

Refinement Operators and Information Flow Security

A. Bossi, R. Focardi, C. Piazza, and S. Rossi

**Department of Computer Science
University Ca' Foscari of Venezia**

{bossi,focardi,piazza,srossi}@dsi.unive.it

SEFM 2003, September 22-27, 2003, Brisbane

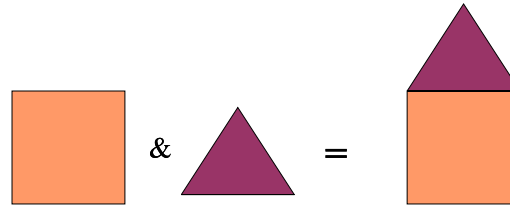
Protect Confidential Data in a Multilevel System

- ▷ **Information Flow Security** aims at guaranteeing that no **high** level (**confidential**) information is revealed to users at **low** level, even in the presence of any possible **malicious process**
- ▷ **Non-Interference** [Goguen-Meseguer'82]: **information does not flow** from high to low if the **high behavior** has **no effect** on what **low** level can **observe**
- ▷ **Development of Complex Systems**: it is important to **preserve the security** properties of interest during the **development steps**

Development of Complex Systems

The systematic development of complex systems usually relies on

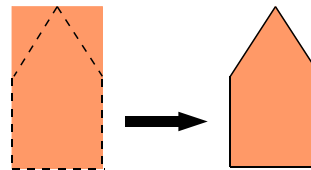
- ▷ **Composition**: building blocks are put together (e.g., parallel composition)



The composition of secure parts has to be secure as a whole

Compositional Non-Interference properties have been studied

- ▷ **Refinement**: abstract specifications are refined into more concrete ones

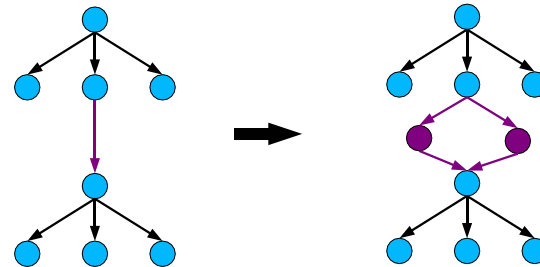


Non-Interference properties based on sets of execution sequences are

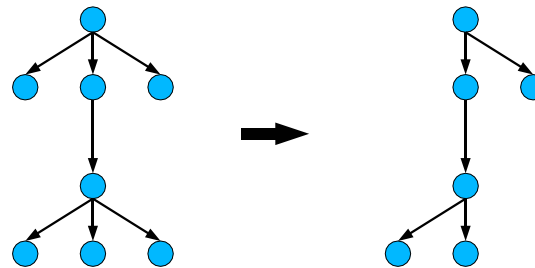
hard to preserve under refinement

Vertical and Horizontal Refinement

- ▷ **Vertical Refinement (Action Refinement)**: abstract-level primitives (actions) are **expanded** into concrete-level implementations (macros)



- ▷ **Horizontal Refinement**: abstract specifications are refined into more concrete ones by **choosing among the allowed behaviors** the ones that will be actually implemented



Refinement and Security

In our work we consider **Horizontal Refinement** and we

- ▷ introduce a new notion of **refinement** based on **simulation** instead of **trace inclusion**
- ▷ analyze under which conditions our notion of refinement **preserves information flow security** properties. In particular, we consider properties defined through **unwinding conditions**

Plan of the Talk

- ▷ The SPA language: syntax and semantics
- ▷ The notion of Refinement
- ▷ Refinements preserving Unwinding Security Properties
- ▷ The P_BNDC security property
- ▷ Other Security properties based on Weak Bisimulation and Trace Equivalence
- ▷ Conclusions

The SPA syntax

E	$::=$	$\mathbf{0}$	<i>empty process</i>
		Z	<i>constant</i>
		$a.E$	<i>prefix</i>
		$E + E$	<i>non-det. choice</i>
		$E \mid E$	<i>parallel composition</i>
		$E \setminus v$	<i>restriction</i>
		$E[f]$	<i>relabelling</i>

- ▷ each constant Z has to be associated to a definition $Z \stackrel{\text{def}}{=} E$
- ▷ H high actions and L low actions

