Designing Software Ecosystems

How Can Modeling Techniques Help?

Mahsa H. Sadi , Eric Yu

Department of Computer Science

University of Toronto

E-mail: mhsadi@cs.toronto.edu

Exploring Modeling Methods for Systems Analysis and Design (EMMSAD) Working Conference At CAiSE 2015 , June 9th, 2015

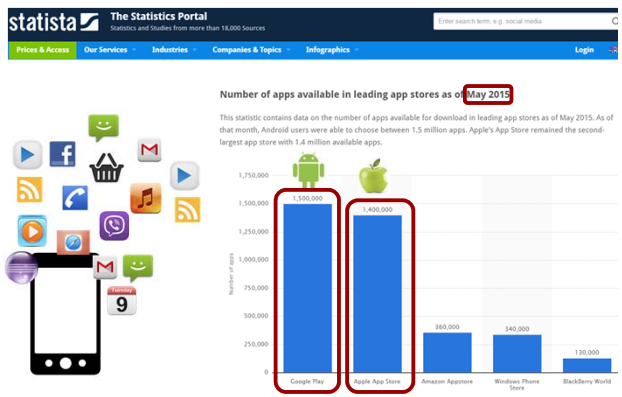
Introduction

Software Ecosystem

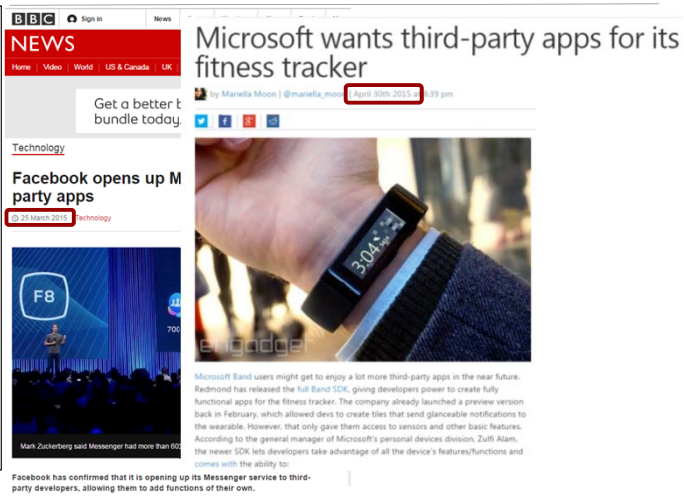
> A recent practice in which a software platform developer collaborates with external developers in order to develop and extend a software platform for a shared market
>
> - Jansen, Finkelstein, & Brinkkemper (2009)

Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009).
"A sense of community: A research agenda for software ecosystems"
In 31st International Conference on Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. (pp. 187-190). IEEE.

Example: The Mobile Software Ecosystem



Other Examples of Software Ecosystems



Research Problem

Challenging issue (for platform developers)

- How to build, grow and sustain a software ecosystem?

Developing a software ecosystem is multi-faceted

- Needs to have
  - A viable business model
  - A well-organized inter-organizational interaction model
  - A well-designed collaborative software development process
  - A software platform that enables the collaboration

Position in This Study

Systematic methods are required

- To support the design and development of software ecosystems
  - Models provide a sound basis for systematic design and development

Modeling can help.

How can modeling help?

Contributions of This Study

1. A Set of Modeling Requirements
2. Reviewing a Set of Modeling Techniques
3. Assessment of the Modeling Techniques
4. Identifying the Gaps

1. Introduction
2. Modeling Requirements
3. Review
4. Assessment
5. Conclusions

Designing Software Ecosystems

A Set of Descriptive and Analytical Requirements

- A Set of Requirements for Describing Software Ecosystems
- A Set of Requirements for Analyzing Software Ecosystems

Describing Software Ecosystems

A Set of Requirements

Primary Elements

- Collaborators
- Interactions
- Activities (or Responsibilities)

Ancillary Elements

- Type
- Constraint
- Attribute
- Characteristic of Collaboration

Describing Software Ecosystems

Primary Elements

Collaborators - Identifying the members and their roles
- **Example** - The keystone platform developer, The external software developers, Software Companies

Interactions - Identifying the relationships among members
- **Example** - The business or technical relationships among the members

Activities (or Responsibilities) - Specifying the activities and commitments of the members
- **Example** - Specific business or software development activity performed by a member

