Structural Evolution of Software: Social Network Perspective

Naveen Kulkarni, Satya Prateek, Madhuri Dasa

Infosys® | Building Tomorrow's Enterprise

Outline of the Presentation

- Motivation
- General Terminology
- Background
- Describe the setup
- Describe the results
- Describe the conclusions

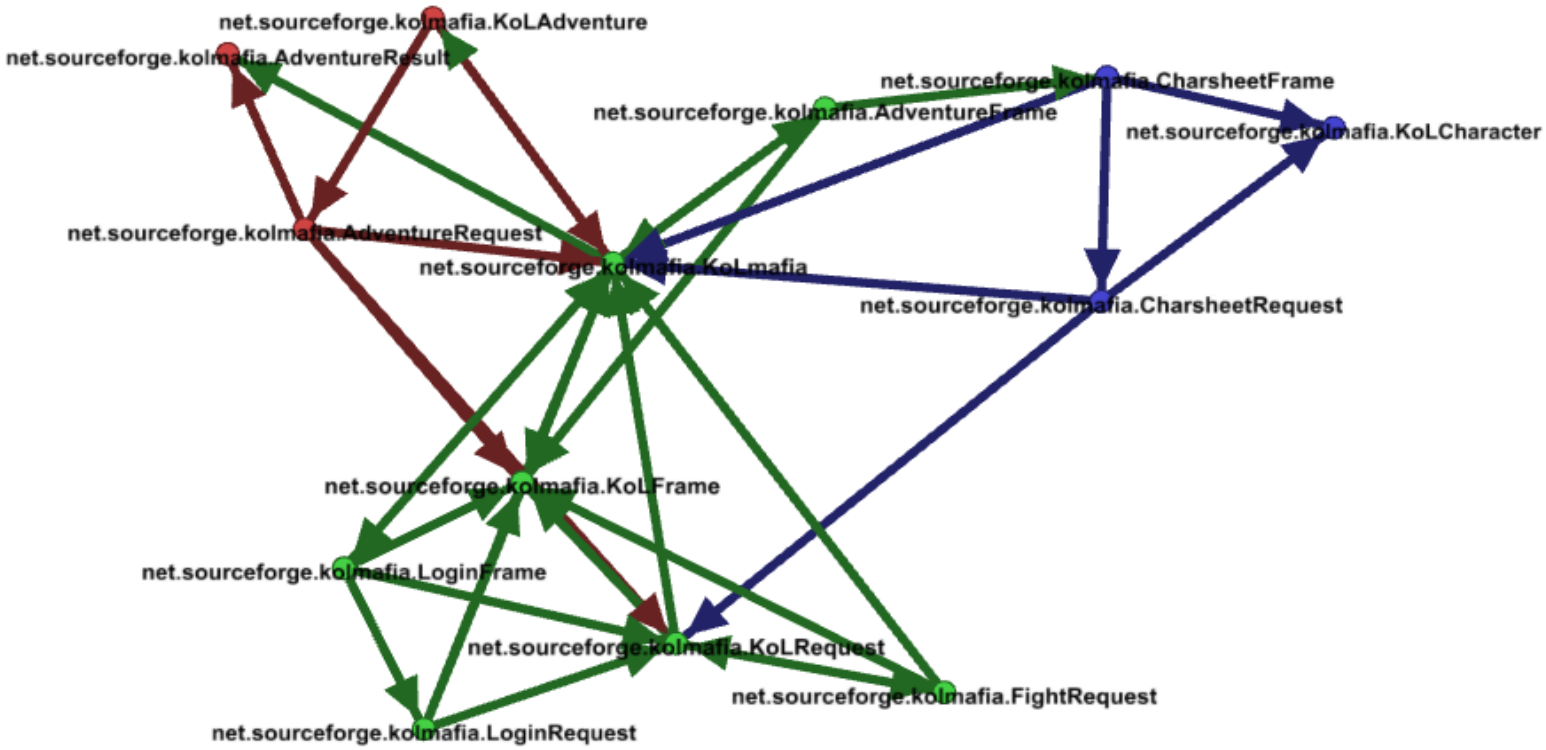
Motivation

- The structure of an Object Oriented (OO) System can degrade due to continuous changes[1]
- Standard OO metrics typically used to measure and monitor structural evolution.
- However OO metrics are object-class centric and do not take into account complex relations between classes.
- Complex network theories which study the relationships between nodes in networks can be used to study evolution.

