



ICSE  
4th International Workshop on  
Software Engineering for Systems-of-Systems  
Austin, Texas, USA  
Monday, May 16, 2016

# Modeling of Information Systems as Systems of Systems through DSM

Paolo Salvaneschi

University of Bergamo, Dept of Management,  
Information and Production Engineering and  
Salvaneschi & Partners

## Table of contents

- Problem statement
- Case study context
- Modeling Information System through DSM
- Implementation
- Query patterns
- Discussion and outlook

## Problem statement

- Large information systems are composed of dozens of software applications

Software application: programs that typically implement a business process or part of it.

## Problem statement

- Applications developed in house or acquired/adapted/integrated
- Applications evolve in time
- Different vendors and development teams work on parts of the whole system

## Problem statement

- Due to time pressure and lack of global knowledge:
  - Local changes may cause decay of modularity and increasing complexity of relationships between software components
  - Increasing difficulty to understand the whole system

## Problem statement

- The problem:  
understanding and managing knowledge about *the Information System-level architecture* where a software component is a software application
- The approach:  
Modeling the Information System as a System of Systems  
Preliminary results

## Case study context

- Two information systems
  - *System 1* - The Italian branch (more than 8000 employees) of a large european retail company. 100 applications + 3 large DB
  - *System 2* - A hospital in North Italy (4000 employees). 100 applications

## Modeling Information System through DSM

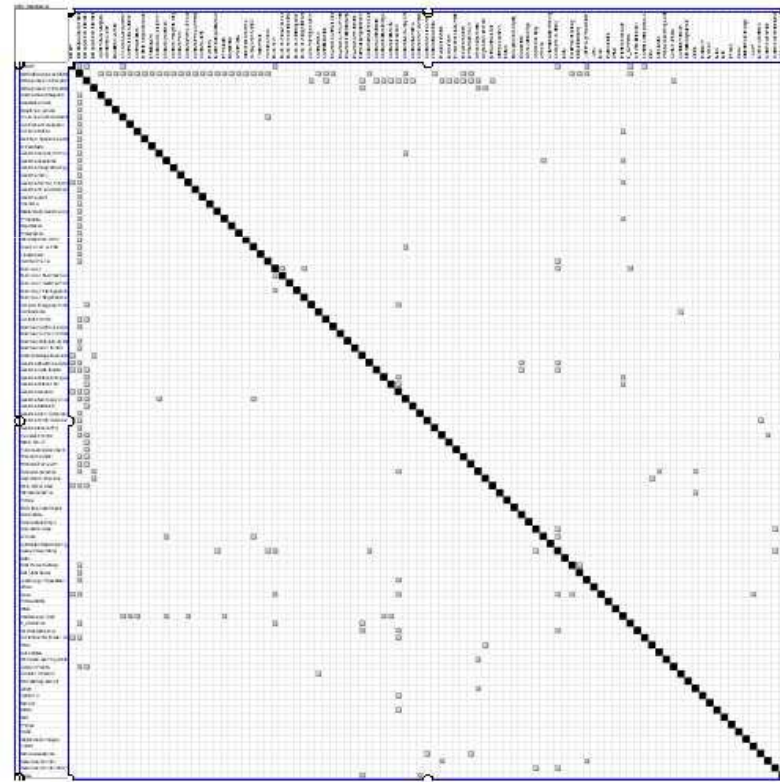
- Conceptual structure of the model
  - Components (software applications) and connectors
  - Classes of users
  - Views

A representation of a specific aspect of the architecture: components and connectors attributes for a class of users



# Modeling Information System through DSM

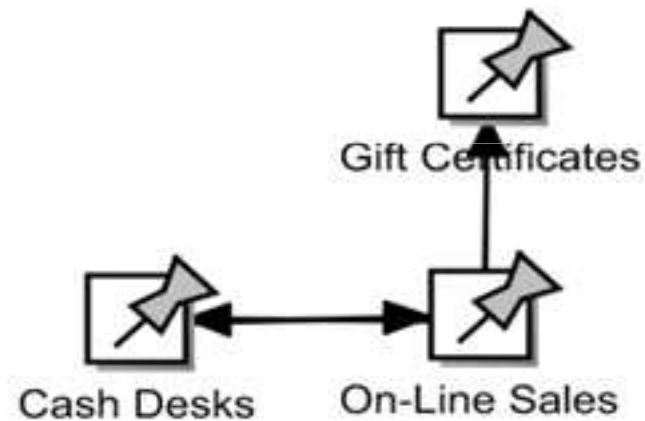
- Components and connectors - DSM
- Software applications and data flow relations between applications.



