

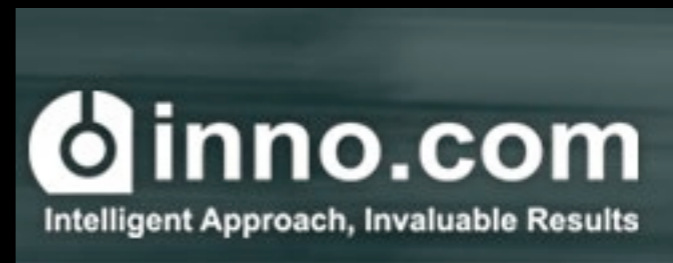
# Verifying the design of an outsourced COBOL system with IntensiVE

Andy Kellens  
Carlos Noguera  
Theo D'Hondt

Luc Jorissen  
Bart Van Passel



Software Languages Lab  
Vrije Universiteit Brussel



inno.com

# Context

- ▶ **Academic-industrial collaboration**
- ▶ **Large system for Belgian bank**
- ▶ **Started development in 2005; currently in production**
- ▶ **COBOL**



**Old language**



**Modern design:**

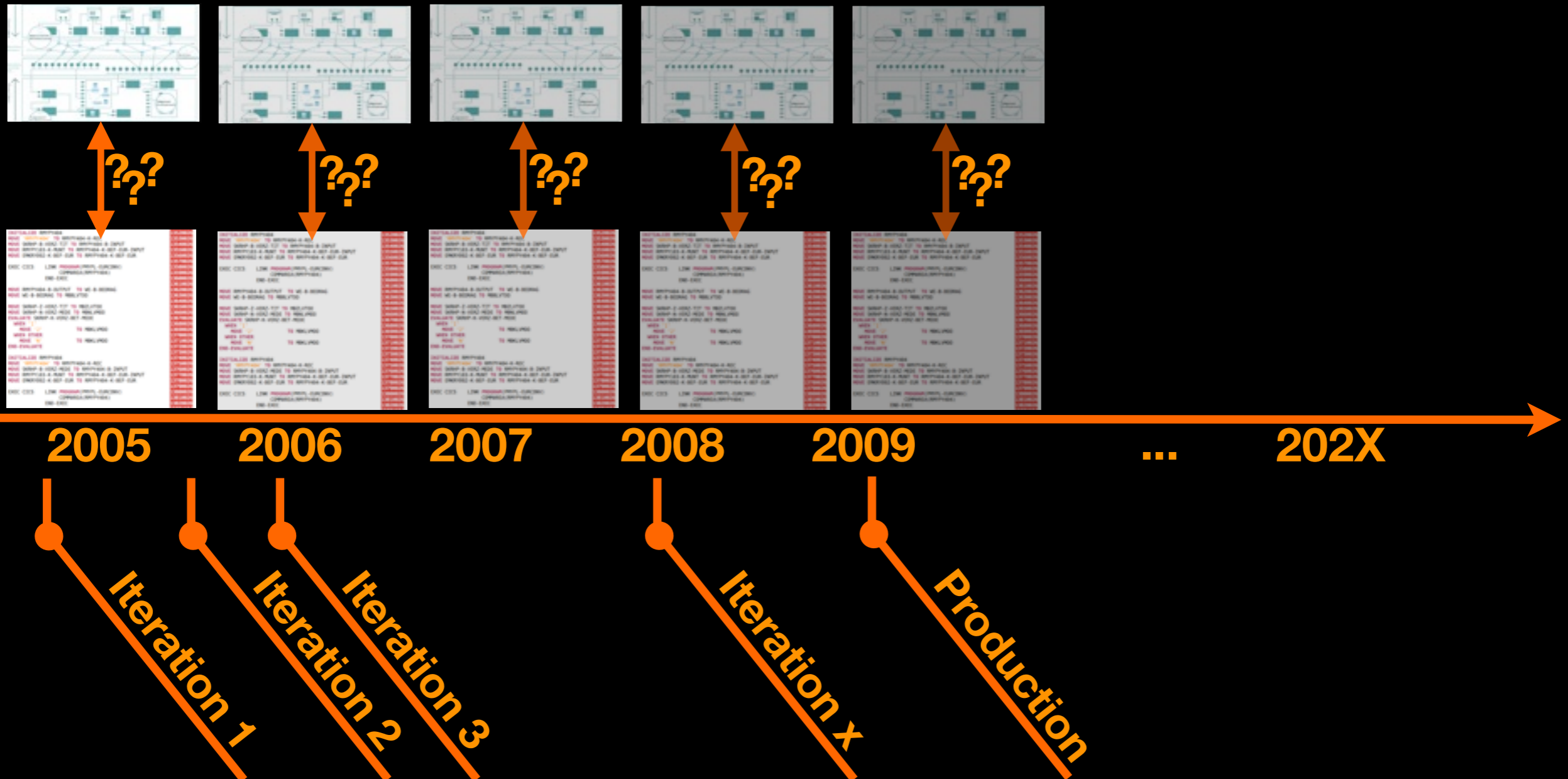
- ▶ **Components**
- ▶ **Services**



# Problem

- ▶ Large investment
- ▶ Outsourcing

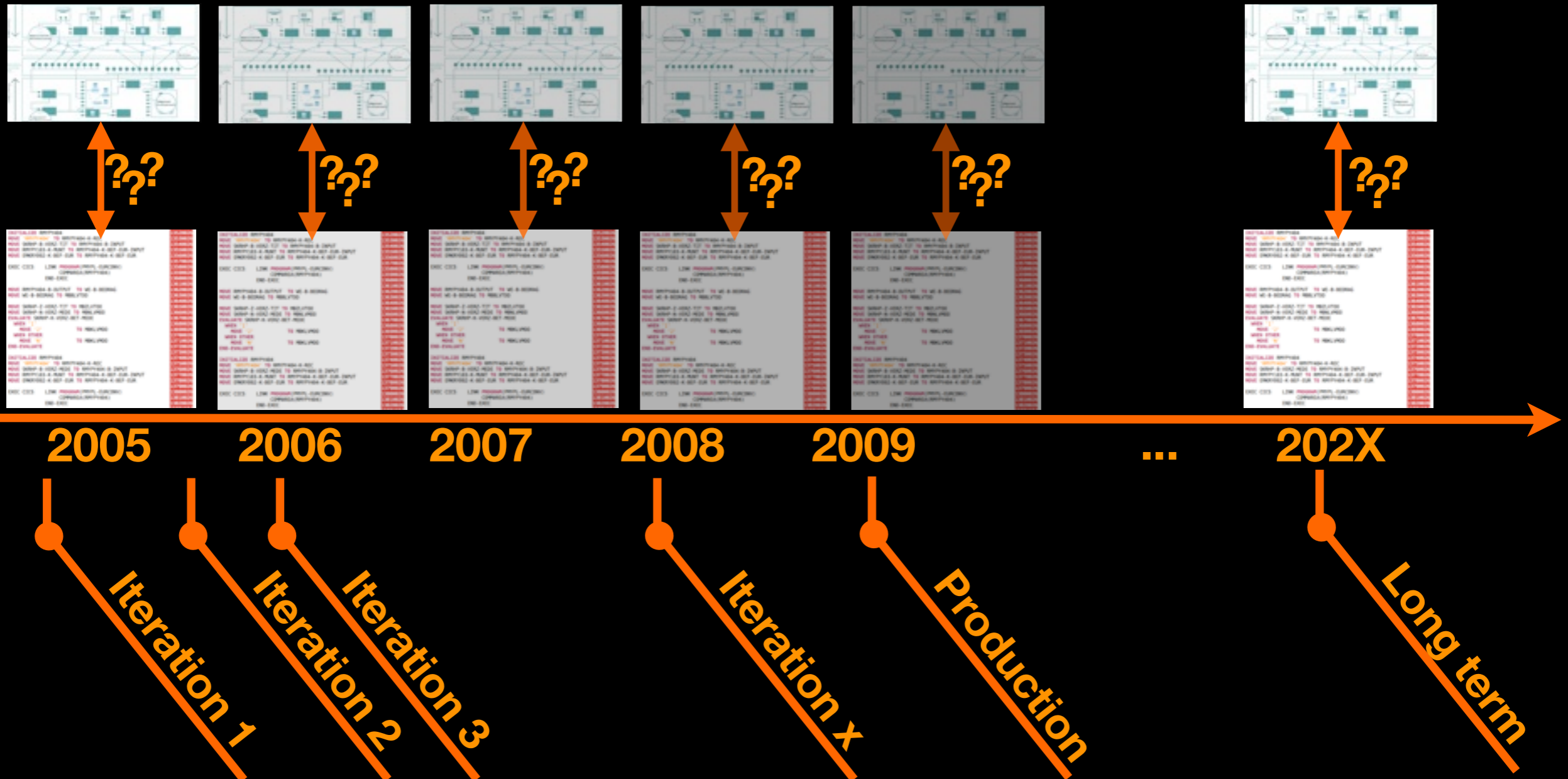
## Prevent drift



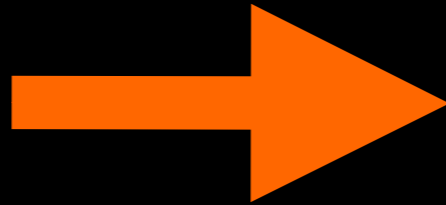
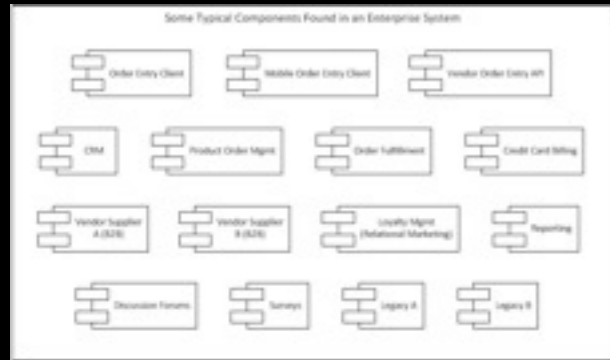
# Problem

- ▶ Large investment
- ▶ Outsourcing

## Prevent drift



# Mapping OO concepts to COBOL



```
PROGRAM IDENTIFICATION DIVISION.  
PROGRAM NAME .  
AUTHOR .  
DATE WRITTEN .  
REVISIONS .  
PERFORM USE CASE  
    INVOKE SUB-OPERATION 1  
    INVOKE SUB-OPERATION 2  
    INVOKE SUB-OPERATION 3  
END PROGRAM
```

