
CommunityExplorer

A Framework for

Visualizing Collaboration Networks

Leonel Merino
Mohammad Ghafari
Oscar Nierstrasz

Software Composition Group
University of Bern

CommunityExplorer

A Framework for

Visualizing Collaboration Networks

Leonel Merino
Mohammad Ghafari
Oscar Nierstrasz

Software Composition Group
University of Bern

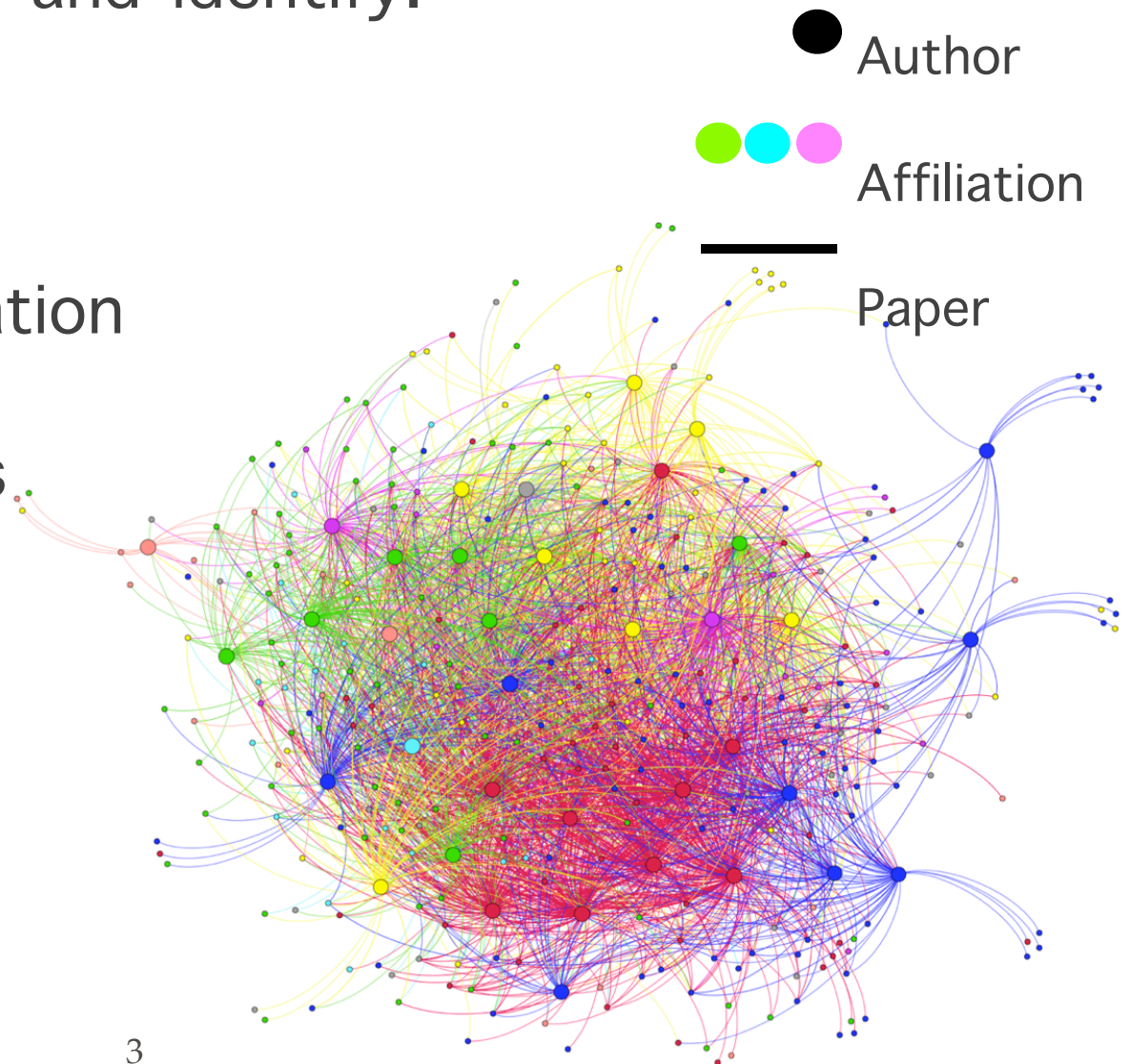
Visualization of Collaboration

Support users to analyze and identify:

RQ1) Groups of collaboration

RQ2) Evolution of groups

RQ3) New & key authors



Visualization of Collaboration

Support users to analyze and identify:

RQ1) Groups of collaboration

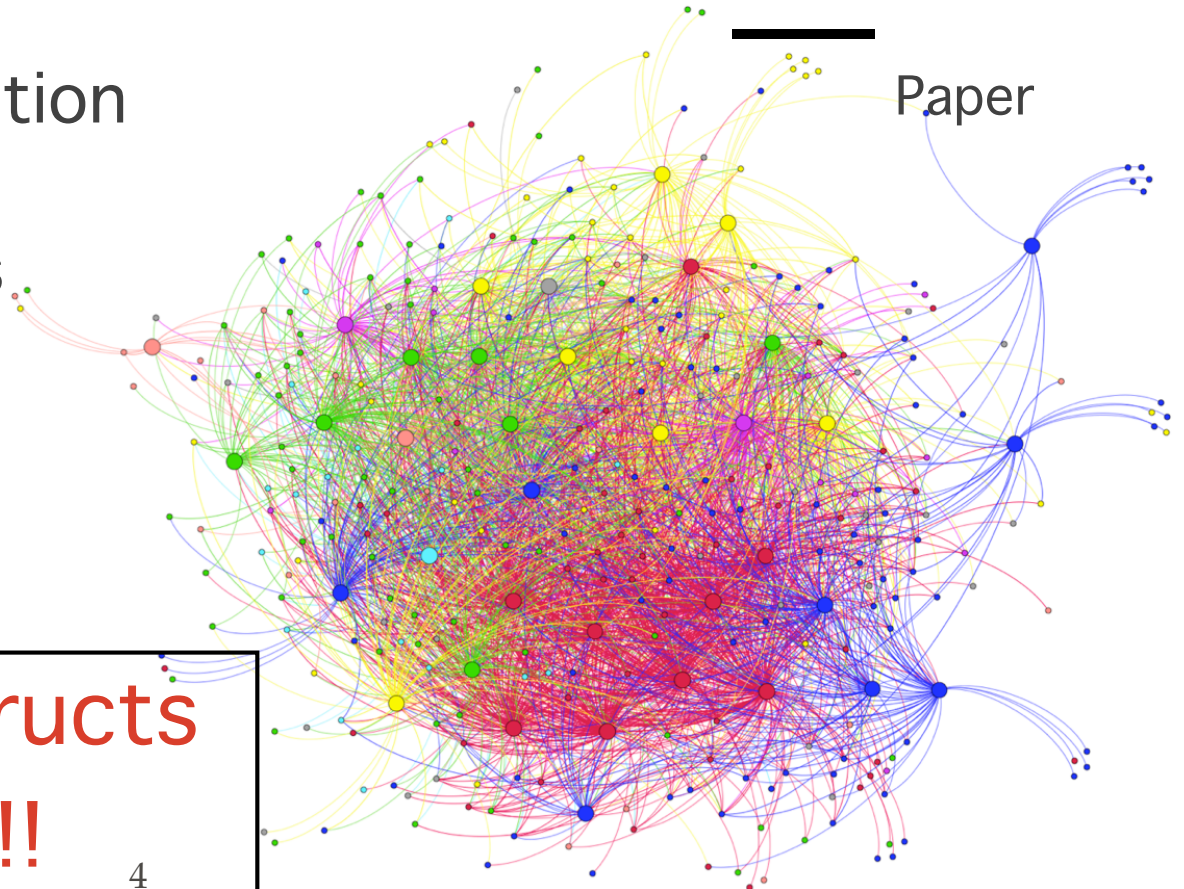
RQ2) Evolution of groups

RQ3) New & key authors

● Author

●●● Affiliation

— Paper



Edge crossing obstructs
comprehension !!

Collaboration in Scientific Communities

IWST and VISSOFT



Collaboration in Scientific Communities

Papers and Authors

IWST and VISSOFT



Deriving an Object Model from Legacy Fortran Code

Gabriel V. Subramanian
Naval - Northern Division
P.O. Box 833971
Richardson, TX 75083-8711
Phone: 214-665-6214
Email: gsubram@nortel.com

Eric J. Byrne
Department of Computer Science and Engineering
The University of Texas at Austin
Box 19813
Austin, Texas 78798-0198
Phone: 817-272-2141
Email: byrne@cs.utexas.edu

Abstract

The practice of software development continues to shift towards the use of object-oriented approaches. The requirements for object models in the highly commercial software industry, however, remain largely unmet. At the same time, the need for maintaining and extending existing software systems is a major concern for many organizations. This paper describes a methodology for deriving an object model from existing Fortran code. The methodology involves the use of a Fortran-to-object model translator, the use of a graphical user interface to assist in the derivation of the object model, and the use of a Fortran-to-object model translator to generate the object model. The paper concludes with a discussion of the implications of this methodology for the future of software development.

1. Introduction

In a recent survey of software engineering, the authors identified a number of key trends in the field. One of the most significant trends was the increasing use of object-oriented programming. This trend is expected to continue in the future, as more and more organizations adopt object-oriented approaches to software development. This paper describes a methodology for deriving an object model from existing Fortran code. The methodology involves the use of a Fortran-to-object model translator, the use of a graphical user interface to assist in the derivation of the object model, and the use of a Fortran-to-object model translator to generate the object model. The paper concludes with a discussion of the implications of this methodology for the future of software development.

Visual Impact Analysis

Kirk B. Callender
Computer Science Department
Loyola College in Maryland
Baltimore, MD 21159
Phone: 410-417-2854

Abstract

The purpose of this paper is to describe a methodology for performing visual impact analysis. Visual impact analysis is a technique used to assess the potential impact of changes to a software system. This paper describes a methodology for performing visual impact analysis. The methodology involves the use of a graphical user interface to assist in the derivation of the object model, and the use of a Fortran-to-object model translator to generate the object model. The paper concludes with a discussion of the implications of this methodology for the future of software development.

1. In

The purpose of this paper is to describe a methodology for performing visual impact analysis. Visual impact analysis is a technique used to assess the potential impact of changes to a software system. This paper describes a methodology for performing visual impact analysis. The methodology involves the use of a graphical user interface to assist in the derivation of the object model, and the use of a Fortran-to-object model translator to generate the object model. The paper concludes with a discussion of the implications of this methodology for the future of software development.

Algorithmic Analysis of the Impact of Changes to Object-Oriented Software

L. J. Callender
Loyola College in Maryland
Baltimore, MD 21159
Phone: 410-417-2854

Abstract

The purpose of this paper is to describe a methodology for performing algorithmic analysis of the impact of changes to object-oriented software. Algorithmic analysis is a technique used to assess the potential impact of changes to a software system. This paper describes a methodology for performing algorithmic analysis of the impact of changes to object-oriented software. The methodology involves the use of a graphical user interface to assist in the derivation of the object model, and the use of a Fortran-to-object model translator to generate the object model. The paper concludes with a discussion of the implications of this methodology for the future of software development.

Panel on "How Much Has Software Maintenance Changed Since 1983?"

Participants

Nancy Schuchman (Chair and Organizer), Naval Postgraduate School
Daf Kalef, Lockheed Martin Space Information Systems
Tom Foght, Techtel, Inc.
Michael Dwyer, Software Management Network

Abstract

The purpose of this panel is to discuss the changes in software maintenance since 1983. The panel will discuss the challenges of software maintenance and the impact of changes in software development practices. The panel will also discuss the impact of changes in software development practices on software maintenance. The panel will conclude with a discussion of the implications of these changes for the future of software maintenance.

1. Panel statements

1.1. How much has software maintenance changed since 1983?

The purpose of this panel is to discuss the changes in software maintenance since 1983. The panel will discuss the challenges of software maintenance and the impact of changes in software development practices. The panel will also discuss the impact of changes in software development practices on software maintenance. The panel will conclude with a discussion of the implications of these changes for the future of software maintenance.