The SPA semantics

Semantics given through transition relations

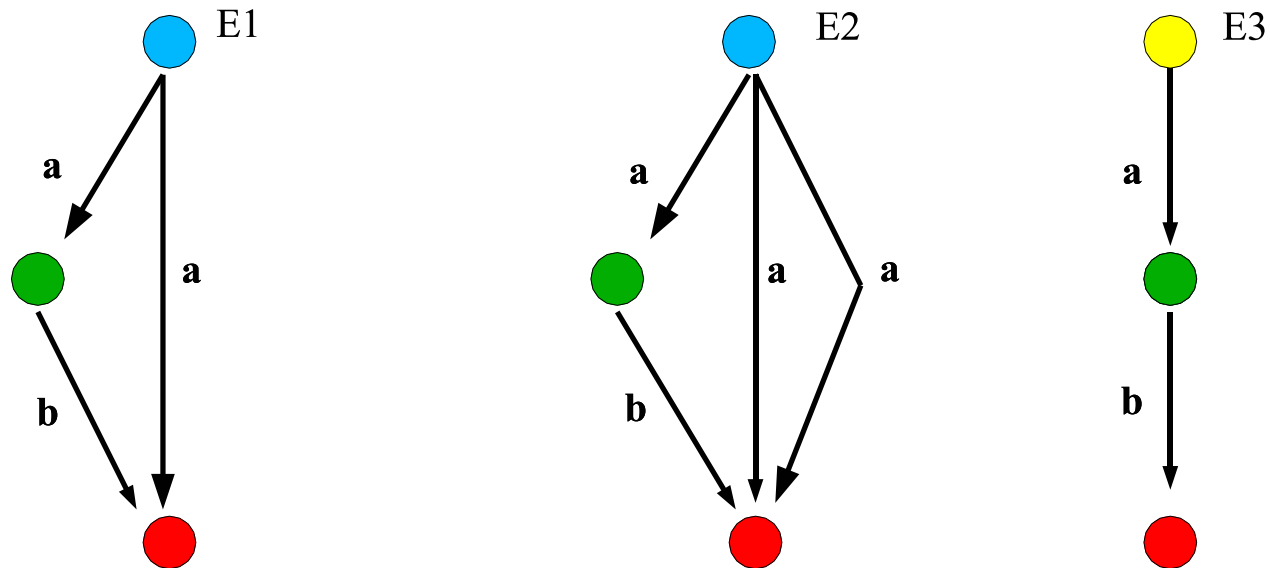
Input		Output	
	$a.E \xrightarrow{a} E$		$a.E \xrightarrow{\bar{a}} E$
Parallel	$E_1 \xrightarrow{a} E'_1$		$E_1 \xrightarrow{a} E'_1 \quad E_2 \xrightarrow{\bar{a}} E'_2$
	$E_1 E_2 \xrightarrow{a} E'_1 E_2$		$E_1 E_2 \xrightarrow{\tau} E'_1 E'_2$

Behavioral equivalences establish equalities among processes

Simulation and Bisimulation

▷ **Idea:** bisimulation is a mutual step-by-step simulation

▷ $E1 = a.b.0 + a.0$ $E2 = a.b.0 + a.0 + a.0$ $E3 = a.b.0$



▷ $E1$ and $E2$ are bisimilar and they both simulate $E3$

Simulation and Strong Bisimulation

Simulation

$\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ over SPA processes such that if $(E, F) \in \mathcal{S}$ then:

$E \xrightarrow{a} E'$ implies $F \xrightarrow{a} F'$ and $(E', F') \in \mathcal{S}$

$E \leq F$ if there exists a simulation \mathcal{S} containing (E, F)

Strong Bisimulation

$\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ such that \mathcal{S} is a symmetric simulation

$E \sim_B F$ if there exists a bisimulation \mathcal{S} containing (E, F)

Strong bisimulation can be used as behavioral equivalence in the semantics

Refinement Operators - Intuition

A refined specification should never show behaviors that were not foreseen in the initial specification

- ▷ each **abstract** state is refined into **at most one concrete** state
- ▷ the **abstract** state **simulates its refinement**, i.e., if the refinement E of F performs an action a reaching E' , then F can perform a reaching F' whose refinement is E'

Refinement Operators - Formalization

Refinement

$\mathcal{R} \subseteq \mathcal{E} \times \mathcal{E}$ over SPA processes such that:

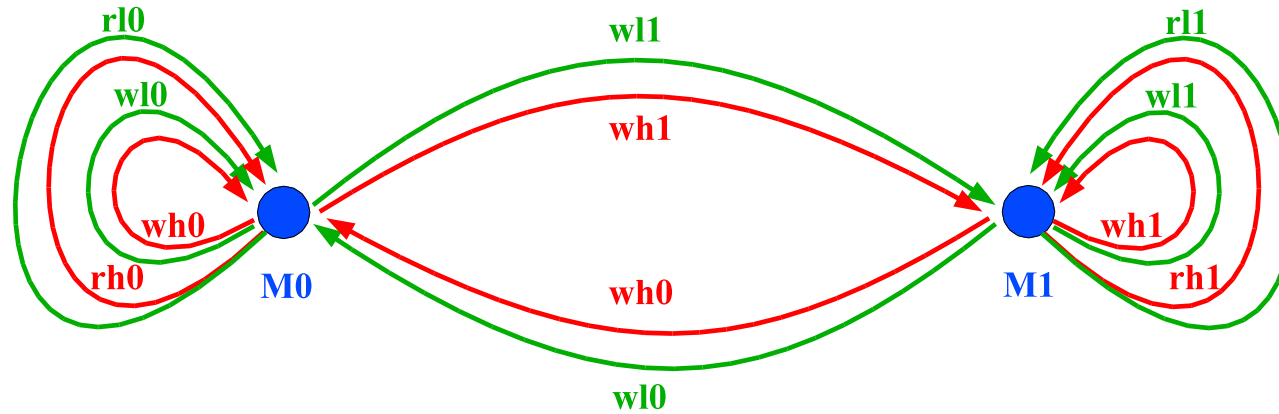
\mathcal{R} is a **partial function** from \mathcal{E} to \mathcal{E}

\mathcal{R}^{-1} is a **simulation**

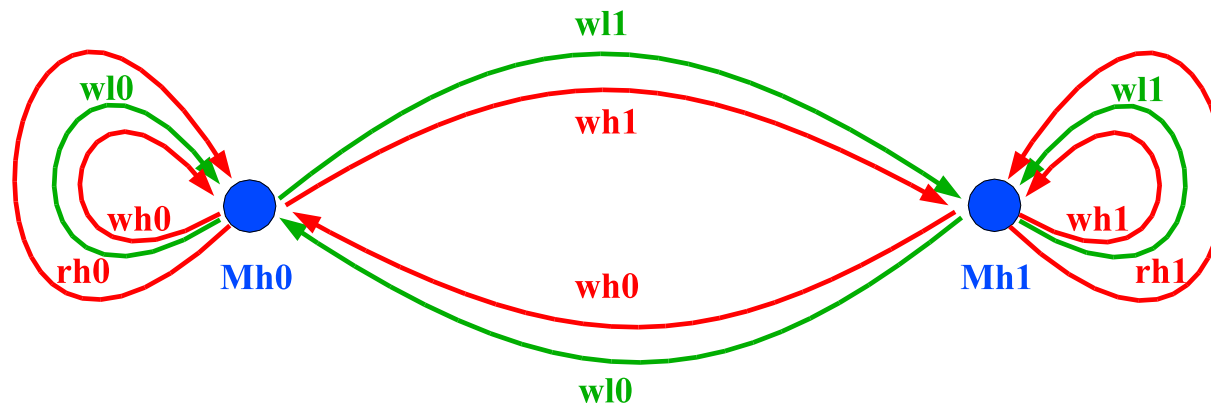
$E \preceq F$, i.e., E is a **refinement** of F , if there exists a refinement \mathcal{R} such that $\mathcal{R}(F) = E$