1) Bianchi et al. “Evaluating Software Degradation through Entropy”, 7th International Symposium on Software Metrics, 2001

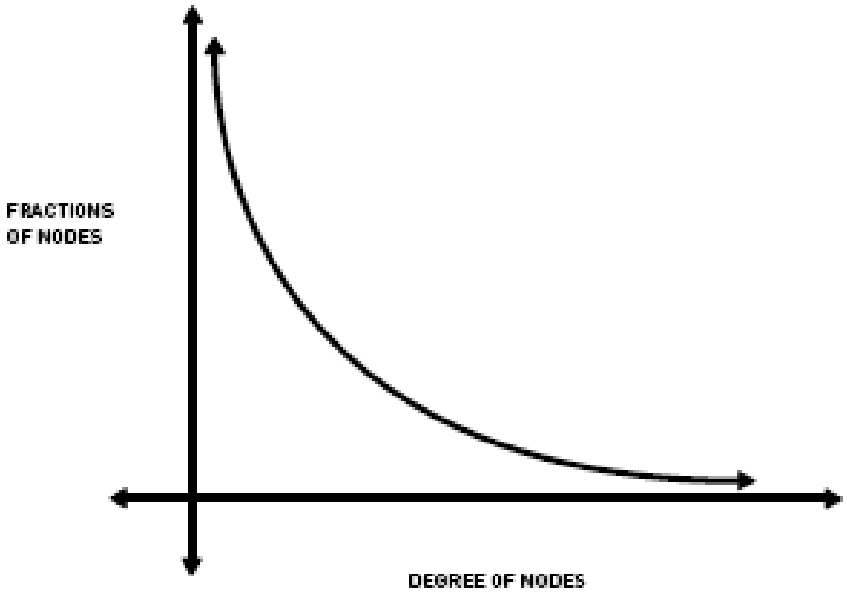
General Terminology

- Class Collaboration Network (CCN)



General Terminology

- Class Collaboration Network (CCN)
- Scale-Free Network



General Terminology

- Class Collaboration Network (CCN)
- Scale-Free Network
- Structural Complexity

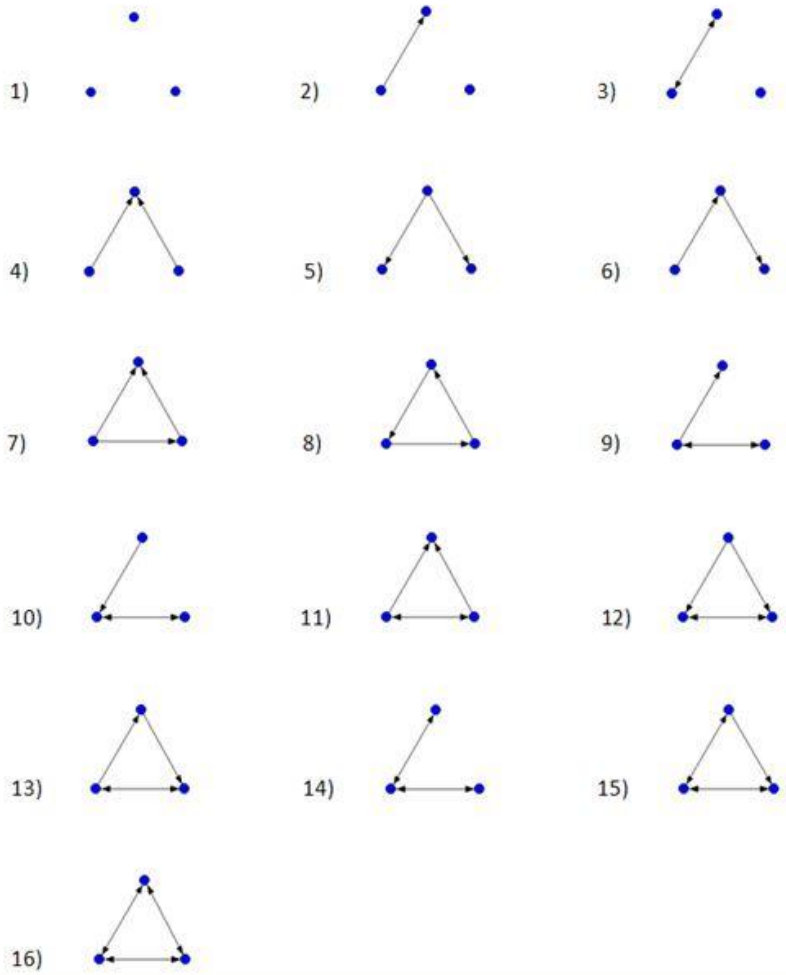
Structural Complexity is defined[2] as the product of the release level measure of coupling (Cbo) and the release level measure of the Lack of Cohesion (Lcom)

