

# Notio

---

A Java API For Developing CG Tools

Finnegan Southey  
University of Waterloo  
Department of Computer Science

James G. Linders  
University of Guelph  
Department of Computing and Information Science



# Overview

---

Introduction

P Motivation

P Design Goals and Decisions

P Features

P Implementation

P Applications

P Future Directions

P Conclusion

---



# The Need For CG Tools

---

## Motivation

- P Conceptual graphs form an active area of research with many frontiers unexplored.
- P Conceptual graphs have established a foothold in real-world applications.
- P Tools are needed to construct, store, manage, and manipulate CG's.
- P The variety of applications requires a wide range of tools.



# Existing CG Tools

---

## Motivation

- P Several quality CG tools exist today.
- P Each offers different capabilities and interfaces for working with CG's.
- P Most are as much a research exercise as they are a usable tool.
- P While each tool offers different capabilities, there is much common ground amongst them.

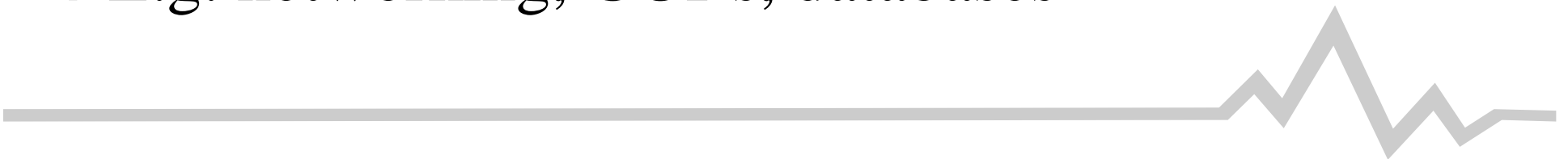


# API's: The Need For A Lower Level

---

## Motivation

- P Given the duplication of effort and the need for widely varying tools it is desirable that a common platform for building such tools should be developed.
- P In most software application areas, such platforms consists of *application programming interfaces* (API's).
- P E.g. networking, GUI's, databases



# Notio: A Java API For CG's

---

## Design Goals and Decisions

- P Notio is a Java-based API for developing CG tools.
- P The Notio API defines a set of classes with their associated methods.
- P These classes provide the structures and operations necessary to build and manipulate CG's.



# Design Goals For Notio

---

## Design Goals and Decisions

- P Easy application development.
  - P Intuitive relationship to familiar CG abstractions.
  - P API independence from underlying implementation.
  - P Portability.
  - P Extensibility.
  - P Generality (by minimizing assumptions about applications).
  - P Flexibility (by allowing varying levels of conformance).
  - P Robust.
- 



# Why Java?

---

## Design Goals and Decisions

- P Object-oriented programming has become an industry standard and Java has achieved rapid and wide-acceptance amongst major software developers.
- P Java is robust (garbage collection and exceptions), feature-rich, and platform independent.
- P Native code and other languages can be employed via the Java Native Interface (JNI).
- P Free and commercial development tools are widely available.
- P Java offers interoperability with CORBA and databases (via JDBC), extensive GUI capabilities, networking (including RMI and Internet protocols), documentation generation (Javadoc), and a component architecture (JavaBeans).