▶ Programs

▶ Sections

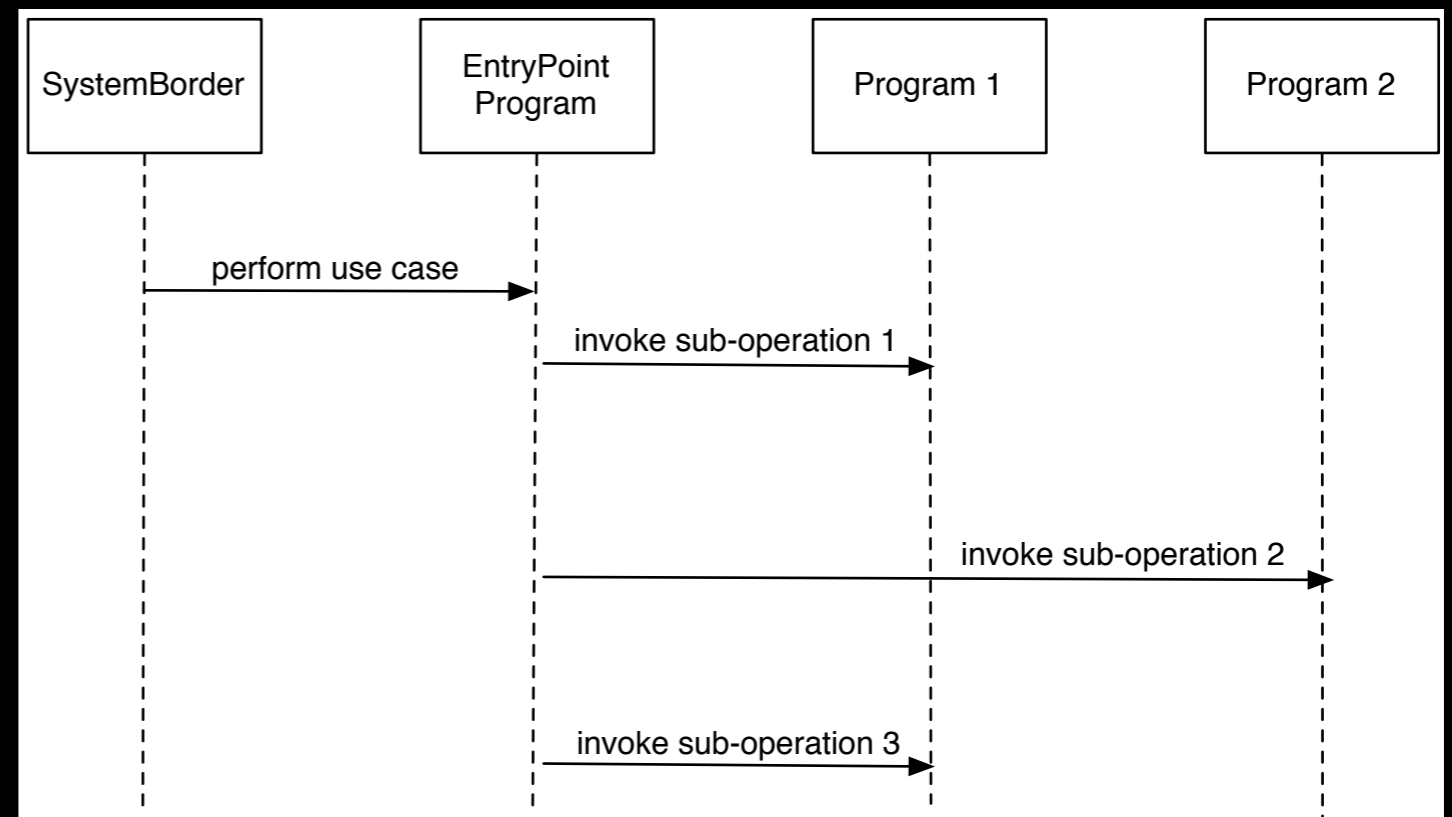
▶ Paragraphs

▶ Statements

## Implementation guidelines

- Patterns
- Conventions
- Idioms
- Naming schemes
- Layering

## Sequence diagrams

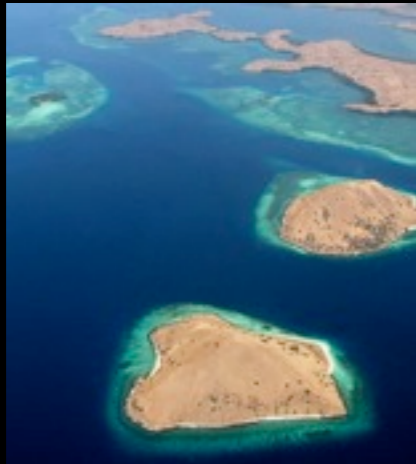






# IntensiVE for COBOL

## Island-based parsing



## Static analyses



## Predicate library

?program isProgramWithIdentifier: ?id  
?program callsProgram: ?callee  
?program writesToTable: ?table  
?section sectionInProgram: ?program  
?section sectionPerforms: ?section  
?section sectionWithName: ?name  
?program usesCopybook: ?copybook

# Example: layering naming convention

“A section can *only* perform sections that have section name with the *same or later beginning letter*”

## Sections with callees

if ?caller sectionPerformsSection: ?callee



Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)