Collaboration in Scientific Communities

Papers and Authors

IWST and VISSOFT



Deriving an Object Model from Legacy Fortran Code

Gökül Y. Subramanian, Eric J. Byrne
Naval Air Warfare Center
P.O. Box 833971
Richland, TX 79083-8971
Phone: 214-645-8214
Email: gsubram@navair.navy.mil

Department of Computer Science and Engineering
The University of Texas at Austin
Box 19815
Austin, Texas 78726-0198
Phone: 817-272-2141
Email: byrne@cs.utexas.edu

Abstract
The practice of software development continues to shift towards the use of object-oriented approaches. The requirements for the next generation of software systems are increasing rapidly. At the same time, the complexity of these systems is increasing. This paper presents an object-oriented software analysis tool that can be used to derive an object model from existing Fortran code. The tool is designed to be used by software engineers who are not familiar with object-oriented programming. The tool is designed to be used by software engineers who are not familiar with object-oriented programming. The tool is designed to be used by software engineers who are not familiar with object-oriented programming.

1. Introduction
It is a natural consequence of software engineering that software systems are designed to be used by software engineers who are not familiar with object-oriented programming. The tool is designed to be used by software engineers who are not familiar with object-oriented programming.

Visual Impact Analysis

Kirk B. Callender
Computer Science Department
Lynch College in Ireland
Dublin, Ireland
Phone: 353-1-452-2854
Email: kcallender@lync.ie

Abstract
The paper presents a visual impact analysis tool that can be used to derive an object model from existing Fortran code. The tool is designed to be used by software engineers who are not familiar with object-oriented programming.

1. Introduction
It is a natural consequence of software engineering that software systems are designed to be used by software engineers who are not familiar with object-oriented programming. The tool is designed to be used by software engineers who are not familiar with object-oriented programming.

Algorithmic Analysis of the Impact of Changes to Object-Oriented Software

L.L.L. L.L.L.
2300 Charleston Blvd., Suite 800
Arlington, VA 22204
Phone: 703-516-7394
Email: lll@lll.com

A. Jefferson O'Hall
IBM Department, 404
Georgia Institute of Technology
Atlanta, GA 30332-0444
Phone: 404-893-1851
Email: o'hall@cs.gatech.edu

Abstract
The paper presents an algorithmic analysis tool that can be used to derive an object model from existing Fortran code. The tool is designed to be used by software engineers who are not familiar with object-oriented programming.

1. Introduction
It is a natural consequence of software engineering that software systems are designed to be used by software engineers who are not familiar with object-oriented programming. The tool is designed to be used by software engineers who are not familiar with object-oriented programming.

Panel on "How Much Has Software Maintenance Changed Since 1983?"

Participants
Norman Schoenfeld (Chair and Organizer), Naval Postgraduate School
Dud Chapman, IBM Ltd
Ted Kiefer, Lockheed Martin Space Information Systems
Tom Fagan, Techstar, Inc.
Nicholas Fragomeni, Software Management Network

Abstract
The panel on "How Much Has Software Maintenance Changed Since 1983?" was held on 10 October 1993 at the IEEE International Conference on Software Maintenance. The panel was moderated by Norman Schoenfeld. The panelists were Ted Kiefer, Dud Chapman, Tom Fagan, and Nicholas Fragomeni. The panel discussed the state of software maintenance in 1983 and in 1993. The panelists discussed the challenges of software maintenance and the role of software engineers. The panelists discussed the importance of software maintenance and the need for software engineers to be involved in software maintenance.

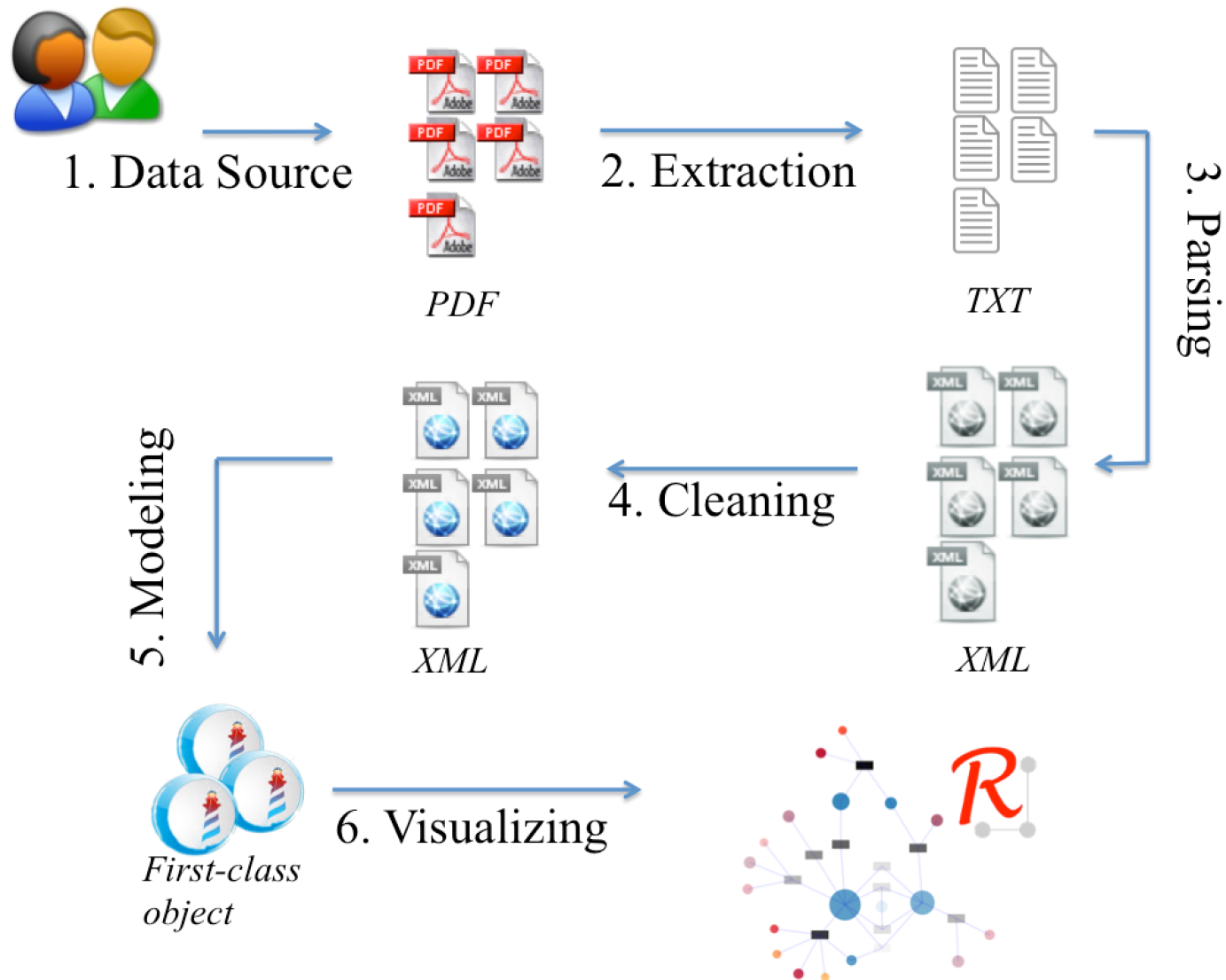
1. Introduction
The panel on "How Much Has Software Maintenance Changed Since 1983?" was held on 10 October 1993 at the IEEE International Conference on Software Maintenance. The panel was moderated by Norman Schoenfeld. The panelists were Ted Kiefer, Dud Chapman, Tom Fagan, and Nicholas Fragomeni. The panel discussed the state of software maintenance in 1983 and in 1993. The panelists discussed the challenges of software maintenance and the role of software engineers. The panelists discussed the importance of software maintenance and the need for software engineers to be involved in software maintenance.