$$SC = Cbo_R \times Lcom_R$$

2) Darcy et al., Exploring complexity in open source software: Evolutionary patterns, antecedents, and outcomes, HICSS, 2010

General Terminology

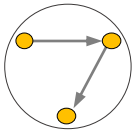
- Class Collaboration Network (CCN)
- Scale-Free Network
- Structural Complexity
- Motifs



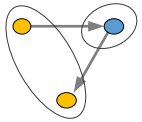
General Terminology

- Class Collaboration Network (CCN)
- Scale-Free Network
- Structural Complexity
- Motifs
- Brokerage Roles

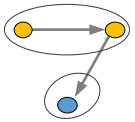
Coordinator



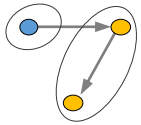
Itinerant



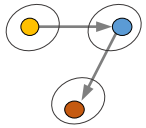
Representative



Gatekeeper

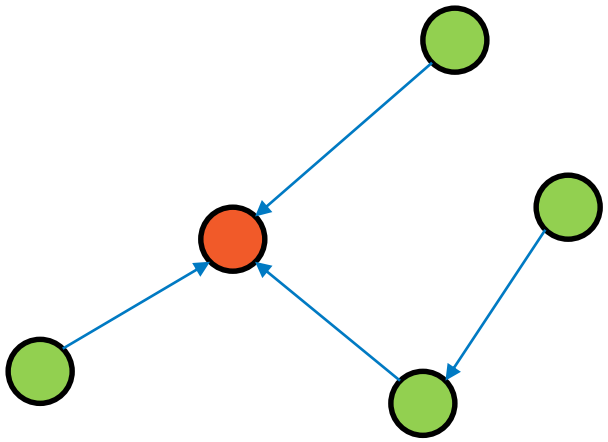


Liaison



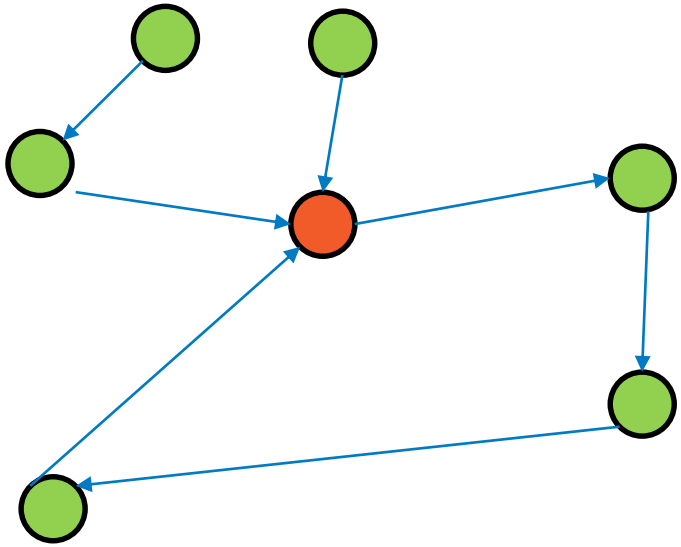
General Terminology

- Class Collaboration Network (CCN)
- Scale-Free Network
- Structural Complexity
- Motifs
- Brokerage Roles
- Proximity Prestige



General Terminology

- Class Collaboration Network (CCN)
- Scale-Free Network
- Structural Complexity
- Motifs
- Brokerage Roles
- Proximity Prestige
- Constraint



Background

- Studies have shown that CCN's exhibit scale-free properties[3]
- The emergence of scale-free network from what could have been a random network suggest inherent process of structural optimization.
- Thus we posit that SNA techniques will provide more understanding of software evolution of software than traditional OO metrics

3) Hyland-Wood et al., Scale-free nature of java software package, class and method collaboration graphs, 5th International Symposium on Empirical Software Engineering, 2006

Problem Statement

Can the process of software structural evolution be better understood from a social network perspective?

