

Design AntiPatterns

Catia Trubiani, Antinisca Di Marco
{catia.trubiani, antinisca.dimarco}@univaq.it

Roadmap

- Background: what's a design antipattern?
- Antipatterns Reference Model
 - Design Pattern vs AntiPattern concepts
- Catalog of design AntiPatterns:
 - Software Development AntiPatterns
 - Software Architecture AntiPatterns
 - Software Project Management AntiPatterns
- Conclusion and References

Background

- AntiPatterns are a literary form that describes a commonly occurring solution to a problem that generates negative consequences
- AntiPatterns are a method for efficiently mapping a general “bad practice” to a specific class of solutions
- AntiPatterns provide real-world experience in recognizing recurring problems in the software industry and provide a detailed remedy for the most common problems
- AntiPatterns provide a common vocabulary for identifying problems and discussing solutions

Origin of Design AntiPatterns

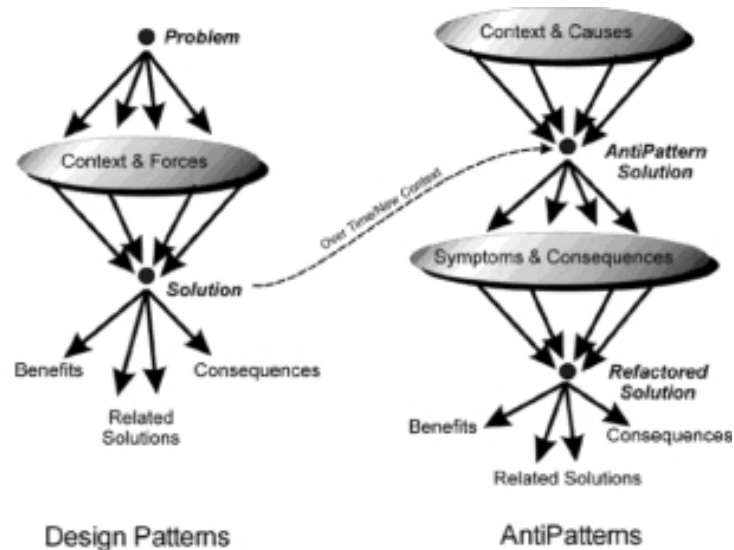
“The study of AntiPatterns is an important research activity. The presence of ‘good’ patterns in a successful system is not enough; you also must show that those patterns are absent in unsuccessful systems. Likewise, it is useful to show the presence of certain patterns (AntiPatterns) in unsuccessful systems, and their absence in successful systems”

Jim Coplien, 1995

History of Design AntiPatterns

- 1979: Brooks, Frederick P., The Mythical Man-Month, Reading, MA: Addison-Wesley
- 1995: Webster, Bruce F., Pitfalls of Object-Oriented Development, New York: M&T Books
- 1995: Coplien, James O., "A Development Process Generative Pattern Language", PLoP
- 1995: Coplien, James O., Object World briefing on design patterns, AT&T Bell Labs Conference Tutorial
- 1998: W. H. Brown et al. "Antipatterns: Refactoring Software, Architectures, and Projects in Crisis", Addison-Wesley

Antipatterns Reference Model



Patterns vs Antipatterns

- Design Patterns are organized with the pattern template for rapid understanding of the pattern purpose and solution concept.
- AntiPatterns are organized around the problem, i.e. depicted as a commonly occurring mistake. The mistaken solution maximizes the catastrophe that may underlie a fundamental problem.