1. Introduction
The panel on "How Much Has Software Maintenance Changed Since 1983?" was held on 10 October 1993 at the IEEE International Conference on Software Maintenance. The panel was moderated by Norman Schoenfeld. The panelists were Ted Kiefer, Dud Chapman, Tom Fagan, and Nicholas Fragomeni. The panel discussed the state of software maintenance in 1983 and in 1993. The panelists discussed the challenges of software maintenance and the role of software engineers. The panelists discussed the importance of software maintenance and the need for software engineers to be involved in software maintenance.

Search engines and Digital Libraries



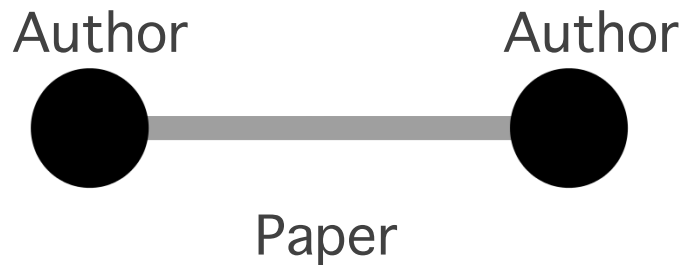
CommunityExplorer



Decreasing EC: Graph vs Bi-Graph

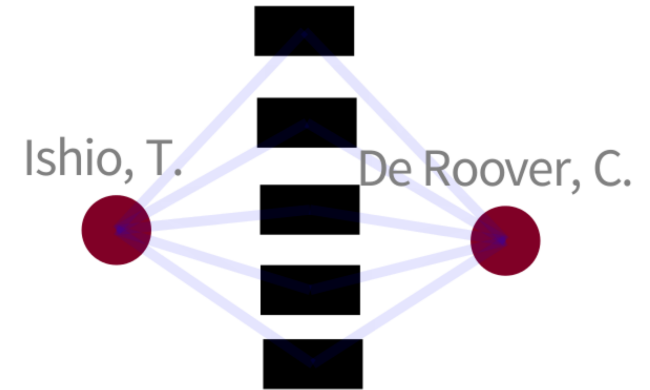
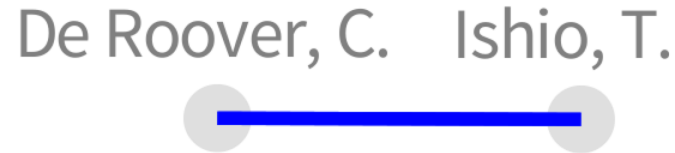
Graph

Bi-Graph

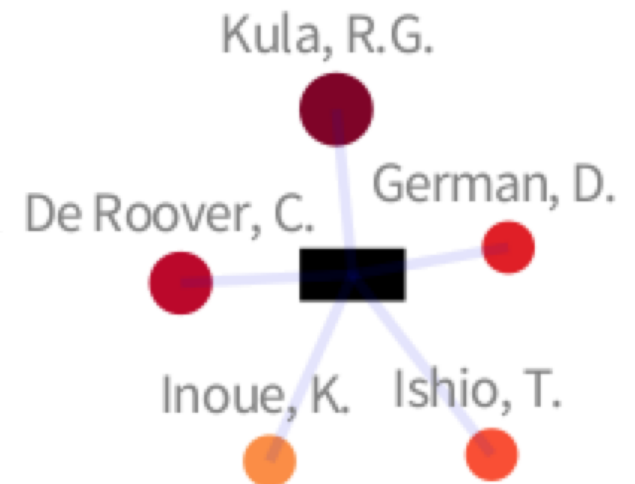
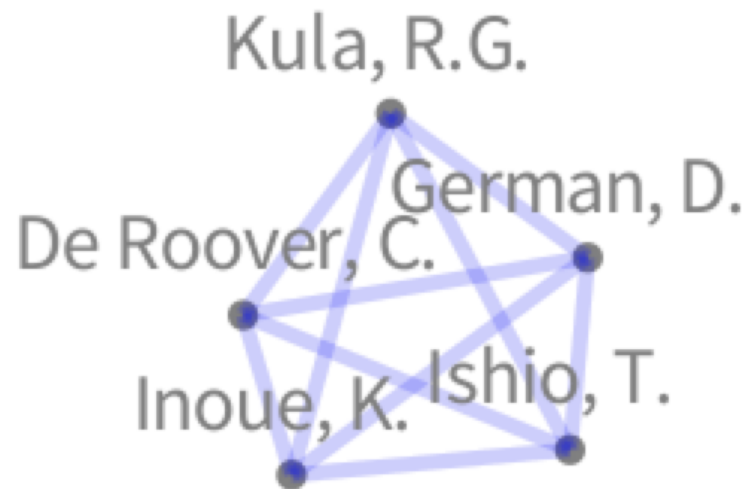