# Three Layers Of A Notio System

---

## Design Goals and Decisions

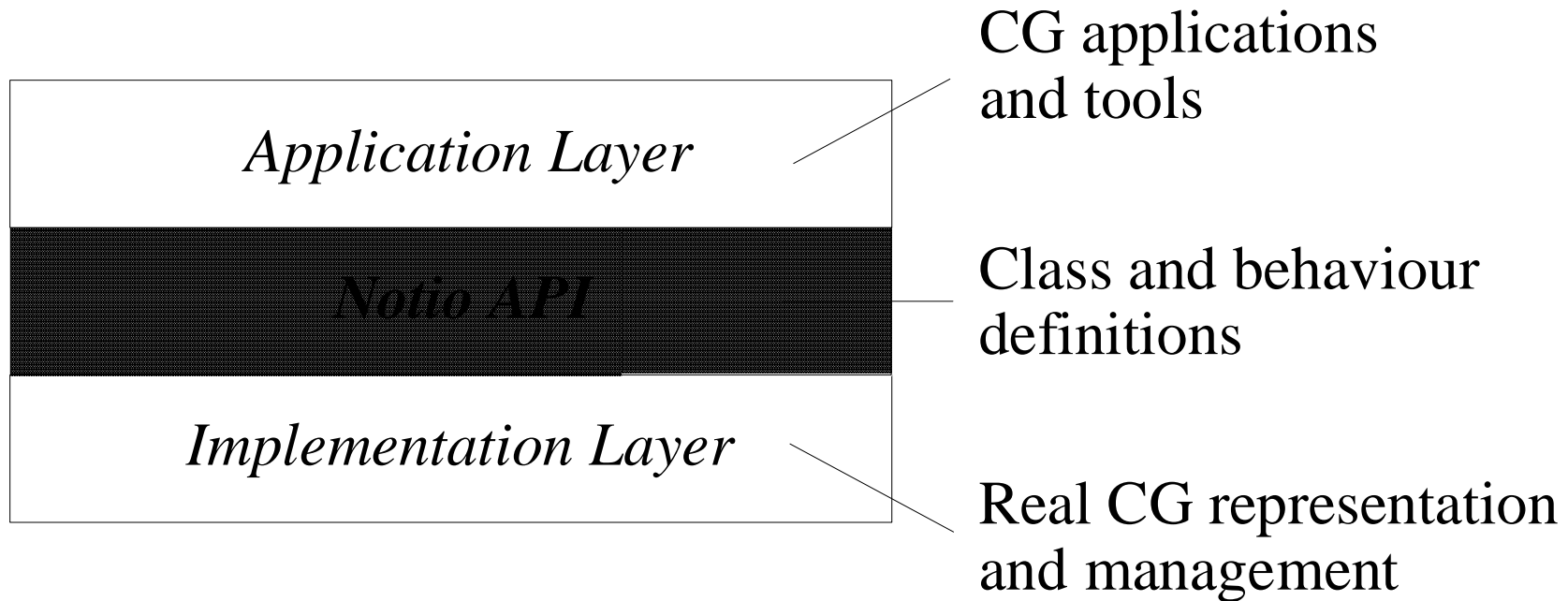
- P Applications written using the Notio API are considered to form an *Application Layer*.
- P The *Notio API Specification* describes the classes and methods an Application Layer may use.
- P An actual implementation of the specification is known as an *Implementation Layer* and provides the representation and operations.



# Structure of a Notio-based System

---

## Design Goals and Decisions



# General Features Of The API

---

## Features

- P Close conformance with the ANSI standard.
  - P Classes and methods for constructing, modifying, and performing operations on simple and compound CG's and type hierarchies.
  - P Detailed exception-based error reporting and handling.
  - P An interface for pluggable parsers and generators.
  - P A formal framework for providing extensions.
  - P Areas allowing variable compliance and optional constraints.
- 



# Structure Classes

---

## Features

### P Graph

- ▶ Concept
  - ConceptType
    - ConceptTypeDefinition
  - Referent
    - Quantifier
    - Designator
      - Descriptor, Literal, Name, Marker
- ▶ Relation
  - RelationType
    - RelationTypeDefinition
  - Actor

### P Concept/RelationTypeHierarchy

---



# Operations

---

## Features

- P Easy traversal of compound graphs.
- P Highly flexible graph matching through “matching schemes”.
- P Standard canonical operations: copy, join, restrict, simplify



# The Reference Implementation

---

## Implementation

- P The *Notio Reference Implementation* is a pure Java implementation of the Notio API Specification using JDK 1.1.
  - P Provides almost all of the functionality described by the current specification.
  - P Includes a CGIF parser and generator and an experimental LF parser and generator.
  - P Extremely flexible graph matching facilities.
-

# Availability

---

## Implementation

- P The Notio Reference Implementation is available under the GNU General Library Public License.
- P Source and complete Javadoc-generated documentation are available via the WWW.
  - ▶ <http://dorian.cis.uoguelph.ca/CG/projects/notio>
- P Testing suites exist to help maintain the stability between versions.
- P The API is not yet frozen so the implementation represents the latest API.



# Our own Work With Noto

---

Applications

Project of Noto: The Noto Group

Ossa: A productive system base concept  
modelling visual realities.

CGBase: A simple, experimental  
graphics language.

CG Abstractor: A database and

.

---





# Other Existing Projects With Notio

---

## Applications

**P** Logicon AG

**P** KmWiki (a clone of WikiWikiWeb)

**P** A web-server providing CG-based graphical management and retrieval of collaborative documents.



# Proposed Corporate Work With Notio

---

## Applications

P ALCATEL Corporate Research Center

P Considering Notio for use as part of a natural-language interface to directories.

P A Java-based CG API may provide significant advantages in interfacing with existing database systems.



# What's next?

---

## Future Directions

- P Further refinement and freezing of the core API.
- P Extending the API to collections of CG's.
- P A pluggable framework for actors.
- P More translators (e.g. KIF).
- P New implementation layers based on existing CG and knowledge base tools (e.g. OKBC, KQML).
- P Figura: A Javabeau CG editor/viewer component.



# A Final Note

---

## Conclusion

- P The current Notio API and reference implementation have been used successfully in several projects by people with variable knowledge of CG's and Java.
  - P We encourage people to examine the API and try it out. Comments, suggestions, and contributions are very welcome.
  - P We hope that Notio will prove a valuable part of the CG research and development community.
-