



Computer Aided solutions to  
SEcure electroNic commerce E Transactions

## Towards a Business Process-Driven Framework for Security Engineering with the UML

José L. Vivas, HP Labs, Bristol, UK

José A. Montenegro, University of Málaga,  
Spain

Javier López, University of Malaga,  
Spain

## Security engineering: State of the Art

- Although security aspects are inherent in any modern software system, **there is very little systematic support for software engineers to produce secure software**
- Security policies are generally specified in terms of highly specialized security models that are not integrated with general software engineering models

**Security engineering must be treated as an integral part of the whole system development process**

- A security implementation that ignores the basic phases of systems engineering: requirements, analysis, design, implementation, maintenance – is bound to fail

## Security requirements engineering

- The most challenging task in security engineering is at the top level of system development: requirements engineering
- Security needs are typically articulated only as high level declarations by users and customers, corresponding to the business view of the system

Business view of security: anonymity, privacy...

Technical view of security: encryption, digital signature...

- Computer systems security must address not just the computer system, but the changing organizational context in which they are inserted
- Conventional requirements modelling cannot represent the organisational procedures that underpin a security policy



## Why security-critical systems fail? An example

### Failure of the Pentagon's OSD network

- During deployment of the OSD network, an emergency was declared after an attempt to implement a security design to meet a very strict set of security policies
- The cause was declared to be that the needs and requirements of both the users and the decision makers were not properly integrated into the security design and implementation
- The following specific observations were made:
  1. The original security hardware components were basically correct and well-designed.
  2. The implementation of the security policies to be enforced by the security hardware is where the system failure occurred.
- **Re-engineering of the system: an intensive effort was undertaken to determine user and system data and communication requirements**



## Requirements for cryptographic protocols

- The findings of formal analysis of cryptographic protocols are often controversial
- Many times the cause may be a misunderstanding of the requirements
- Understanding requirements might be as hard as showing that a protocol satisfies them
- Developers have to make decisions about how to employ cryptographic protocols whose behaviour and requirements are often obscure or hard to understand
- **Two challenges:**
  1. To derive requirements for cryptographic protocols from higher-level system requirements
  2. To derive requirements for protocol implementations from requirements for cryptographic protocols



## Business modelling

- A *business model* is an abstraction of how business functions
- It provides a simplified view of the structure of a business and a definition of the information systems requirements necessary to support the business
- Business modelling involves answering important questions for the security requirements of a system such as
  - How do different actors interact
  - What activities are part of their work
  - What are the ultimate goals of their work
  - What systems or resources are involved
  - What rules govern their activities and structures

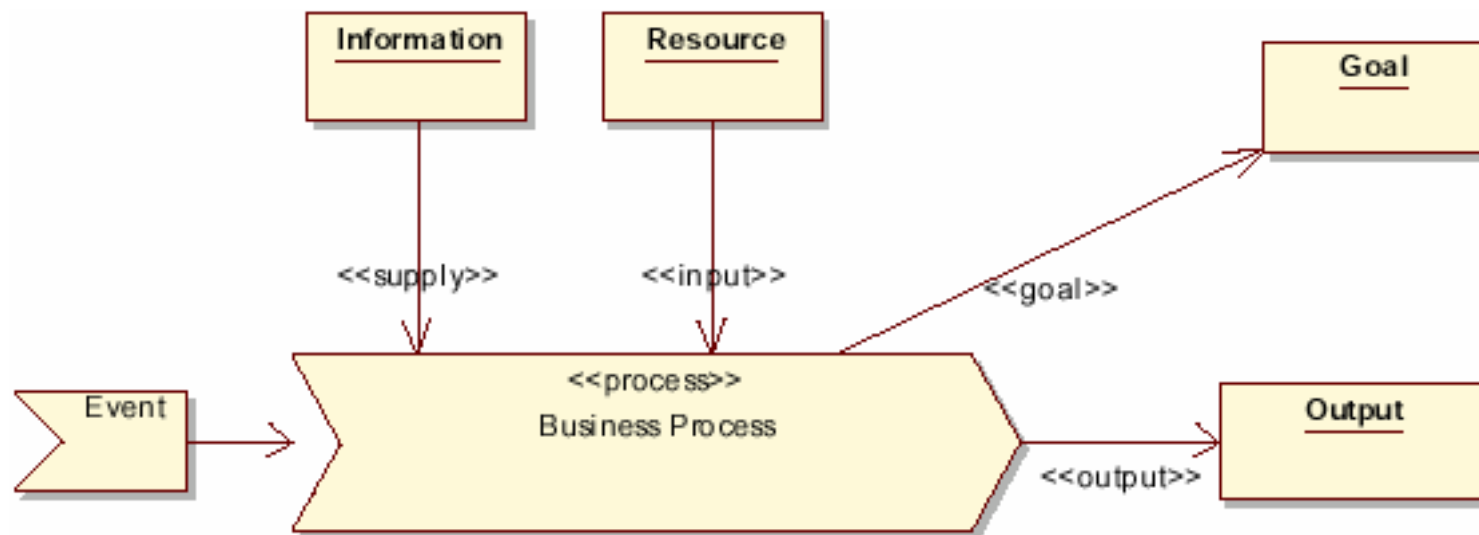
We need a model in which most of these notions can be expressed