Example

Consider a binary memory cell



We refine it into a **high level cell** by imposing **no read up**

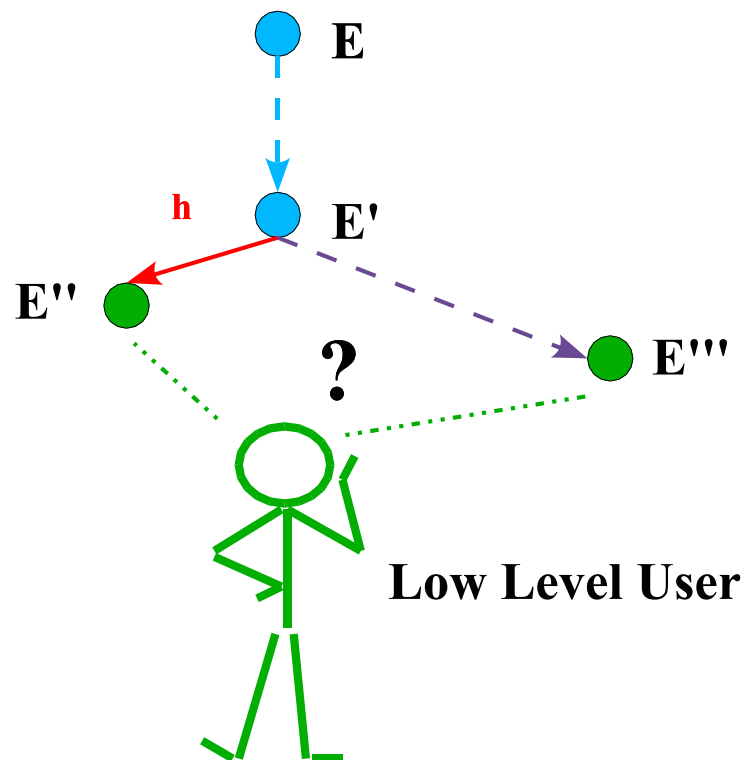


Properties of the Refinements

- ▷ **Composition of Refinements:** if \mathcal{R}_1 and \mathcal{R}_2 are refinements, then $\mathcal{R}_1 \circ \mathcal{R}_2$ is a refinement
- ▷ **Refinement and Reachability:** if $\mathcal{R}(F) = E$, $\mathcal{R} \cap (\text{Reach}(F) \times \text{Reach}(E))$ is a refinement
- ▷ **Mutual Refinement:** if F is finite state and $F \preceq E \preceq F$, $F \sim_B E$
- ▷ **Compositionality of Refinement:** if $\mathcal{R}(F) = E$ and $\mathcal{R}(G) = I$,
 - ▷ $a.E \preceq a.F$, if $a.F \notin \text{Reach}(F)$
 - ▷ $E + I \preceq F + G$, if $F + G \notin \text{Reach}(F) \cup \text{Reach}(G)$
 - ▷ $E|I \preceq F|G$, $E \setminus v \preceq F \setminus v$, $E[f] \preceq F[f]$

Security as Unwinding - Intuition

If the **high** level user can perform h reaching E'' from E' , then also E''' is **reachable** from E' and E'' and E''' are undistinguishable for the **low** level user



Many security properties are instances of this scheme: **P_BNDC**, **SBNDC**, **CP_BNDC**, **SNDC**

Security as Unwinding - Formalization

Let \sim^l be a low level observational equivalence

Let \dashrightarrow be a reachability relation

Generalized Unwinding

$$\mathcal{W}(\sim^l, \dashrightarrow) = \{E \in \mathcal{E} \mid \forall F, G \in \text{Reach}(E), \text{ if } F \xrightarrow{h} G \text{ then} \\ \exists G' \text{ such that } F \dashrightarrow G' \text{ and } G \sim^l G'\}$$

Refinements preserving Unwinding

We say that a refinement \mathcal{R} **preserves** a relation \odot iff

$$G \odot G' \text{ implies } (\mathcal{R}(G) \uparrow \text{ and } \mathcal{R}(G') \uparrow) \text{ or } (\mathcal{R}(G) \odot \mathcal{R}(G'))$$

Unwinding Theorem

Let \mathcal{R} be a refinement preserving \sim^l and \dashrightarrow such that $\mathcal{R}(F) \downarrow$