# Example: layering naming convention

“A section can only perform sections that have

Sections

if ?caller section

program -> Program PMKTL385	
section -> Section B0204-ADDRESS-OF-NOTARY	
callee -> Section E9000-INIT-ABEND-VELDEN	
program -> Program PMKTC372	
section -> Section B0002-COMM-CHECK	
callee -> Section E9000-INIT-ABEND-VELDEN	
program -> Program PMKTW378	
section -> Section B0031-DECR-DATE	
callee -> Section E9000-INIT-ABEND-VELDEN	
program -> Program PMKTD374	
section -> Section B0001-GET-SYS-DATE-TIME	
callee -> Section E9000-INIT-ABEND-VELDEN	
program -> Program PMKTL339	
section -> Section B0202-CHK-AMT-TYP-ACTN	
callee -> Section S9001-FUNC-ABEND	
program -> Program PMKTR003	
section -> Section E9001-LOG-ABEND	
callee -> Section S9002-FUNC-LINK-CSMT	
program -> Program PMKTD381	
section -> Section B0028-GET-AMOUNT-DETAILS	
callee -> Section E9001-LOG-ABEND	
program -> Program PMKTC352	
section -> Section B2100-MQS-PROCESS-LAYOUT-03	
callee -> Section E9000-INIT-ABEND-VELDEN	
program -> Program PMKTL345	
section -> Section B0203-GET-CREDIT-START-DATE	
callee -> Section E9000-INIT-ABEND-VELDEN	
program -> Program PMKTR638	
section -> Section D1901-CLOSE-TC0022	
callee -> Section E9001-LOG-ABEND	
program -> Program PMKTR632	
section -> Section B0101-GET-ALL-PRVLDCHG	
callee -> Section D0103-FETCH-TC0002	
program -> Program PMKTR537	
section -> Section B0202-GET-THEOCAP-BAL	
callee -> Section S0000-LINK-PMKTR535	
program -> Program PMKTC391	
section -> Section B2100-MQS-PROCESS-LAYOUT-27	
callee -> Section E9001-LOG-ABEND	
program -> Program PMKTL330	
section -> Section B0202-CHECK-CREDIT-VERSION	



Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)

# Example: layering naming convention

“A section can *only* perform sections that have section name with the *same or later beginning letter*”

## Sections with callees

if ?caller sectionPerformsSection: ?callee



Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)

# Example: layering naming convention

“A section can *only* perform sections that have section name with the *same or later beginning letter*”

## Sections with callees

if ?caller sectionPerformsSection: ?callee

∀ ?invocation ∈ *Sections with callees*:

?invocation.caller isSectionWithName: ?callerName,  
?invocation.callee isSectionWithName: ?calleeName,  
[?callerName ≤ ?calleeName]



Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)

# Example: layering naming convention

“A section can *only* perform sections that have section name with the same or later beginning letter”

Verification failed

File Edit View Services Help

If a section performs another section, this section should have a first letter that comes later in the alphabet than the calling section.

Domain: 21401 out of 21653 (98.8362%) Range: 0 out of 0 (0%)

If a section performs another section, this section should have a first letter that comes later in the alphabet than the calling section.

program -> Program PMKTR571  
section -> Section B1002-INSERT-ACCNT-STAT  
callee -> Section S0000-LINK-PMKTR671

program -> Program PMKTR889  
section -> Section B0510-GET-DET-FOR-ASSC  
callee -> Section S0003-LINK-PMKTR447

program -> Program PMKTR542  
section -> Section B2401-RET-CR-DATA-PMHAL048  
callee -> Section P0010-SET-MOVEMENT

program -> Program PMKTL331  
section -> Section B0301-PROCESS-OK  
callee -> Section B0313-SRV-038-REG-CRED-MGMT