## The Eriksson-Penker Model

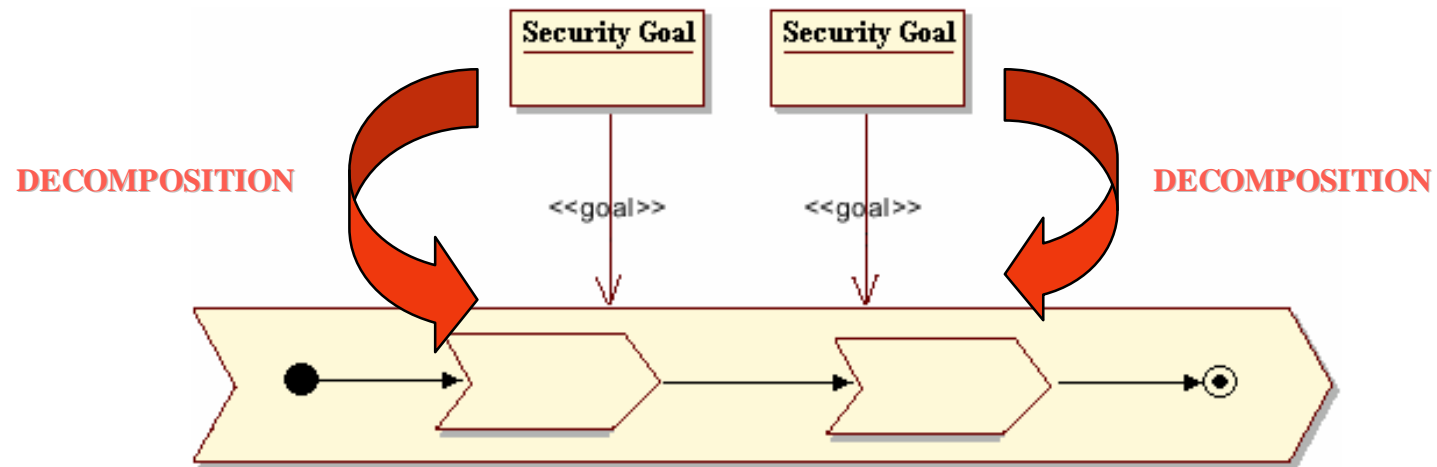
**The Eriksson-Penker business process model contains many of the features that are relevant for security**

A process is a specific ordering of work across time and place, with a beginning, an end, and clearly defined inputs and outputs



## Security-enriched process model

- The metamodel might be extended with e.g. security goals to accommodate security features
- Security goals must also be decomposed
- Traceability must be ensured under decomposition