Describing Software Ecosystems

Ancillary Elements

Further Details in the Paper

Type - Specifying different types of the primary elements
- **Example:** Identifying a financial or knowledge exchange relationship

Constraint - Specifying constraints and rules on the primary elements
- **Example:** Describing the level of access of a software developer to the platform

Attribute - Specifying characteristics of primary elements
- **Example:** Identifying an important or a reliable collaborator or a critical interaction

Characteristic of Collaboration - Specifying the attributes of the configuration of the primary elements
- **Example:** Identifying a healthy, productive or secure collaboration between two or more collaborators

Identifying the Analysis Requirements

A top-down approach

1. Identifying the general steps and activities in building or transitioning into a software ecosystem approach
2. Identifying the analysis concerns and questions that are raised in each of the steps and activities

Models should help answer the identified questions.

Note:
Cross-cutting analysis and design concerns
They can be raised for
A software business manager
A software project manager
A software developer

Steps in the Development of a Software Ecosystem

1. Setting up the software ecosystem

Further Details in the Paper

Main Activities:
- To motivate external developers to collaborate and contribute to a software platform

2. Opening up the software development processes and the software platform to external collaborators

3. Monitoring and governing the software ecosystems

Analysis Requirements in Software Ecosystems

1. Setting up the software ecosystem

To motivate external developers to collaborate and contribute to a software platform

S-1 Analyzing the incentives and motivations of collaborators:

Example Questions [1]:

Q1. How to motivate external developers and companies to participate and contribute?

Q2. What are the intrinsic and extrinsic motivations of software developers and software companies for contributing to a platform?

[1] Koch, S., & Kerschbaum, M. (2014). Joining a smartphone ecosystem: Application developers' motivations and decision criteria. Information and Software Technology, 56(11), 1423-1435.

Requirements for Analyzing Software Ecosystems

Analyzing for:

Further Details in the Paper

- Incentives and Motivations of Collaborators
- Trust and Reliability
- Risk, Vulnerability, Tolerance, Costs and Benefits
- Distributing and Decentralizing Responsibility and Resources
- Distributing control, authority, decision making and access
- Distributing ownership and power
- Openness and sharing in collaboration
- Security and privacy
- Health, productivity, robustness, performance
- Alignment and conflict resolution

1. Introduction
2. Modeling Requirements
3. Review
4. Assessment
5. Conclusions

Review

Techniques Used for Modeling Software Ecosystems

1. Review Process

Further Details in the Paper

[An Overview of the Selected Techniques]

2. Summary of the Reviewed Modeling Techniques

Review Process

Collecting the Resources

- Using two recent systemic reviews of 2013 [1],[2] as an initial catalog for collecting the resources
- Updating and extending the collection with more recent literature

1. **Manikas, K., & Hansen, K. M. (2013)**. Software ecosystems–a systematic literature review. *Journal of Systems and Software*, *86*(5), 1294-1306.
2. **Werner, C., & Jansen, S. (2013)**. A systematic mapping study on software ecosystems from a three-dimensional perspective. Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, 59-81.

Review Process

Selecting the Resources

Inclusion Criteria:

1. The collected resource **must use a modeling techniques to represent the structure** of software ecosystems.
2. The modeling technique **must have a well-define syntax and semantics**.
3. The **members** of a software ecosystem and the **relationships** among them **must be explicitly modeled**.

Included Modeling Techniques:

SSN, i*, BMC, VN, e^{3}Value

Further Details in the Paper

Excluded Models: Applying the inclusion criteria led to omitting the following models: Product Deployment Context (PDC) Diagram, Technical Ecosystem Modeling Notation (TECMO) meta-model , UML Deployment Diagram, Associate Models, the SPO software ecosystem meta-model, SPEM meta-model, Graph representations, Food-web models

Summarizing Process

			SSN
A-1	Focal viewpoint	Business	+
		Inter-organizational	+
		Technical (Software Development)	*/+
A-2	Intended users	Software Business Manager	+
		Software Project Manager	–/–
		Software Developer	–/–
B-1	Support for analysis	Qualitative	+
		Quantitative	–/*
B-2	Representation mode	Textual	–
		Visual	+
B-3	Refinement and traceability		–
B-4	Multiple views		*
B-5	Formal syntax and semantics		+
B-6	Tool support		–
C-1	Experimentation maturity for software ecosystems		+
C-2	Methodology support for software ecosystems		×
C-3	Documentation support for software ecosystems		*

