

Combining Conceptual and Domain-Based Couplings to Detect Database and Code Dependencies

*Malcom Gethers, Amir Aryani,
Denys Poshyvanyk*



RMIT University, Australia



Common problems in software maintenance

Systems with legacy code, e.g., COBOL

Hybrid systems, e.g., Python and Java

Multi-tier systems

Inaccessible maintenance history

Despite the issues...

Perform impact analysis

Information suitable for
domain experts

User Interface
Components (UIC)



How?

Domain-based coupling

Conceptual coupling

Combination



Motivations

Domain-based approach works without access to source code or design documents

Conceptual coupling approach is language independent

The approaches complement each other



Example of UICs

Domain variables

Vendor Details Chemical Product, inc Chemical Product, inc GardenAdmin@GardenWorld.Fertilizer [adempiere]...

File Edit View Go Tools Window Help

Client GardenWorld

Business Partner Chemical Product, inc

Product UltraGlue_Ultra Glue

Organization *

Quality Rating

Active ☒ Currency

UPC/EAN

Currency EUR

List Price 0.00

PQ Price 0.00

Last PO Price 0.00

UOM litre

Minimum Order Qty 0

Promised Delivery Time 0

Cost per Order 0.00

Part UltraGlue

Manufacturer

Discontinued ☐

Spare parts Azalea Bush Azalea Bush GardenAdmin@GardenWorld.Fertilizer [adempiere]localhost-adempiere-adem