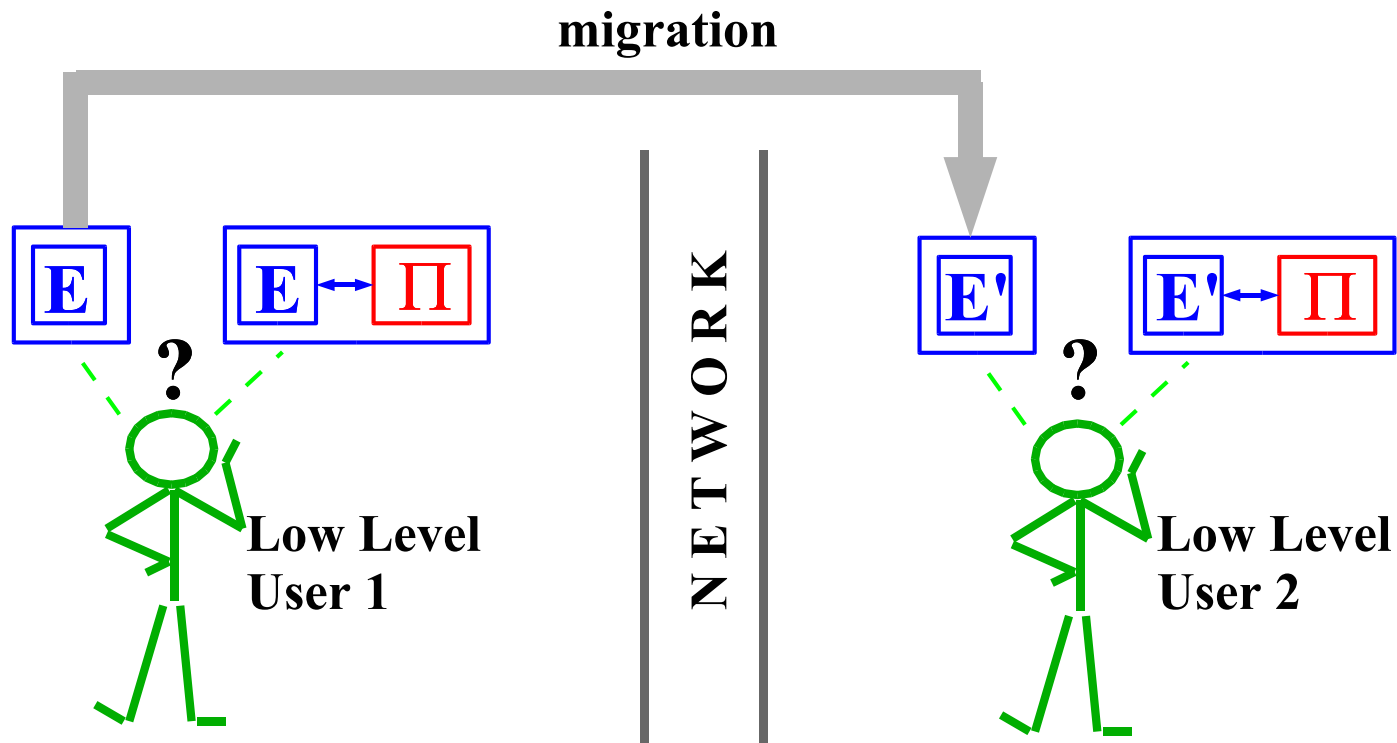
$$F \in \mathcal{W}(\sim^l, \dashrightarrow) \text{ implies } \mathcal{R}(F) \in \mathcal{W}(\sim^l, \dashrightarrow)$$

Composition Theorem

If \mathcal{R}_1 and \mathcal{R}_2 preserve \odot , then $\mathcal{R}_1 \circ \mathcal{R}_2$ preserves \odot

The P_BNDC property

Aim: check all the states reachable by the system against all high level (potentially malicious) processes



Persistent BNDC: $\forall E' \text{ reachable from } E, \forall \Pi \in \mathcal{E}_H E' \approx_B^l E' | \Pi$

P_BNDC and Unwinding

Weak Bisimulation on Low Actions

$\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ such that if $(E, F) \in \mathcal{S}$ then for all $l \in L \cup \{\tau\}$:

$E \xrightarrow{l} E'$ implies $F \xrightarrow{\hat{l}} F'$ and $(E', F') \in \mathcal{S}$

$F \xrightarrow{l} F'$ implies $E \xrightarrow{\hat{l}} E'$ and $(E', F') \in \mathcal{S}$