Graph vs. Bi-Graph

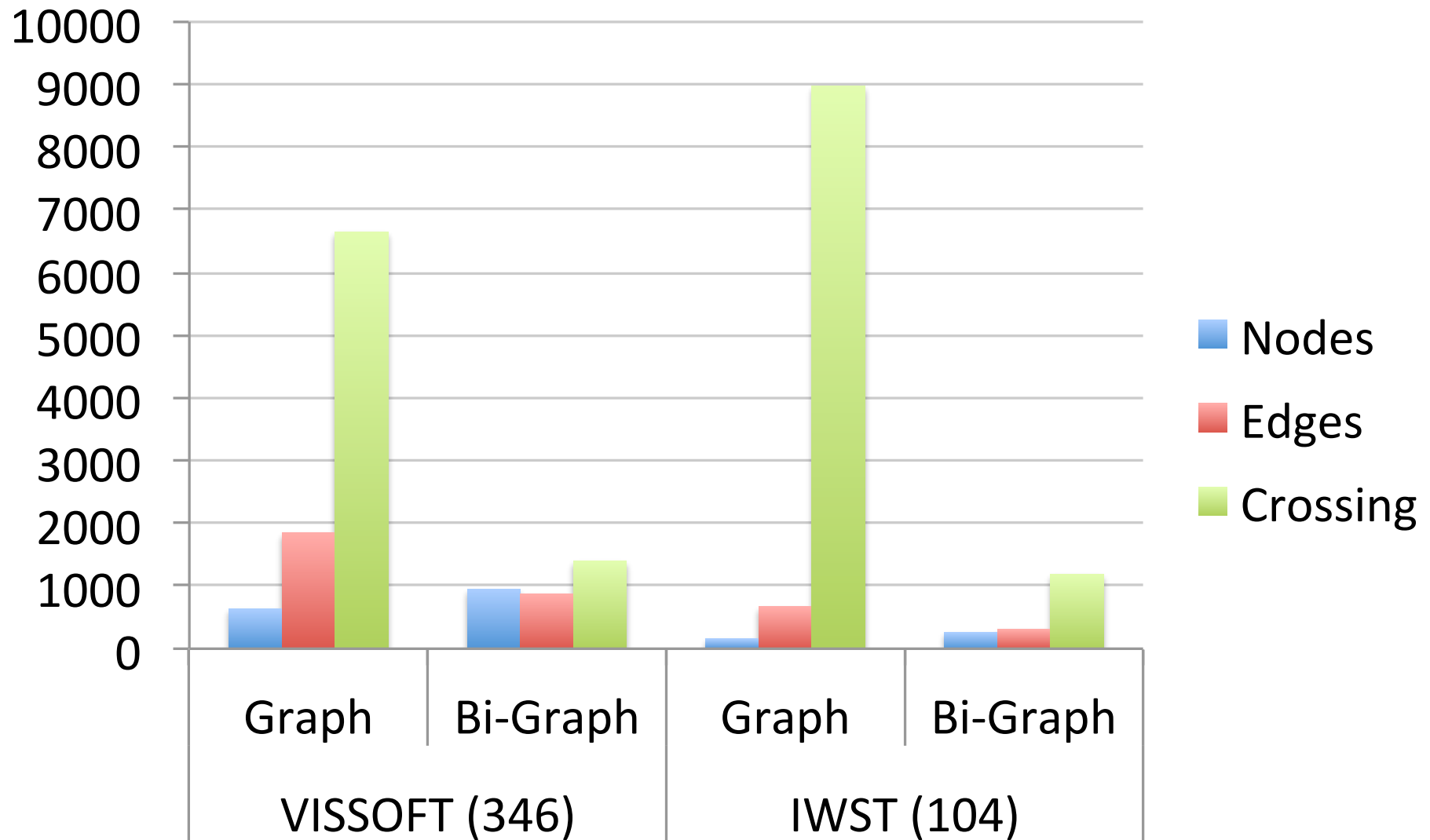
#Authors:2
#Papers:5



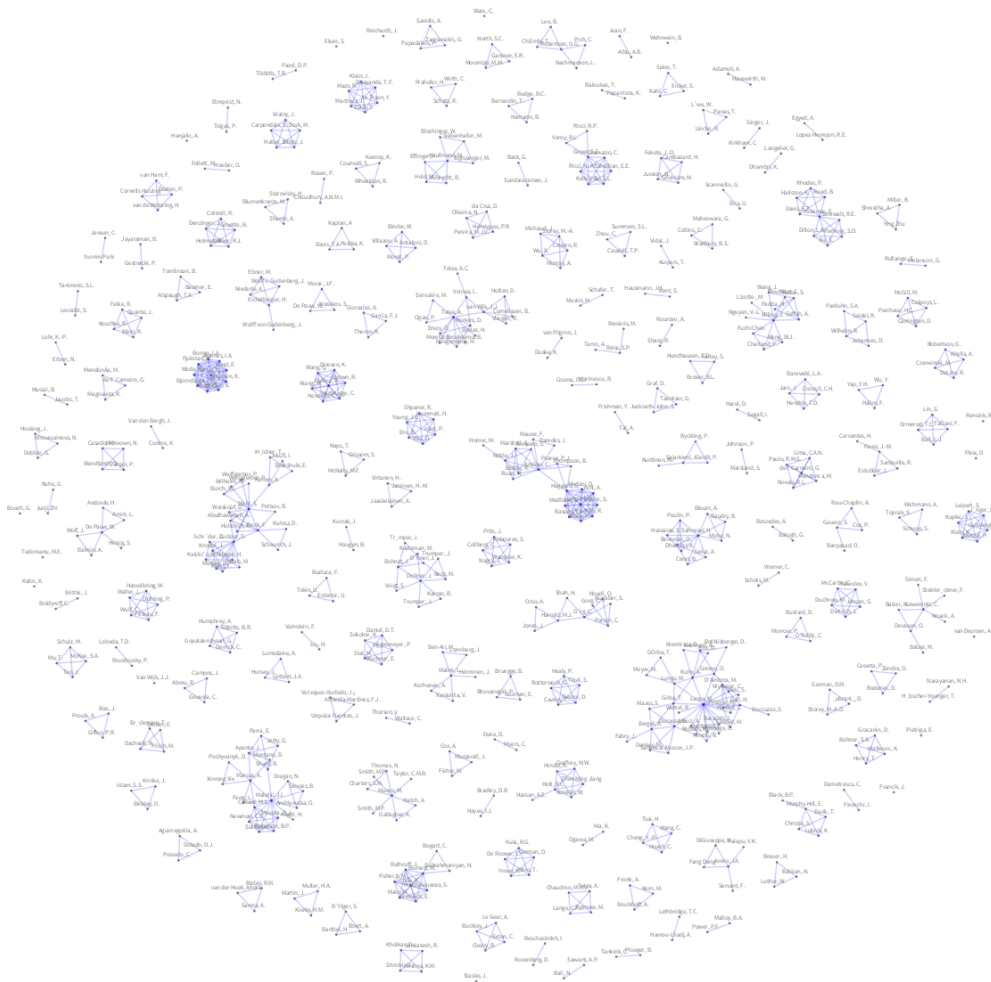