Experimental Setup

- Studied evolution of 8 open source software systems.
- Analyzed each software system over 3 phases:

  PR1 - Within 1 year of first public release

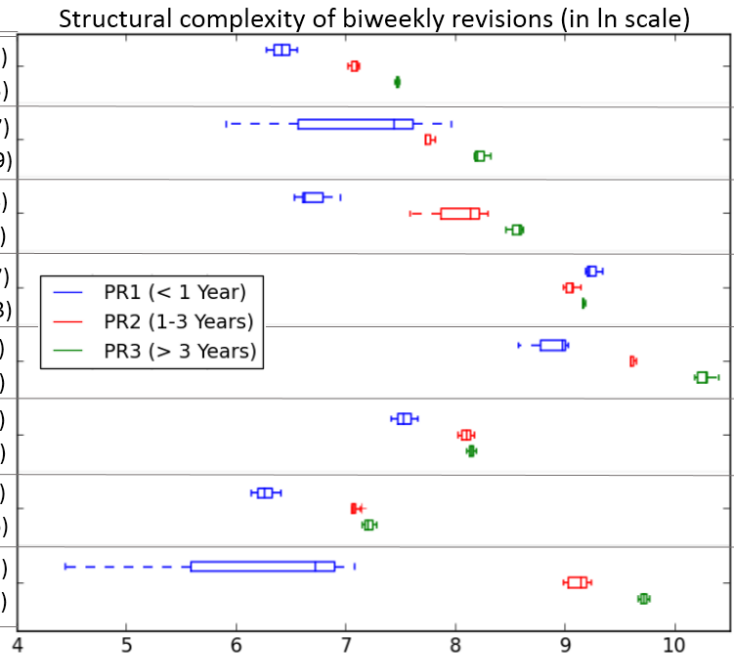
  PR2 - Between 1-3 years after first public release

  PR3 - After 3 years of it's first public release
- Within each phase we consider a 6 month period for analysis.
- Bi-weekly revisions were extracted for each software and the metrics were calculated on that.

Analysis of Structural Complexity

Revisions	PR1, <1 years	PR2, 1-3 years	PR3, >3 years
Ant	24-Oct-2000 (114) 02-Mar-2001 (118)	16-Apr-2003 (298) 16-Oct-2003 (325)	16-Dec-2006 (494) 30-Jun-2007 (503)
Azureus	03-Oct-2003 (132) 16-Mar-2004 (694)	03-May-2005(1397) 11-Sep-2005 (1509)	21-Aug-2009(1827) 20-Feb-2010 (1939)
Cassandra	23-Jul-2009 (315) 19-Jan-2010 (274)	28-Aug-2010 (295) 05-Mar-2011 (444)	14-Jan-2013 (634) 21-Jun-2013 (814)
FindBugs	25-Jun-2004 (245) 22-Dec-2004 (281)	11-Jun-2006 (429) 23-Dec-2006 (539)	21-Aug-2009(1027) 25-Mar-2010(1043)
Groovy	12-May-2004 (280) 29-Sep-2004 (299)	30-Apr-2007 (487) 29-Sep-2007 (623)	23-Oct-2008 (839) 06-Apr-2009 (888)
iText	18-Apr-2002 (214) 26-Sep-2002 (200)	16-Nov-2004 (297) 06-Apr-2005 (326)	11-Apr-2007 (459) 09-Sep-2007 (394)
JabRef	20-Feb-2004 (118) 27-Aug-2004 (155)	17-Aug-2006 (358) 30-Jan-2007 (392)	01-Feb-2007 (392) 01-Aug-2007 (426)
KolMafia	25-Dec-2004 (13) 27-Jun-2005 (104)	15-May-2007 (177) 12-Oct-2007 (184)	03-Jan-2009 (348) 13-Jul-2009 (366)