## Towards a UML-based framework for security engineering

- Development of a UML-based framework for representing security semantics in an integrated model driven development environment
- Development of automated means to
  1. assist a developer to integrate the security requirements into the functional description of a process
  2. assist in mappings or transformations between models at different levels of abstraction
- The methodology includes
  1. translating UML, including extensions for security, into machine-readable notations such as XMI (XML Metadata Interchange)
  2. making extensive use of a repository of security patterns developed at different levels of abstraction
  3. translating UML diagrams into a formal notation for the purpose of validation, verification, simulation, and threat analysis

## Model Driven Architecture (MDA)

- Our methodology is consistent with the principles of the Model Driven Architecture (MDA), a standard approach to model-driven development
- MDA features three kinds of model
  1. CIM: Computer Independent Model (business model)
  2. PIM: Platform Independent Model (specification model)
  3. PSM: Platform Specific Model (implementation model)
- Business models (CIMs) are models of real-world objects and their behaviour
- In a business model we include only the interfaces of the software systems, i.e. the services they provide

## Computer Independent Model (CIM)

- The requirements for the system are modelled in a computation independent model
- A CIM is a model of a system that shows the system in an environment in which it will operate, and thus it helps in presenting exactly what the system is expected to do
- CIM is useful both as an aid in understanding a problem and as a source of a shared vocabulary for use in other models
- CIM requirements should be traceable to PIM and PSM constructs that implement them, and vice versa