#Authors:5
#Papers:1



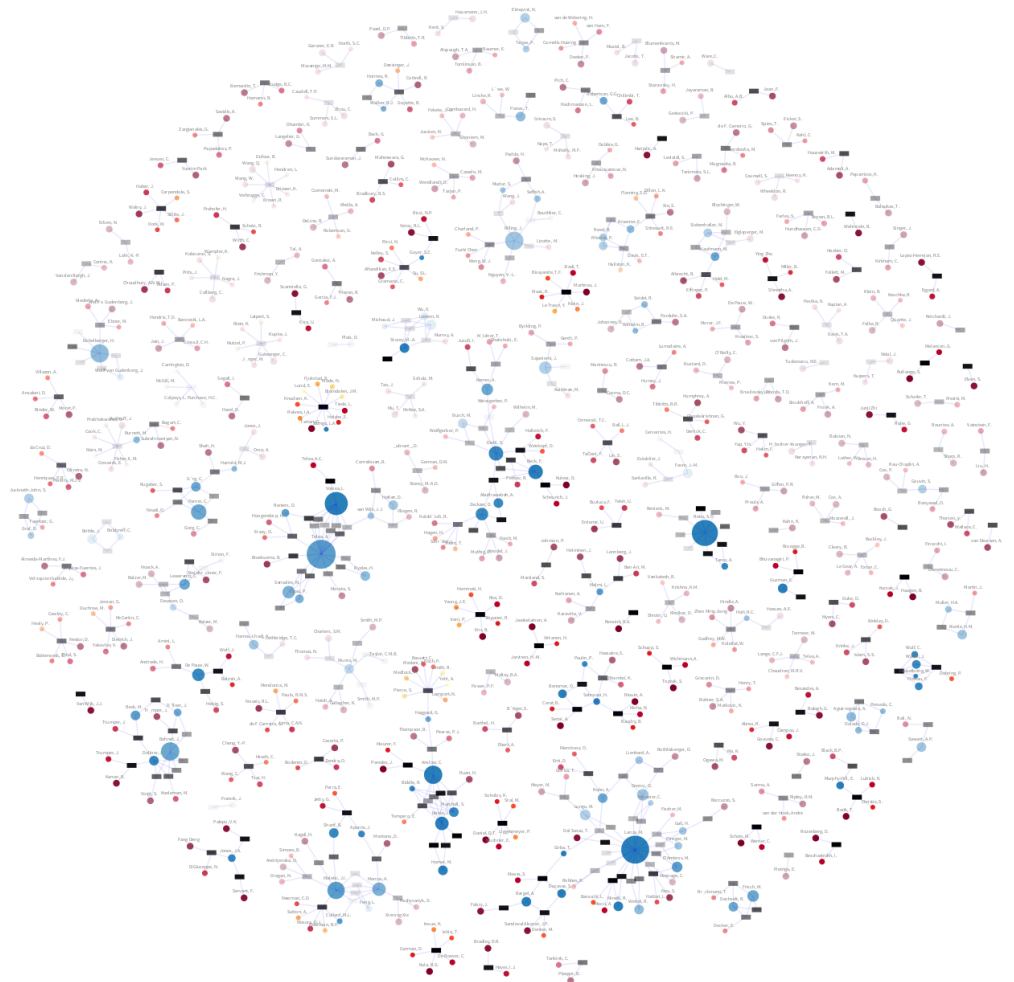
Case Study: VISSOFT vs IWST



VISSOFT (2002 - 2015)