# Modeling Information System through DSM

- Design Structure Matrix - Example

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	On-Line Sales	Cash Desks	Gift Certificates
<input type="checkbox"/> On-Line Sales	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Cash Desks	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Gift Certificates	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>







## Modeling Information System through DSM

- Classes of users
  - Developers and system integrators - know the relations between applications
  - The architecture office - monitor the complexity, discover architectural smells, verify architectural rules
  - Business people - relationship between business processes and software applications

# Modeling Information System through DSM

- Views
  - Software applications and data flow relations between applications
  - Types of connectors between applications
  - Architectural smells
  - Mapping between business processes and software applications

## Modeling Information System through DSM

- Model quality
  - The information was derived manually from the documentation of software applications and knowledge of application owners
  - Automatically extract information from code is better, but hardly implementable in practice in an industrial context





# Query patterns

- Query patterns

- Pattern name.
- Classes of users.
- View and model attributes.
- Model query.
- Explanations and suggestions.
- Example.

# Query patterns

- Patterns

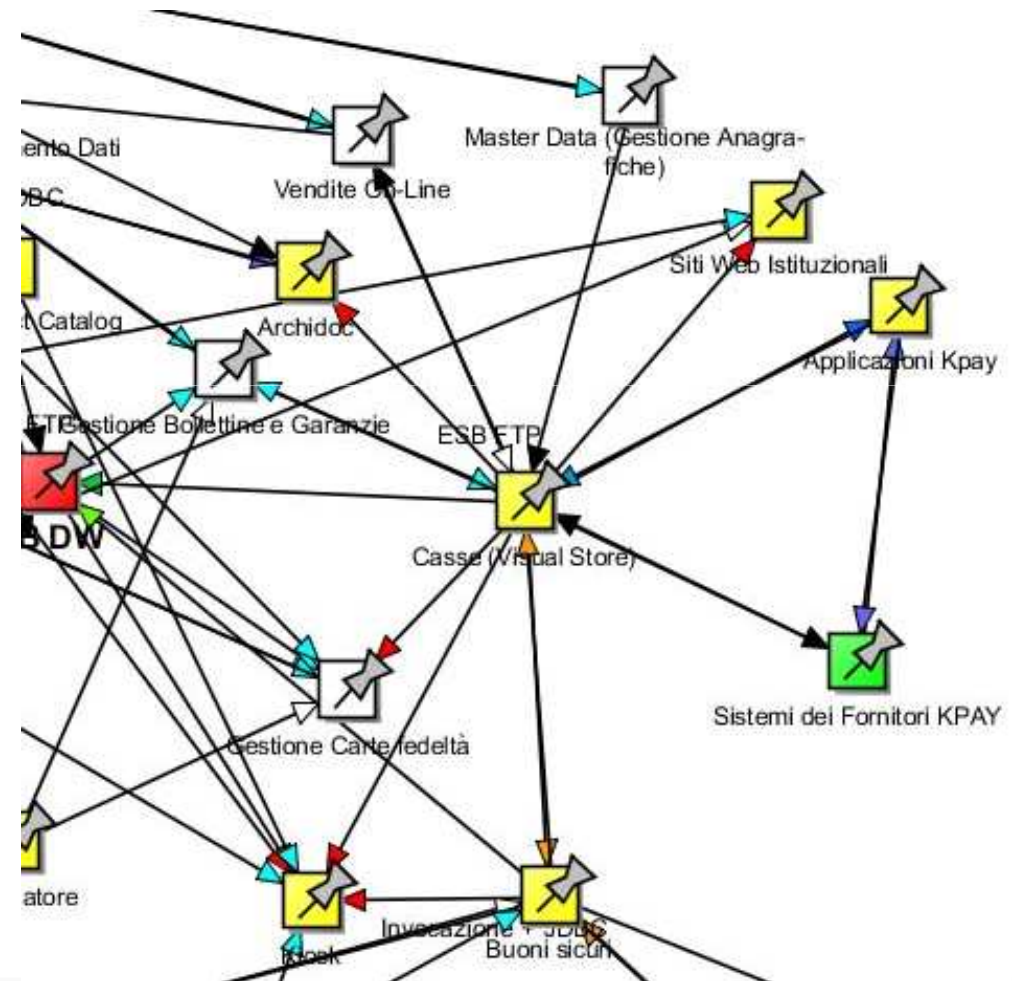
- Applications and data flow relations between applications.
- Databases and data flow relations with other applications.
- Applications and type of connectors between applications.
- Architectural smell: an application is connected to many other applications.
- Architectural smell: an application is connected through many types of connectors.
- Architectural smell: an application is connected to many databases.
- Architectural smell: a database of type data-warehouse has output connections with applications.
- Applications and connectors implementing the business process A, B,...