Types of AntiPatterns

- **Software Development AntiPatterns:**
 - Problems encountered by programmers
 - Deal with refactoring as a form of code modification, used to improve the software structure
- **Software Architecture AntiPatterns:**
 - Problems encountered in the system structure
 - Deal with common problems and mistakes in the creation, implementation, and management of architectures
- **Software Project Management AntiPatterns:**
 - Problems encountered in the communication
 - Deal with software projects to enhance the success for people issues, processes, resources, and external relationships

Some examples of Design AntiPatterns

Software Development AntiPatterns	Software Architecture AntiPatterns	Software Project Management AntiPatterns
The Blob	Autogenerated Stovepipe	Blowhard Jamboree
Continuous Obsolescence	Stovepipe Enterprise	Analysis Paralysis
Lava Flow	Jumble	Viewgraph Engineering
Ambiguous Viewpoint	Stovepipe System	Death by Planning
Functional Decomposition	Cover Your Assets	Fear of Success
Poltergeists	Vendor Lock-In	Corncob
Spaghetti Code	Wolf Ticket	Intellectual Violence
.....

Blob AntiPattern

Blob AntiPattern description

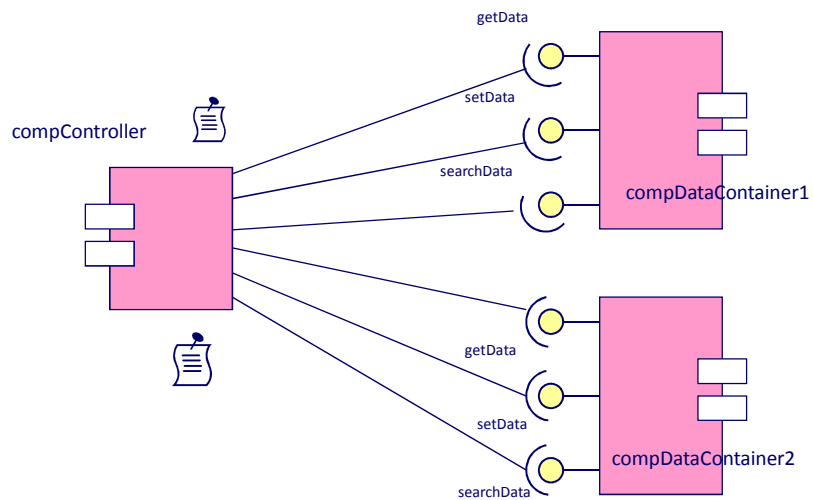
- The Blob is found in designs where one class monopolizes the processing, and other classes encapsulate data. This AntiPattern is characterized by a class diagram composed of a single complex controller class surrounded by simple data classes.
- The key problem here is that the majority of the responsibilities are allocated to a single class.



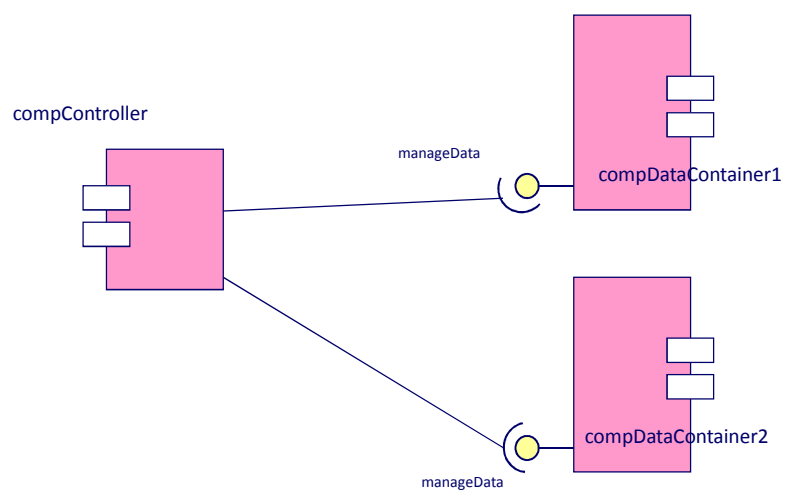
Blob Antipattern at a glance

- Problem - Procedural-style design leads to one object with numerous responsibilities and most other objects only holding data.
- Solution- Refactor the design to distribute responsibilities more uniformly and isolate the effect of changes.