Price ef File Edit View Go Tools Window Help

Product Client GardenWorld

Replenish Product Azalea Bush_Azalea Bush

Business Partner Joe Block

Organization *

Active ☒

Partner Product Key

Partner Category

Quality Rating

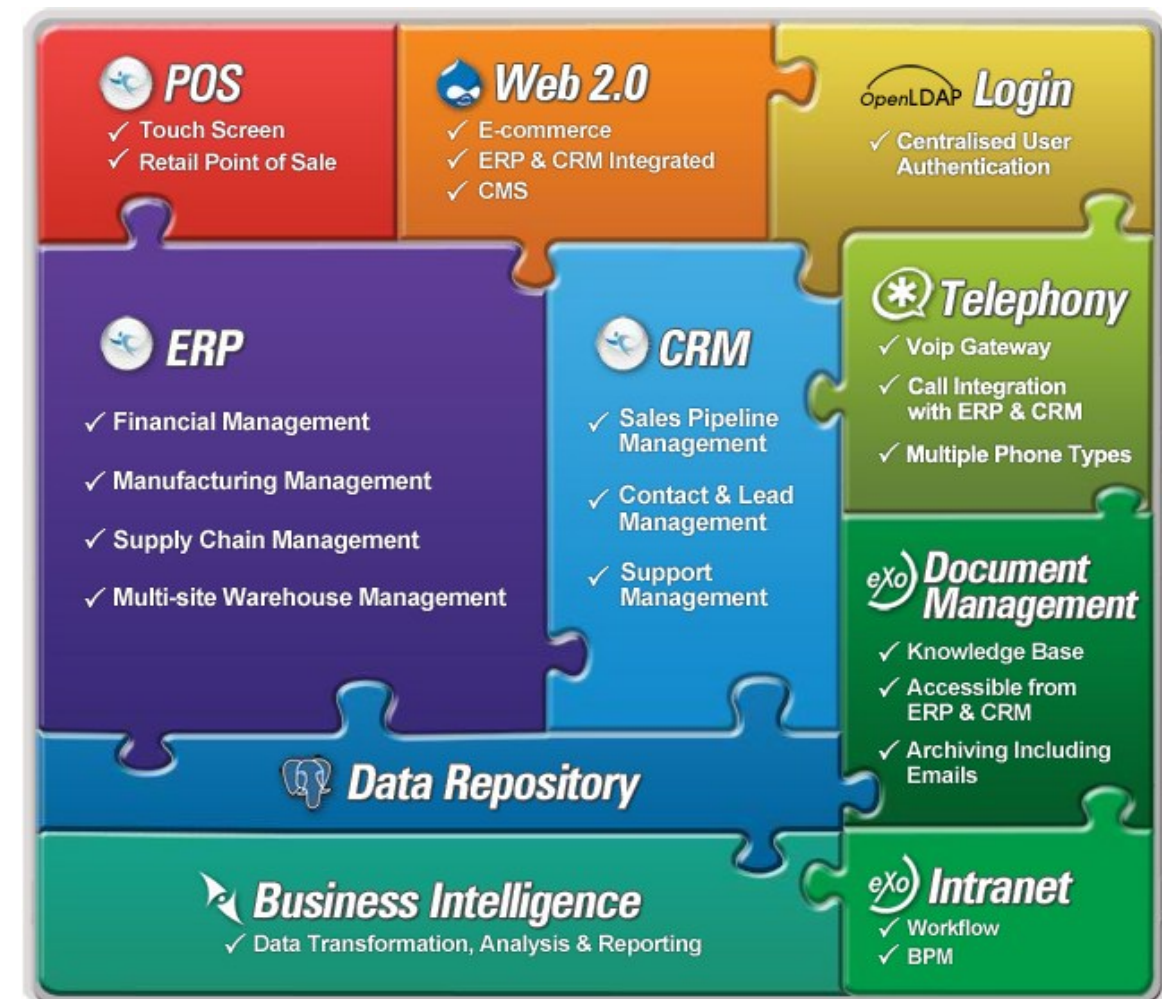
Min Shelf Life % 0

Min Shelf Life Days

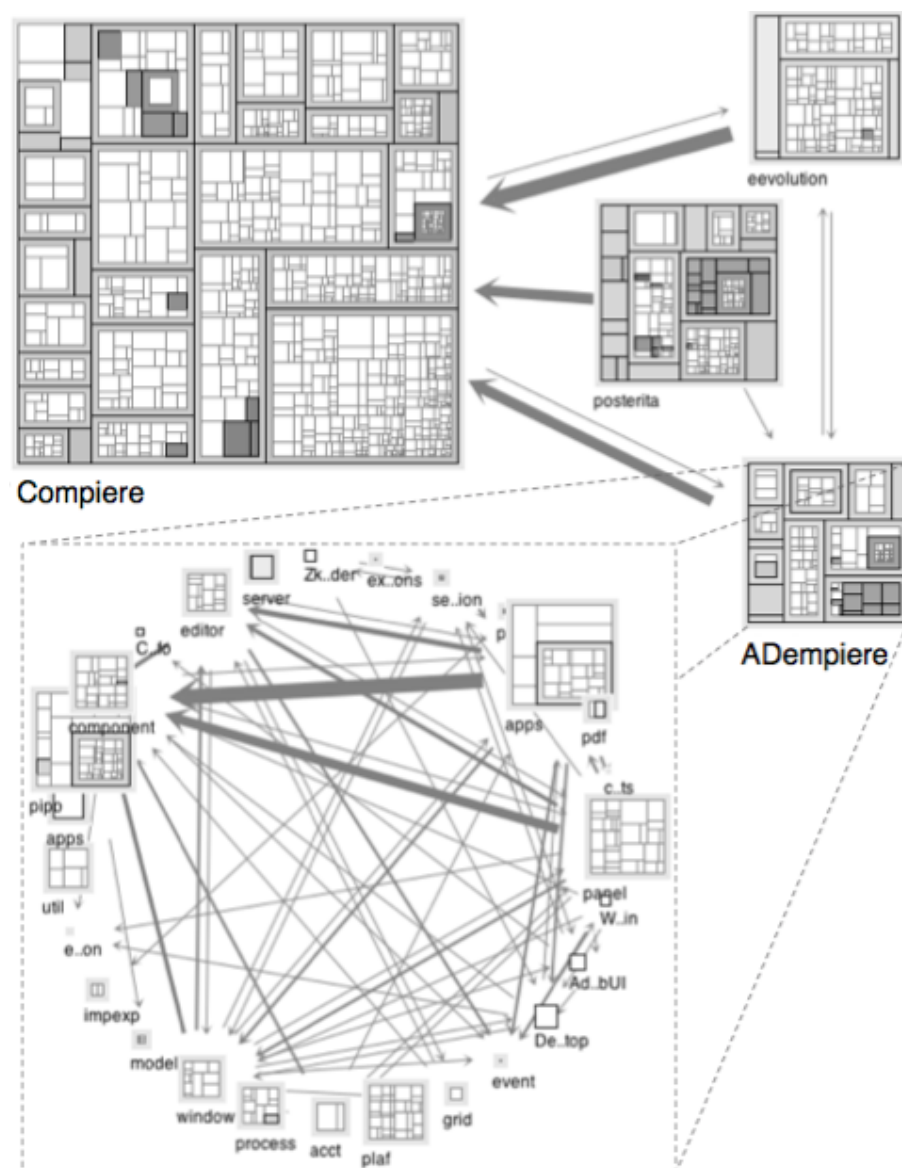


Case Study

- 120,111 times downloaded in 2011
- 3,531 Java Classes
- 2,569,854 lines of code
- Four distinct interfaces
- 347 screens



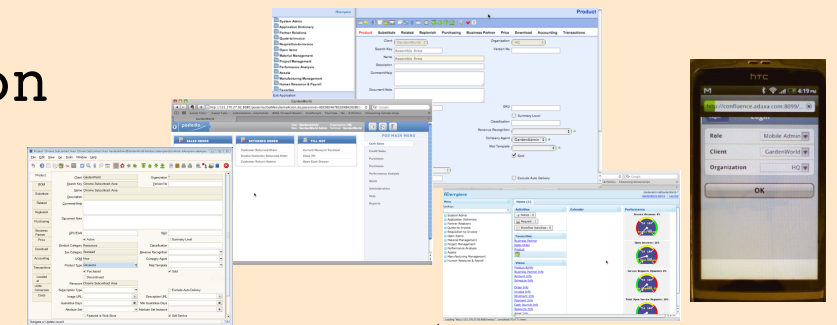
Dependencies



M. Lungu and M. Lanza, Softwareaut, CSMR 2006

17,605 Architectural dependencies

Presentation Layer



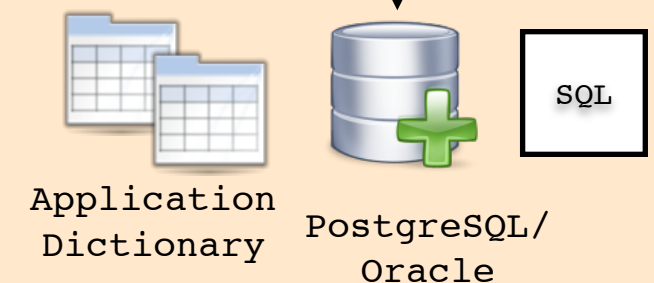
14,898 Source code dependencies

Source Code Layer



20,310 Database dependencies

Data Layer



Case Study - Orthogonality

Do conceptual and domain-based coupling
identify orthogonal dependencies?



Case Study - Orthogonality

	<u>Architectural Dependencies (UICs)</u>		
	CP 10	CP 20	CP 30
C (int) D	26%	26%	26%
C (diff) D	35%	38%	39%
D (diff) C	25%	25%	25%

C (int) D: Set intersection of correct dependencies identified by both conceptual and domain-based coupling

C (diff) D: Set difference of correct dependencies identified by conceptual and domain-based coupling

D (diff) C: Set difference of correct dependencies identified by conceptual and domain-based coupling

Case Study - Orthogonality

	Architectural Dependencies (UICs)	
	CP 20	CP 30
C (int) D	26%	26%
C (diff) D		39%
D (diff) C		5%

Metrics are orthogonal!

C (int) D: Set intersection of correct dependencies identified by both conceptual and domain-based coupling

C (diff) D: Set difference of correct dependencies identified by conceptual and domain-based coupling

D (diff) C: Set difference of correct dependencies identified by conceptual and domain-based coupling