# Query patterns

## *System 1 – Smells*

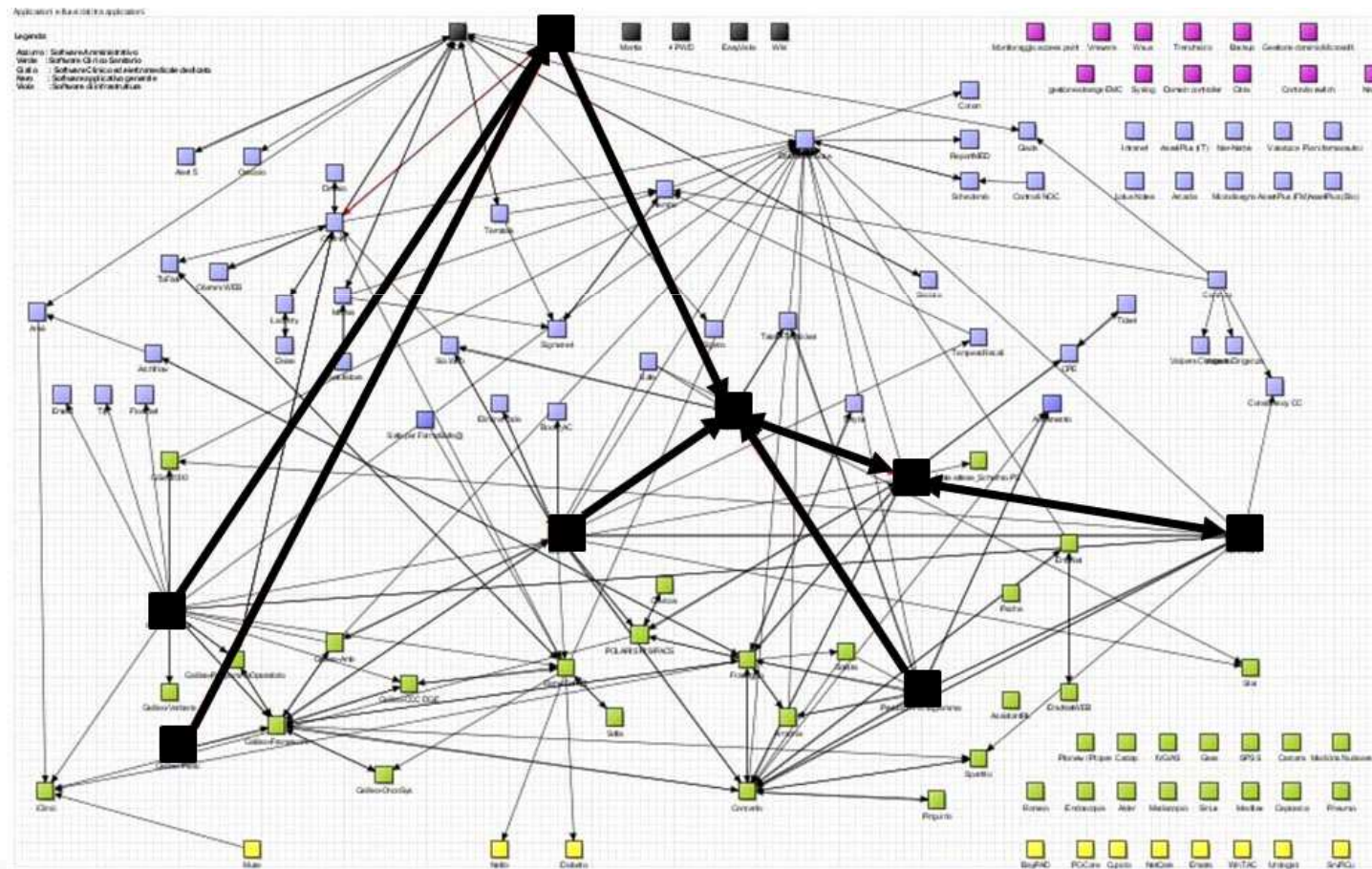
Application connected to many other applications.

Many types of connectors.



# Query patterns

## *System 2* - applications and connectors implementing a business process



## Discussion and outlook

- Finding quality information to develop the models is a difficult process
  - *System 1* - A wiki maintains the key information to support software evolution
  - *System 2* - The knowledge was collected through the interviews of the application owners.
- In both cases the management supported the project and is aware of the relevance of the issue

## Discussion and outlook

- An open problem
- The limitations of the class of tools we used:
  - They don't offer the full functionality of a database management system and flexible navigation of the graph
  - It is not easy to associate complex attributes to the elements of the matrix
  - The ability to query the graph is limited