$E \approx_B^l F$ if $(E, F) \in \mathcal{S}$ weak bisimulation on low actions

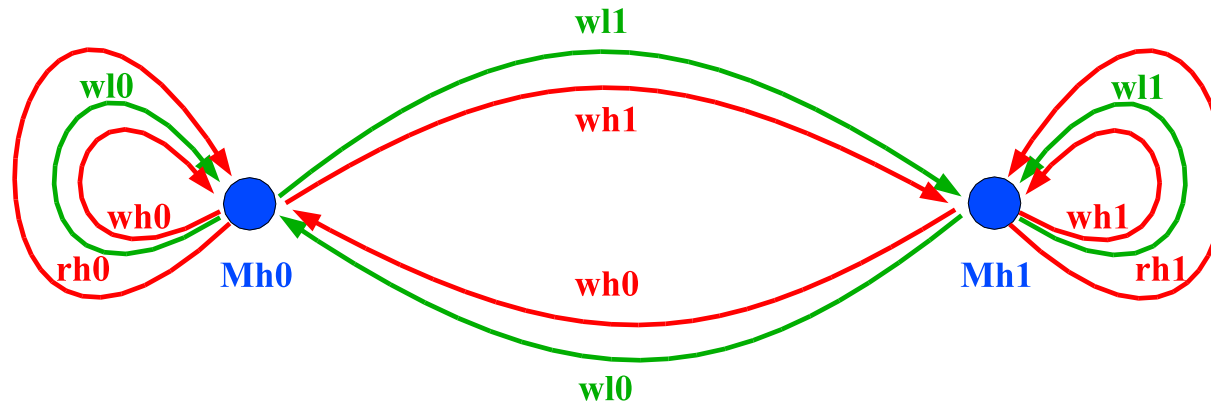
Silent Reachability

$E \xrightarrow{\hat{\tau}} F$ if E reaches F with a sequence of τ actions.

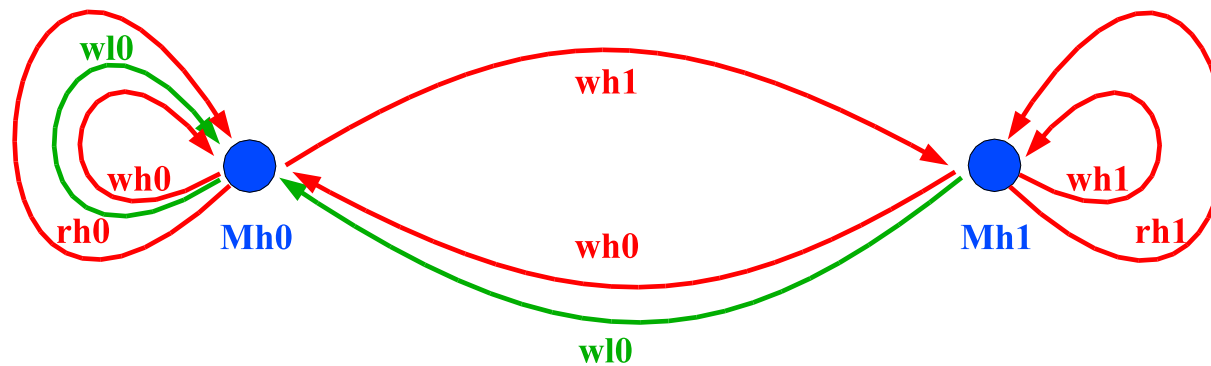
$E \in \text{P_BNDC}$ if and only if $E \in \mathcal{W}(\approx_B^l, \xrightarrow{\hat{\tau}})$

Example

Consider again the **high level memory cell**. It is **P_BNDC**



We refine it: the **low level user** can only write **0**. It remains **P_BNDC**



Other Security Properties

We obtain similar results on:

- ▷ **SBNDC**: which corresponds to $\mathcal{W}(\approx_B^l, \equiv)$
- ▷ **CP_BNDC**: which corresponds to $\mathcal{W}(\approx_B^l, \xrightarrow{\tau})$
- ▷ **SNDC**: which corresponds to $\mathcal{W}(\approx_T^l, \equiv)$

From **SNDC** we get a sufficient condition for **PSP**

Conclusions

- ▷ We introduce a new notion of **Refinement** based on **Simulation**
- ▷ Refinements **can be composed** and are **compositional** w.r.t. the **SPA operators**
- ▷ Refinements which **preserve Unwinding Conditions** have been **characterized**
- ▷ We apply our results on **Security Properties** based on both **Bisimulation** and **Trace Equivalence**

Related Works

- ▷ [Mantel 2001](#): Deterministic event systems are considered. Conditions for the preservation of trace-based properties are presented
- ▷ [Lowe 2002](#): [NDC](#) is not preserved under [CSP](#) refinements. A system is secure iff all its refinements are secure
- ▷ [Jacob 1989](#): A method to refine unsecure systems into secure ones is described
- ▷ [Roscoe et al 1994](#): A notion of non-interference which is preserved under [CSP](#) refinements is introduced in the case of low level deterministic processes