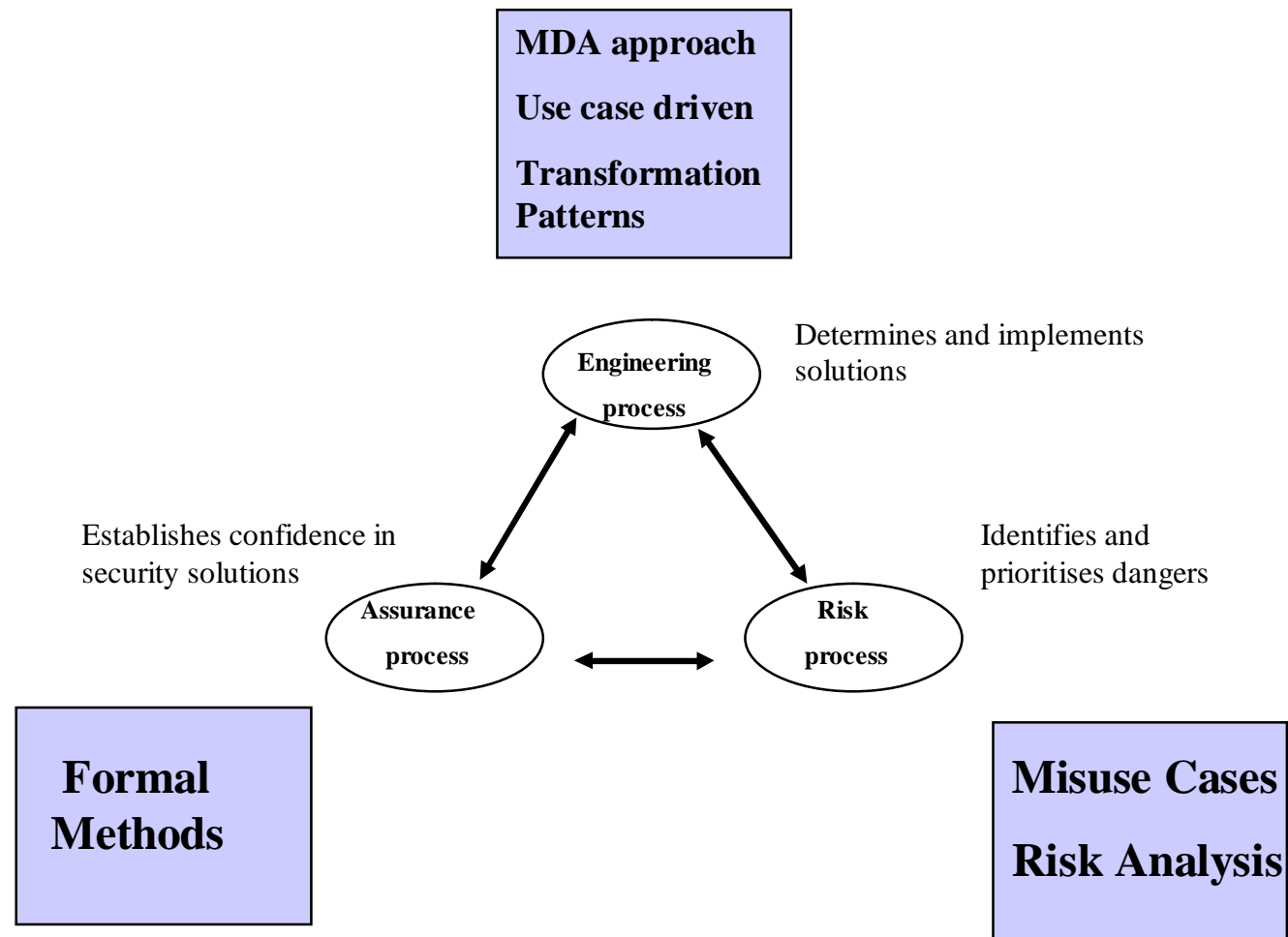
## Business Security Patterns

- Business Security patterns can be used to derive robust security solutions in a cost effective way: security by design
- Business security patterns can be used to identify the relationships between security goals and security solutions
- Patterns should be defined by the important aspects of security such as the **who, what, where, how, why** and **when**
- Business process models typically include many such features

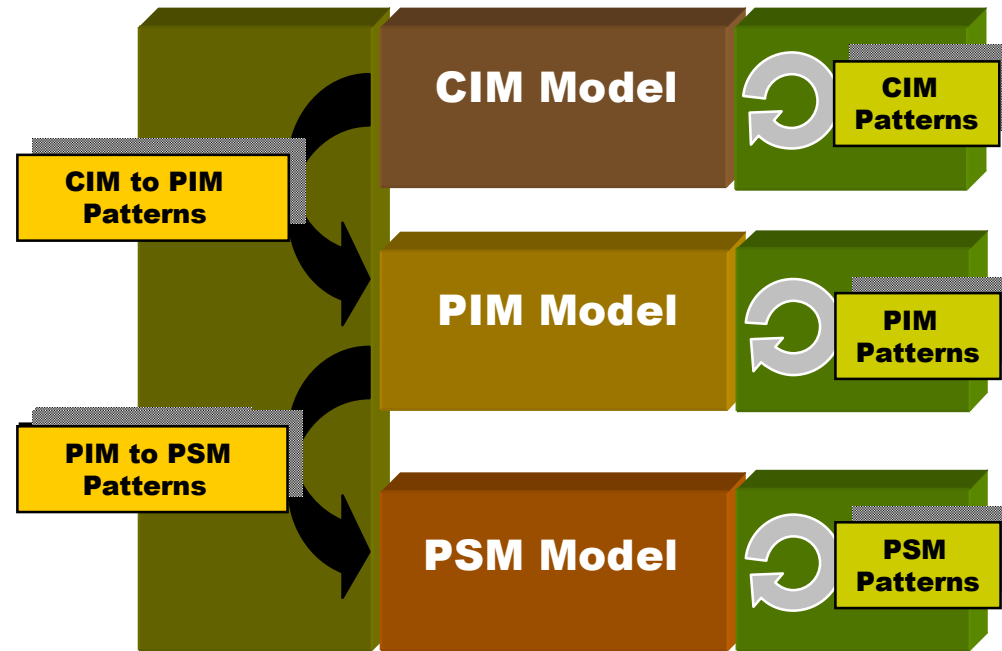
## Use case driven development

- Use cases and business process or workflows are two closely related concepts
- A use case is an interaction between a user of a system and the system itself
- Use cases are used to capture the behavioural requirements of software systems
- Security requirements can be viewed as *anti-functional*: they usually have a visible impact on the way a user interacts with a system
- Therefore, security requirements should be analyzed together with the functional requirements
- Even threat analysis is currently being performed in the context of use cases and scenarios, cf. the notions *misuse cases* and *abuse cases*