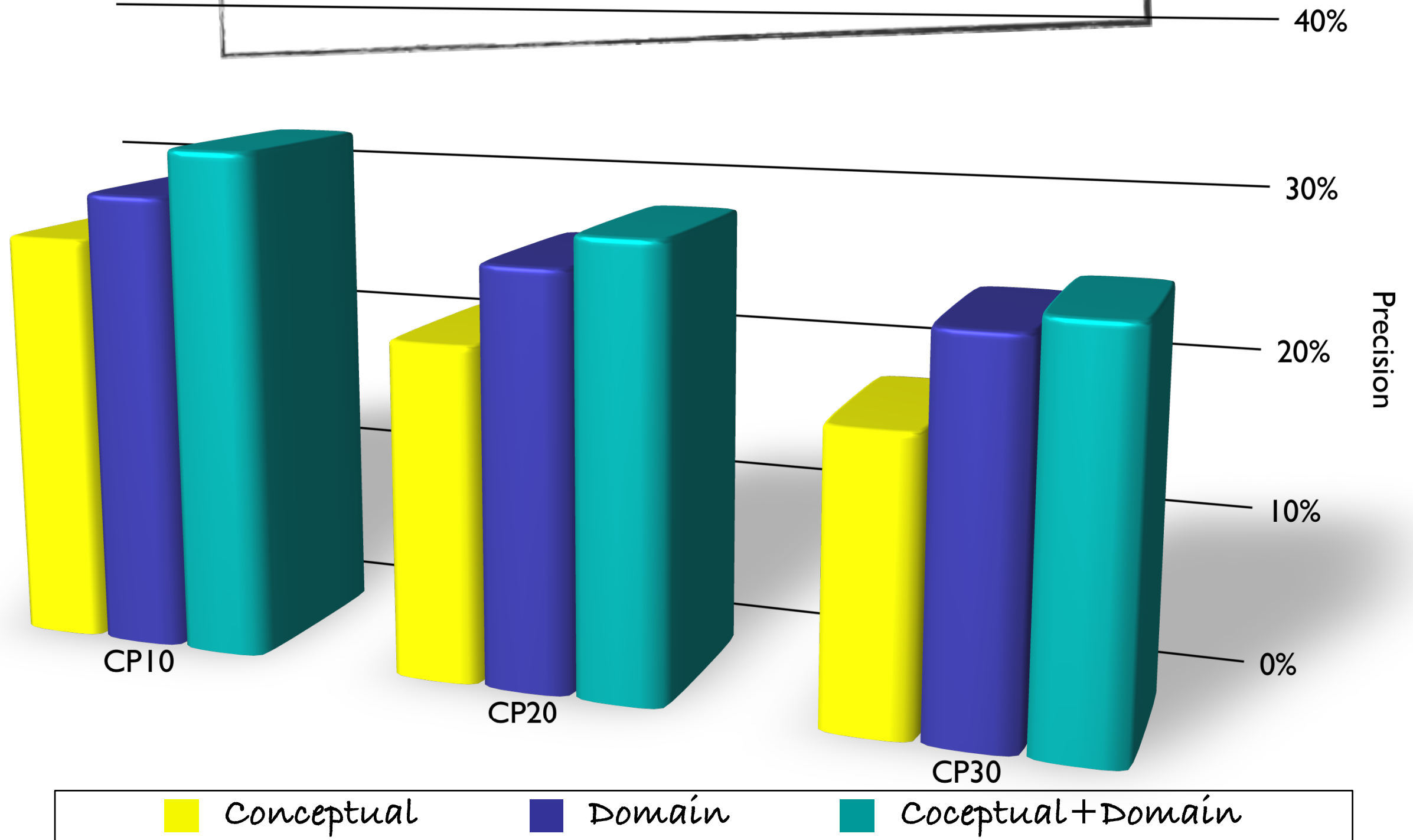
Case Study - Accuracy

Does combining conceptual and domain-based coupling improve the accuracy our ability to identify dependencies?



Case Study - Accuracy

Is it possible to improve the accuracy?



Case Study - Accuracy

The combination of conceptual and domain dependencies yields an improvement for identifying dependencies

Wilcoxon sign-ranked test indicates our findings are typically statistically significant



Case Study - Accuracy

combination of conceptual and domain
improvement for
ident

*The combination outperforms either
individual technique.*

Wilcoxon sign-rank
are typically statistically sign



Conclusion

Conceptual and domain-based coupling identify orthogonal sets of dependencies

Combining the metrics improves our ability to predict dependencies

Recall improvements of up to 7% over the baseline approach

Precision improvement up to 24% over the baseline approach

Thank You



SEMERU @ William and Mary

<http://www.cs.wm.edu/semeru>