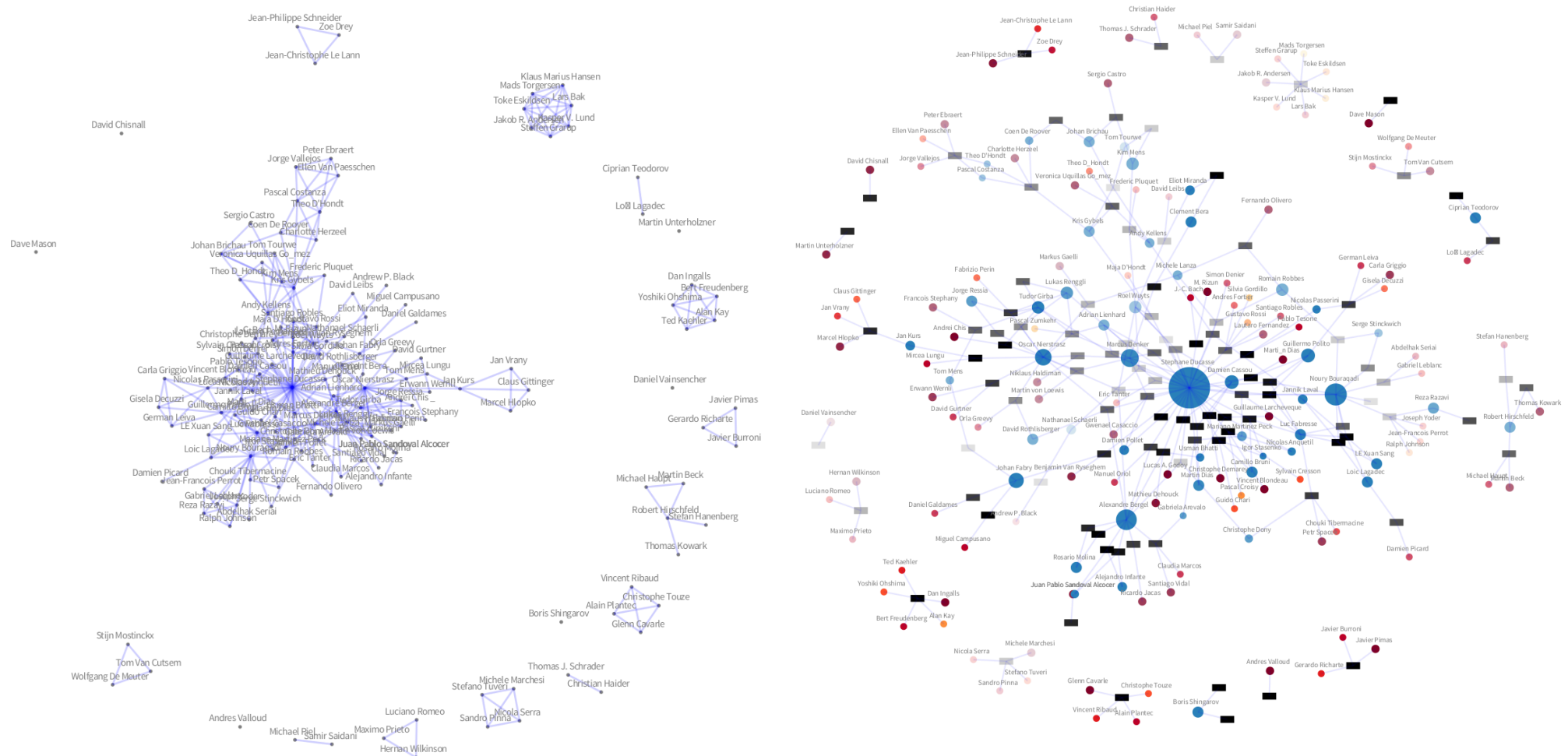


Graph



Bi-Graph

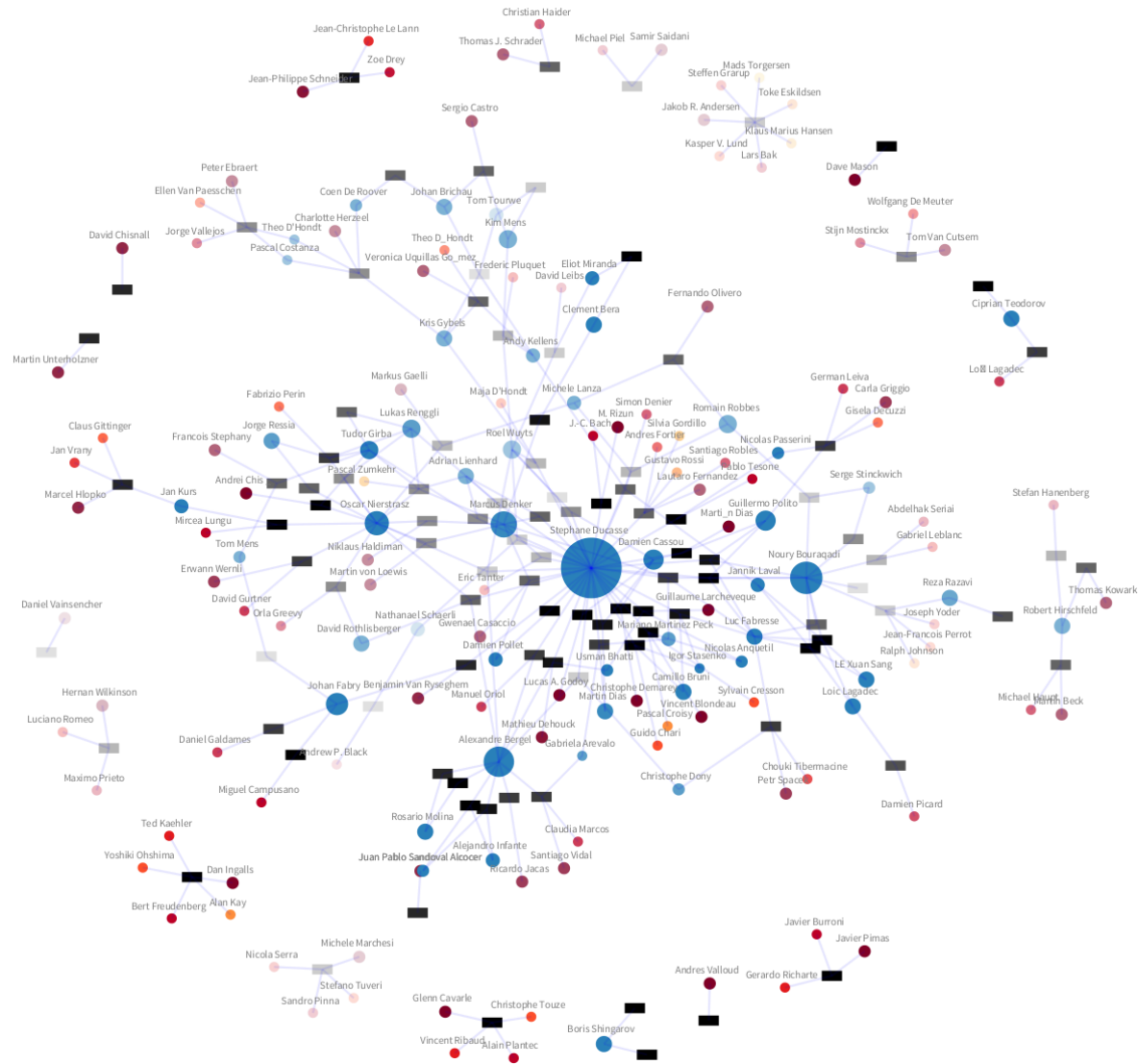
IWST (2002 - 2015)