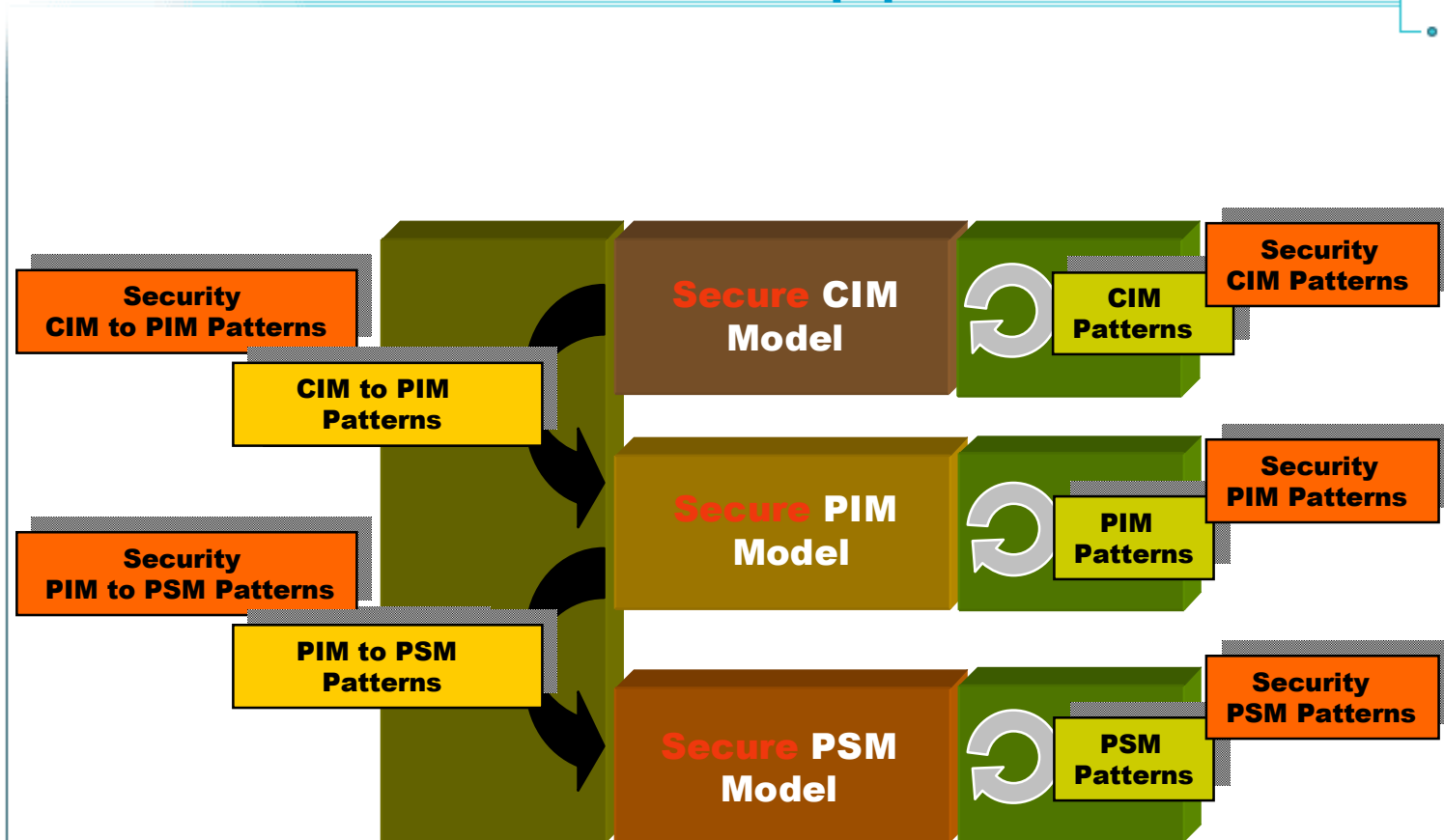
# Security Engineering in SSE-CMM



# MDA Approach



# Secure MDA Approach





Computer Aided solutions to  
SEcure electroNic commercE Transactions

Towards a  
Business Process-Driven  
Framework for Security  
Engineering with the UML