program -> Program PMKTC391  
section -> Section B2100-MQS-PROCESS-LAYOUT-50

Not in domain

Not in range

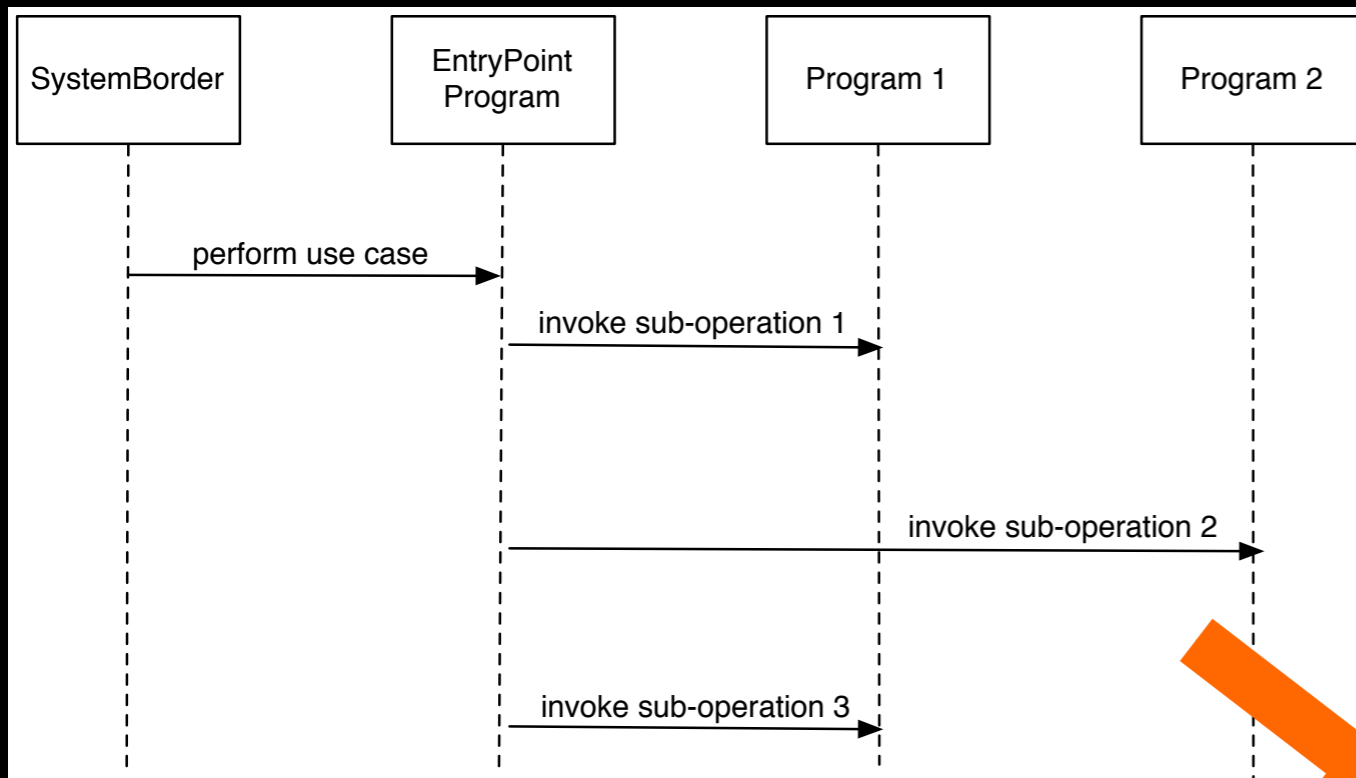
program -> Program PMKTR607, section -> Section D1201-FETCH-TC0001, callee -> Section B9999-ARR-LIMIT-ABEND  
program -> Program PMKTR635, section -> Section D1301-FETCH-TC0011, callee -> Section B1302-GET-AMT-DET-WITHOUT-AD  
program -> Program PMKTR484, section -> Section F0001-LAY02-OST-0019-UTILITY, callee -> Section E9001-LOG-ABEND  
program -> Program PMKTR889, section -> Section D0501-GET-KT-HEADER, callee -> Section B0502-GET-AND-LOCK-CREDIT  
program -> Program PMKTM377, section -> Section D0009-LOG-BTCH-END-INST, callee -> Section B0007-CALL-BATCH-UTILITY  
program -> Program PMKTR507, section -> Section P0000-CHK-FTR-IN-PACK, callee -> Section E9001-LOG-ABEND  
program -> Program PMKTR635, section -> Section D0702-FETCH-TC0020, callee -> Section B0702-GET-LST-REGSTRD-ADJ  
program -> Program PMKTL315, section -> Section P0000-GET-CREDIT-NO, callee -> Section E9000-INIT-ABEND-VELDEN  
program -> Program PMKTW378, section -> Section D0008-LOGICAL-END-BTCH-INST, callee -> Section B0007-CALL-BATCH-UTI



