audio n-Genie’s project started in 2002 around requirements for tools to speed-up development of audio applications.

Main requirement: reduce the impact of recurrent tasks in the development process.

In 2006:
Project Partners

• **ICIMSI**
  – CIM Institute for Applied Computer Science and Industrial Technology, specialized in automated design technologies
  – Located in the technology park of Lugano-Manno
  – Founded in 1992

• **Anagram Technologies SA**
  – Swiss SME specialized in audio and audio design technologies
  – Located in Preverenges, Switzerland
  – Founded in 2000
  – Customers: IC manufacturers, consumer electronics companies

• **KTI / CTI**
  – Swiss Federal Innovation Promotion Agency
  – Provides funding for collaboration with Universities or Technical Institutes
System Designer

- 5 applied research projects from 2002 to 2007
- 2 applied research project started in 2007 and 2008
- Development team ranging from 2 to 4 people
Competences

Tools for Rapid Prototyping, Development and Configuration of Domain Specific Software.

Dedicated development tools for software factories through domain specific languages.

- Component based, model driven, aspect oriented generative Programming.
- Visual editors and assisting design functionalities.
- Support for automatic structural optimizations.
- Integrated compilation, debugging and simulation.
- Support for integration of frameworks, APIs and pre-existing code.
- Built on Eclipse (www.eclipse.org)
Problem and Solution

Software development is an incremental process difficult to predict.

Rapid Prototyping

- Design by combining reusable components
  - Tests prototypes and request stakeholder feedback
  - Generate design Prototypes
  - subsequent design phase

Tuning and Customization

- Optimize and customize by managing specific aspects
  - Tests pre-releases and request stakeholder assessment
  - Generate pre-releases for optimization and debugging
  - subsequent optimization or customization phase

- Produce optimized final product
- Release software

Software development is an incremental process difficult to predict.
Development Approach

Tool based on **visual domain specific language** and:

**Component Based**
- different pre-existing parts of code are treated as black boxes and joint together to compose software applications.

**Model Driven**
- technique that focuses development around a high-level, abstract, often iterative data structure: the model.

**Generation Driven**
- software development approach focused on the exploitation of automatic processes for the production of source code’s replicable parts.

**Aspect Oriented**
- programming paradigm converging towards separation of concerns during software design and development.

**Pattern Based**
- a structural characteristic of software that may be replicated to provide solutions for a recurring problem in software design.
Structure of the Tool

GUI

Assisting Functionalities
- Model validation
- Model refactoring
- Model transformations

Model
- Components abstraction
- Patterns abstraction
- Aspects abstraction

Source Code Generators

Generated Applications
Components - Patterns - Aspects
Structure of the Tool

Audio n-Genie
EMF  GMF  GEF  CDT
Eclipse Platform
JAVA - JRE

User Interfaces
Processing Engines

Standard development tool functionality
• CVS / Subversion
• Search / Navigation Support
• C and C++ editing / compile / run
• ...

Standard development tool functionality
• CVS / Subversion
• Search / Navigation Support
• C and C++ editing / compile / run
• ...
Model Abstraction

- **Unit**
  - SampleRate
  - BufferSize
  - LatencyType
  - SampleLatency
  - Stop
  - Left
  - Right

- **Composite**
  - MicroReader
    - SampleRate
    - BufferSize
    - LatencyType
    - SampleLatency
    - Stop
    - Left
    - Right

- **Configuration**
Processing Phases

- **Startup**
  - PreProcess
  - PrepareParams
    - Process
      - FinalizeParams
        - PostProcess
          - Shutdown
            - Par Gates
            - Data Gates
              - Execution Loop
              - Block

- Par Gates
- Data Gates
- Par Gates
- Par Gates
Integration Mechanisms
User Interfaces
Further Features

- The Model persistence is human readable (XMI).
- Generated code is readable and easy to understand.
- Both Bottom-up & Top-down design approaches are possible.
- Extraction mechanisms favor incremental development.
- Generated code exists side by side with model and is compiled and debugged in the same environment.
Sample Applications
Sample Applications
Harmonizer
Audio n-Genie is open source (EPL)

www.systemdesigner.ch
www.anagramtech.com/audio-n-genie/
tiziano.leidi@icimsi.ch