(A) The objective and the intended users

(B) General usability features

(C) The specific support and maturity of the techniques for modeling software ecosystems

An Overview of the Selected Techniques

Summary of the Reviewed Modeling Techniques

			SSN	i*	VN	BMC	e³V
A-1	Focal viewpoint	Business	+	+	+	+	+
		Inter-organizational	+	+	+	–/–	+
		Technical (Software Development)	*/+	*/+	–/–	–/–	–/–
A-2	Intended users	Software Business Manager	+	+	+	+	+
		Software Project Manager	–/–	–/+	–/–	–/–	–/*
		Software Developer	–/–	–/+	–/–	–/–	–/–
B-1	Support for analysis	Qualitative	+	+	+	+	+
		Quantitative	–/*	–/×	–/–	–/–	*
B-2	Representation mode	Textual	–	–	–	+	–
		Visual	+	+	+	–	+
B-3	Refinement and traceability		–	+	–	–	–
B-4	Multiple views		*	–	–	–	–
B-5	Formal syntax and semantics		+	+	–[4]	+	+
B-6	Tool support		–	+	–	+	+
C-1	Experimentation maturity for software ecosystems		+	×	×	*	×
C-2	Methodology support for software ecosystems		×	–	–	–	–
C-3	Documentation support for software ecosystems		*	*	×	×	×

Focus on modeling business and inter-organizational relationships reflecting the viewpoint of software business managers

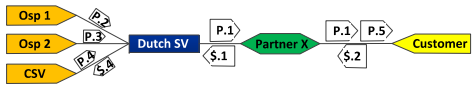
Poor support for representing technical relationships
The viewpoint of software project managers and software developers is not reflected.

1. Introduction
2. Modeling Requirements
3. Review
4. Assessment
5. Conclusions

Assessing the Reviewed Modeling Techniques

- **Evaluating Support for Describing Software Ecosystems**
- **Evaluating Support for Analyzing Software Ecosystems**

Evaluating Description Support of SSN



	P-1: Collaborator			P-2: Activity (or Responsibility)			P-3:Interaction			A-4
	A-1	A-2	A-3	A-1	A-2	A-3	A-1	A-2	A-3	
SSN	+/(Actor)			–			+/(I-O Flow)			–
	+	–	+	–	–	–	+	–	–	

Primary Elements

- P-1 Collaborator
- P-2 Interaction
- P-3 Activity (or Responsibility)

Ancillary Elements

- A1 Type
- A-2 Constraint
- A-3 Attribute
- A-4 Characteristic of Collaboration

Evaluation Results – Description Support

	P-1: Collaborator			P-2: Activity (or Responsibility)			P-3:Interaction			A-4
	A-1	A-2	A-3	A-1	A-2	A-3	A-1	A-2	A-3	
SSN	+/(Actor)			–			+/(I-O Flow)			–
	+	–	+	–	–	–	+	–	–	
i*	+/(Actor/ Role/Agent)			+/(Tasks/Resources/Goals/ Softgoals)			+ /(Strategic Dependency)			–
	–	–	–	–	–	–	–	–	+	
VN	+/(Actor)			–			+/(Activity Flow)			–
	–	–	–	–	–	–	+	–	–	
BMC	+/(Actor)			* /(Activities / Resources)			–			–
	–	–	–	–	–	–	–	–	–	
e^3V	+ / (Market Segment /Actor)			+ (Activities)			+/(I-O Flow/Activity Flow)			–
	–	–	–	–	–	–	+	–	–	

All the modeling techniques support representing collaborators

Except in i* and e^3V, the notion of activity is not supported

Evaluating Support for Analyzing Software Ecosystems

Evaluation Criteria

Information Support

Does the model express the required information to draw conclusion about an analysis concern?

Analysis Representation

Is the information related to the analysis captured inside the model or outside the model?

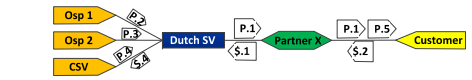
Alternative Analysis and Comparison

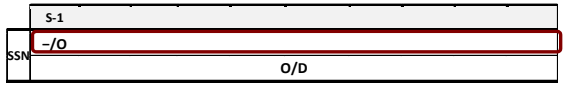
Does the modeling technique enable representing and comparing the consequences of two or more alternatives to address an analysis concern?