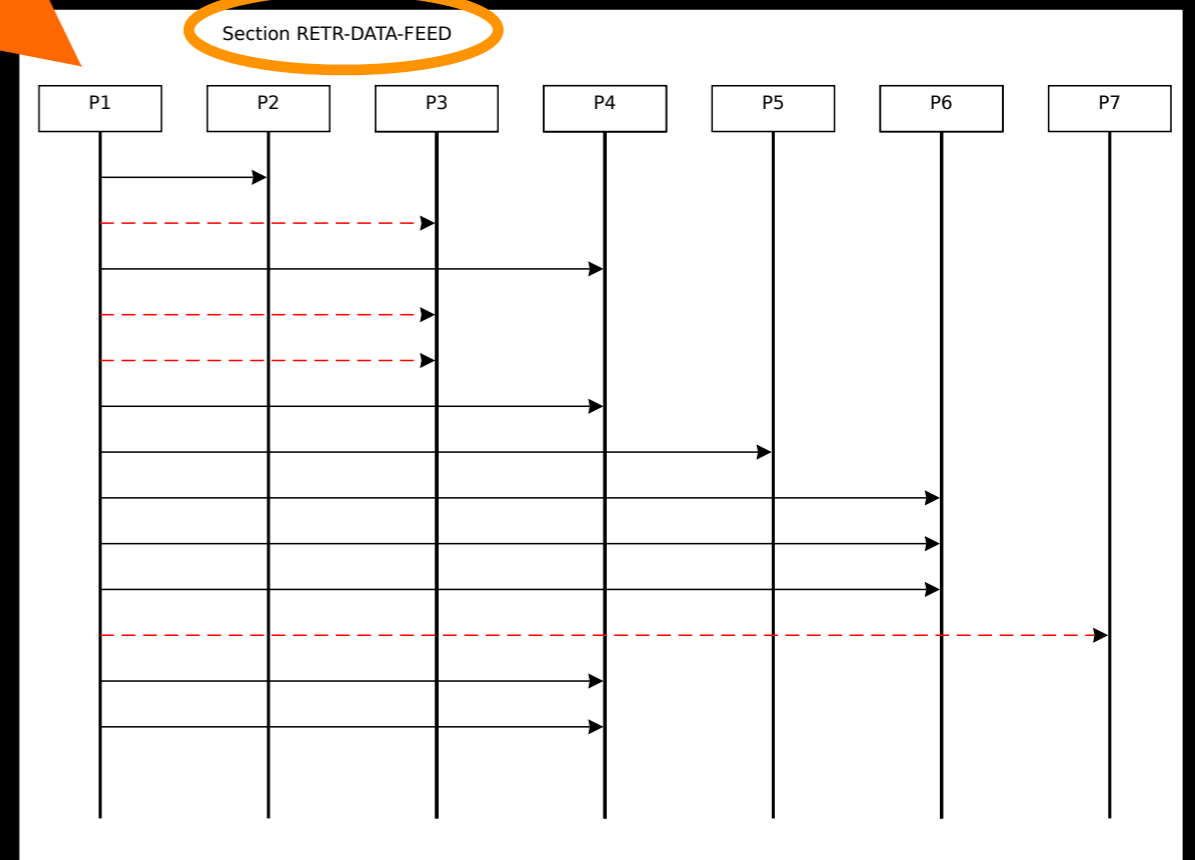
Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)



# Verifying sequence diagrams (2)



Tool identifies best match



Dedicated tool

# Experimental setup

Version	LOC	#diagrams	parsing (sec)	analysis (sec)	verification (sec)	total (sec)
Version 1	548 560	895	105	150	192	447
Version 2	665 220	917	134	228	204	566
Version 3	1 053 381	1366	413	479	288	1180

- ▶ **3 versions**
- ▶ **Increase in size**
- ▶ **Scalable tool**

# Results

Version	#diagrams	mappable	unmappable	% mappable	consistent	inconsistent	% inconsistencies
Version 1	895	408	487	45.59%	326	82	25.15%
Version 2	917	476	441	51.90%	386	90	23.33%
Version 3	1366	763	603	55.85%	637	126	19.78%

## ▶ Half mappable:

- Mismatch documentation/implementation
- Informal parts of documentation

## ▶ Over time, less inconsistencies

## ▶ Inconsistencies:

- Diagrams without implementation
- Order of calls
- Calls missing

# Lessons learned

## Academic

### ▶ Required pragmatic solutions:

- Island-based parsing
- Customized tool
- Reporting vs. integrated tool

## Industrial

### ▶ Quality control when outsourcing development

- Early-on detection of conformance problems
- Outsourcing firms: demonstrate quality guarantees

# Conclusions

- ▶ **Outsourced COBOL system**
- ▶ **Using IntensiVE tool**
  - dedicated support for COBOL
- ▶ **Verify mapping of design onto COBOL code**
- ▶ **Verify high-level design documentation**

# Verifying the design of an outsourced COBOL system with IntensiVE

<http://soft.vub.ac.be>

<http://www.intensional.be>

<http://www.inno.com>