(a)

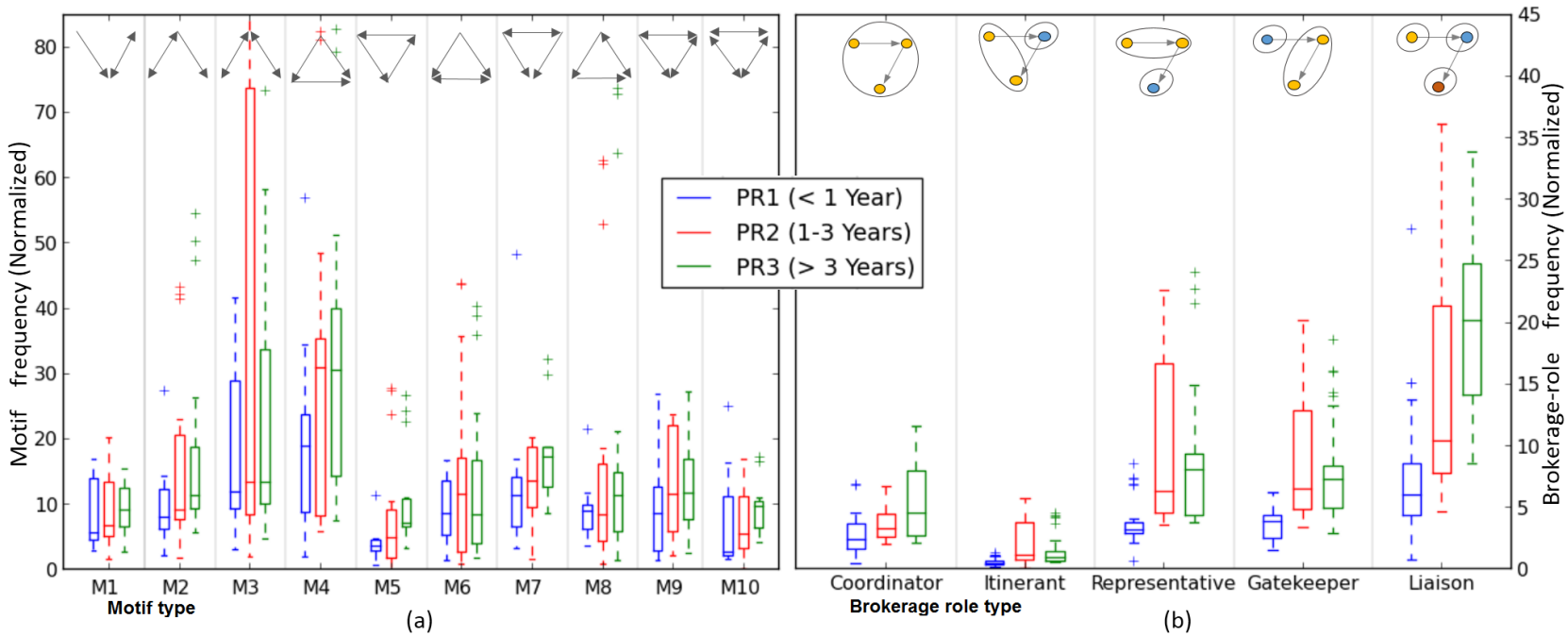


(b)

Analysis of Structural Complexity

- Structural Complexity of PR1 had higher variation than later releases.
- This suggests that as software system evolves it undergoes a form of structural optimization.
- We conjecture that this optimization process is constrained by two factors
  a) Complex relations between object-classes representing concepts that are hard to isolate
  b) Core object-classes within each concept that are hard to modify.