Type of Analysis

Does the modeling technique enable descriptive analysis or prescriptive analysis?

Evaluating Support of SSN for Analysis - 1





Analysis Requirements

S-1 Analyzing for Incentives and Motivations of Collaborators

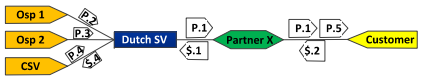
Q1. What are the intrinsic and extrinsic motivations of software developers and software companies for joining the software ecosystem?

Q2. How to foster collaboration and how to motivate external developers and companies to participate and contribute to the platform?

Evaluation Criteria (First Row)

- Information Support
- Analysis Representation

Evaluating Support of SSN for Analyzing - 2



	S-1	S-2	O-1	O-2	O-3	O-4	O-5	O-6	O-7	O-8
SSN	–/O	–/O	+/O	–/O	–/O	–/O	+/O	–/O	–/O	–/O
	O/D									

Analyzing for:

S-1 Incentives and Motivations of Collaborators

S-2 Trust and Reliability

O-1 Risk, Vulnerability, Tolerance, Costs and Benefits

O-2 Distributing and Decentralizing Responsibility and Resources

O-3 Distributing control, authority, decision making and access

Analyzing for:

O-4 Distributing ownership and power

O-5 Openness and sharing in collaboration

O-6 Security and privacy

O-7 Health, productivity, robustness, performance

O-8 Alignment and conflict resolution

Evaluation Results

	S-1	S-2	O-1	O-2	O-3	O-4	O-5	O-6	O-7	O-8
SSN	–/O	–/O	+/O	–/O	–/O	–/O	+/O	–/O	–/O	–/O
	O/D									
i*	+/I	+/I	+/I	+/I	+/I	–/I	+/I	+/I	–/I	+/I
	I/D									
VN	–/O	–/O	+/O	–/O	–/O	–/O	+/O	–/O	–/O	–/O
	O/D									
BMC	–/O	–/O	–/O	–/O	–/O	–/O	–/O	–/O	–/O	–/O
	O/D									
e³V	–/O	–/O	+/O	+/O	+/O	+/O	+/O	–/O	–/O	–/O
	O/D									

Poor information support for addressing the analysis concerns

Poor support for analysis representation

Poor support for alternative analysis and comparison

No support for prediction and prescription

1. Introduction
2. Modeling Requirements
3. Review
4. Assessment
5. Conclusions

Summary of the Review and Evaluation

Lack of support for representing the technical aspect of collaboration

Lack of alignment between the business and organizational viewpoints and the technical viewpoints

Weak representation support

Weak methodological support

Weak analysis support

Summary

Software ecosystem is a recent practice in software development

To support the systematic design of software ecosystems, modeling is a crucial tool

Modeling software ecosystems is in early stages

Contributions of this study

- A set of modeling requirements
- Assessing the support of a set of techniques that are currently used
- The gaps between the requirements and the current coverage of model-based approaches
- Suggestions for how to enhance the modeling support

Limitations of the Study

Confirming the initial set of modeling requirements

- Requirements are identified through the analysis of the published literature
- Need to be confirmed with practitioners and/or other active research groups in software ecosystems

Complementing the assessment of the modeling techniques

- Elaborating the evaluation criteria
- Conducting more detailed studies on the modeling techniques

References

Bosch, J. (2009)
"From software product lines to software ecosystems"
In Proceedings of the 13th International Software Product Line Conference (pp. 111-119). Carnegie Mellon University

Cataldo, M., & Herbsleb, J. D. (2010)
"Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration"
In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (pp. 65-72). ACM.

Sadi, M. H., Dai, J., & Yu, E. (2015)
Designing Software Ecosystems: How to Develop Sustainable Collaborations?.
In *Advanced Information Systems Engineering Workshops* (pp. 161-173). Springer International Publishing.

Sadi, M. H., & Yu, E. (2014)
"Analyzing the evolution of software development: From creative chaos to software ecosystems"
In IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), (pp. 1-11). IEEE.

E-mail: mhsadi@cs.toronto.edu