Graph

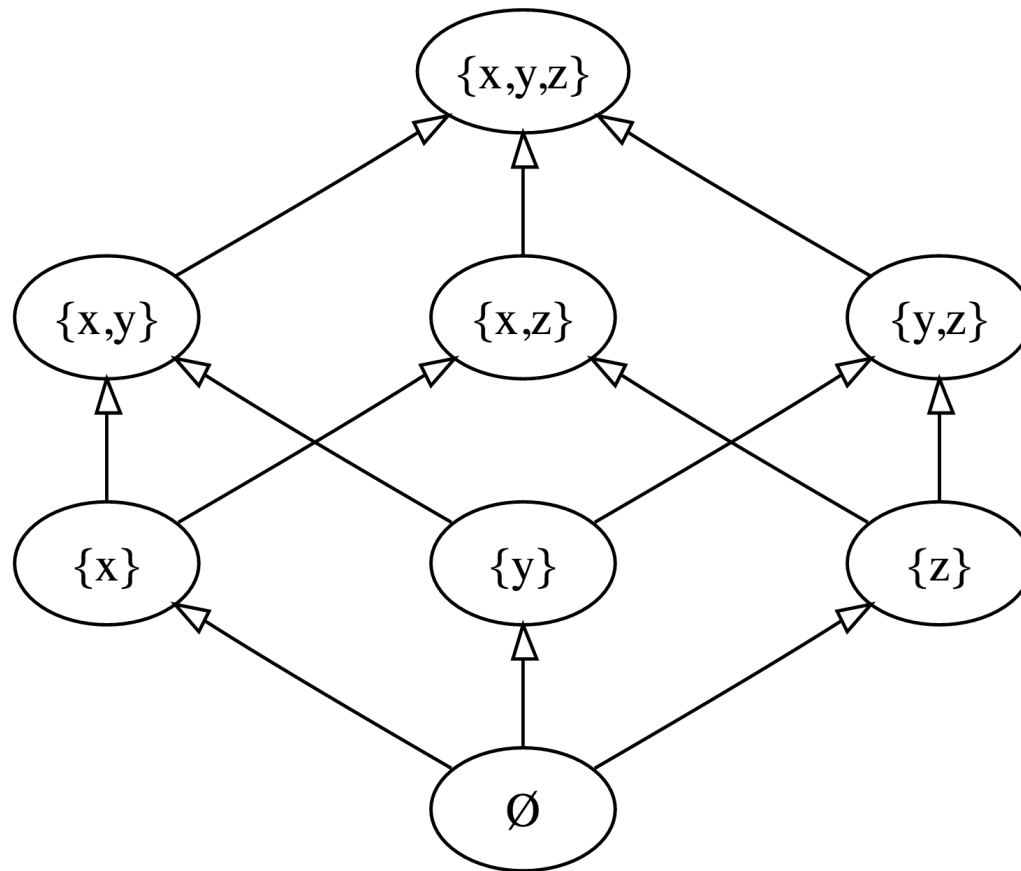
Bi-Graph

Demo



Future work: FCA

Formal Concept Analysis to decrease edge crossing.



Future work: Extending Pipeline

- Extend the pipeline to extract other fields such as paper's sections, figures, listings.
- Add visualizations for these types of data.

e.g., visualize how popular is Smalltalk across collaboration groups?

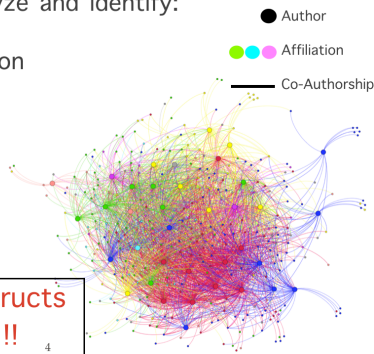
Summary

<http://scg.unibe.ch/research/CommunityExplorer>

Visualization of Collaboration

Support users to analyze and identify:

- Groups of collaboration
- Evolution of groups
- New & key authors



Edge crossing obstructs comprehension !!

Collaboration in Scientific Communities

Papers and Authors



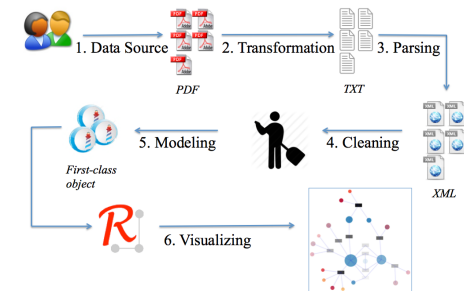
IWST and VISSOFT



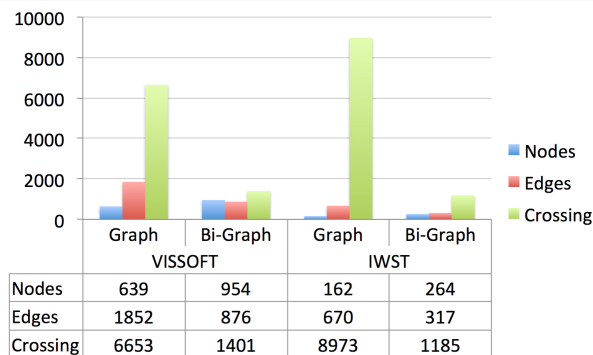
Search engines and Digital Libraries



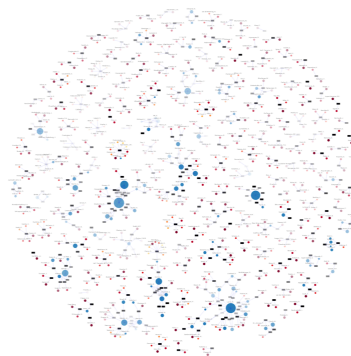
CommunityExplorer



Case Study: VISSOFT vs IWST



Demo



Future work: FCA

Formal Concept Analysis to decrease edge crossing.

Future work: Extending Pipeline

- Extend the pipeline to extract other fields such as paper's sections, figures, listings.
- Add visualizations for these types of data.

e.g., visualize how popular is Smalltalk across collaboration groups?