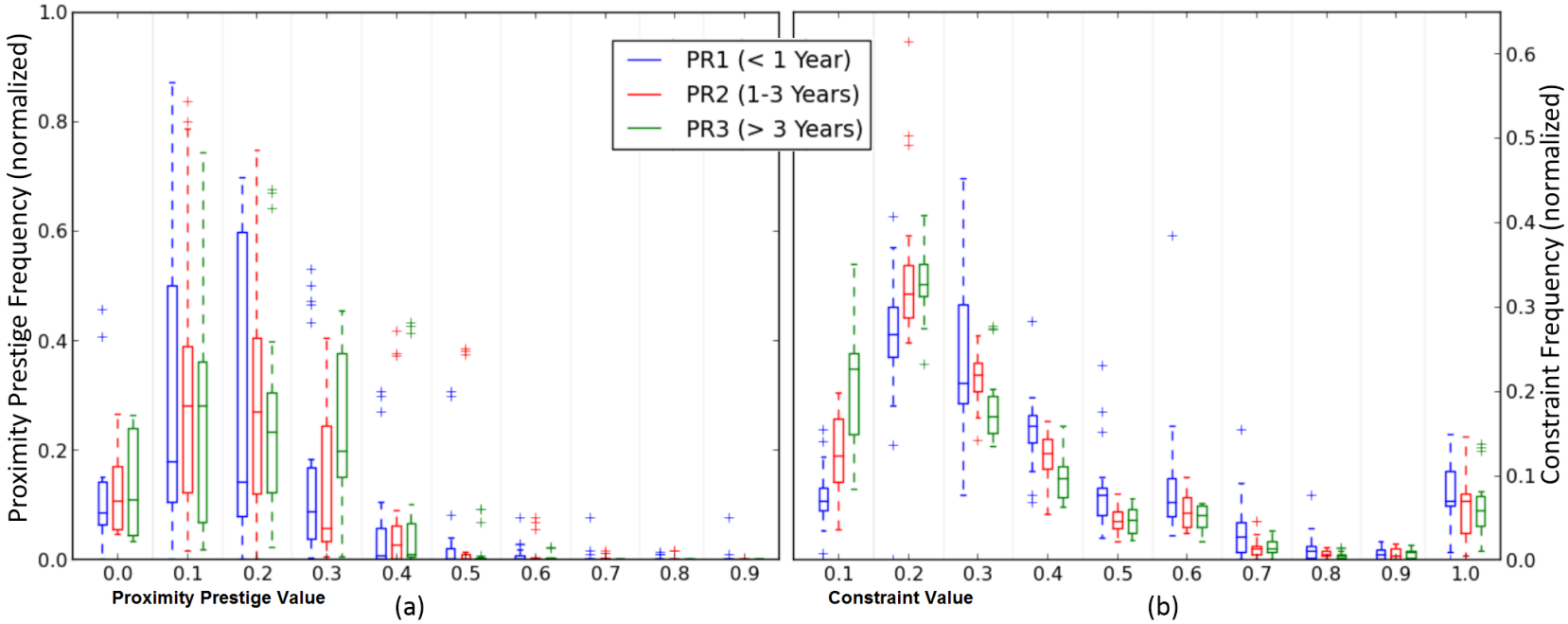
Analysis of Social Relation



Analysis of Social Relation

- Simple motifs (M1 - M3) and complex motifs (M4 -M10) were found to occur throughout. Also ratio between simple and complex motifs were maintained.
- Preferential selection of complex motifs for few periods was observed.
- For brokerage roles all classes had had uniform affiliations to all roles during PR1. During PR2 and PR3 classes had similar affiliations towards representative and gatekeeper roles.
- Higher affiliation towards liaison role in PR2 and PR3

Analysis of Social Capital



Analysis of Social Capital

- During PR1 all classes were equally distributed among PP values.
- Increase in the classes with PP value 0.1 and 0.2 can be seen during PR2
- Increase in number of classes attaining PP value 0.3 during PR3
- Increase in the number of classes with lower constrain values, while number of classes with higher constraint values decreased.

Conclusions

- As software system grows, simple relations evolve towards complex relations.
- Certain complex relations were observed at different stages of evolution. This suggests that each of the complex relations also further evolves independently.
- Complex relations are more likely to form between groups rather than within groups.
- Now Influential nodes are an outcome of the hierarchical optimization process. But the existence of many influential nodes suggests that multiple concepts acquire hierarchy independently.

Applications

- Analyzing distribution of complex motifs representing collaboration patterns will help uncover structural evolution patterns.
- Calculation of groups in our study was done using topological features. If group distribution was identified based on the concepts then we can study the complex relationships among them.
- Social network perspective can help software development team to characterize evolutionary path and identify relations that are overly complex or resistant to change.

Thank You

Infosys® | Building Tomorrow's Enterprise