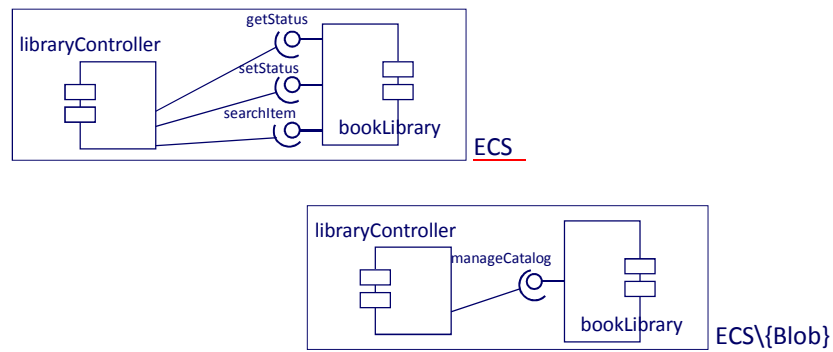
Blob Antipattern: a problem example



Blob Antipattern: a solution example

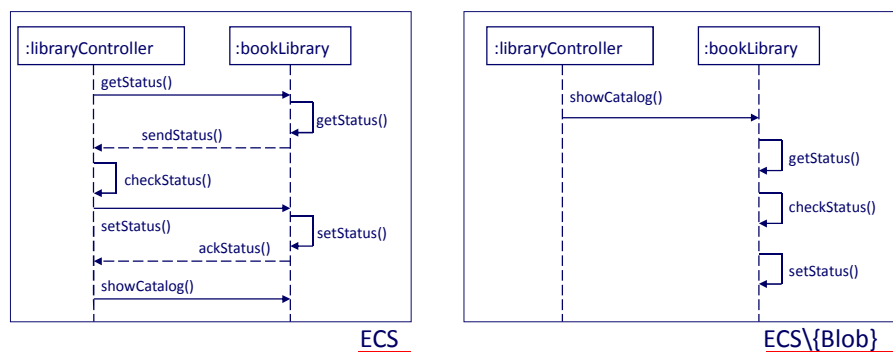


Blob Antipattern at work on the E-Commerce System (ECS)



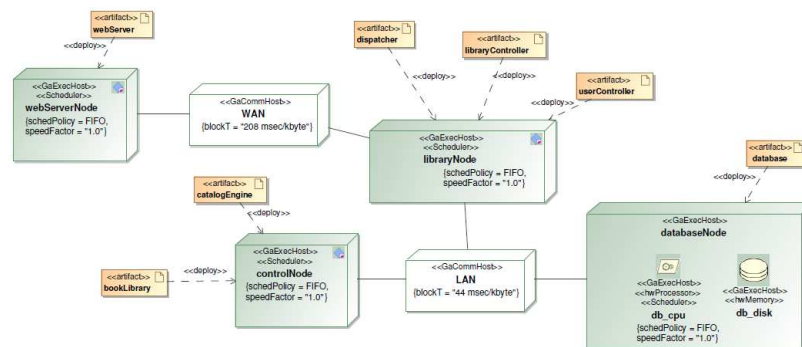
UML Component Diagram

Blob Antipattern at work on the E-Commerce System (ECS)



UML Sequence Diagram

ECS: how to get negative consequences?



Requirement	Required Value	ECS Observed Value	ECS\{blob} Observed Value
RT(browseCatalog)	1.5 sec	1.61 sec	1.44 sec



Conclusion

- What Makes it a AntiPattern? An AntiPattern must:
 - Detect a problem and describe how to solve it
 - Recur in relevant situations
 - Provide sufficient understanding to tailor the solution
 - Have a name and be referenced consistently

Further readings and References

- Other Design AntiPattern:
 - A Software Development AntiPattern: "Spaghetti code"
- Main reference:
 - W. H. Brown et al. "Antipatterns: Refactoring Software, Architectures, and Projects in Crisis", Addison-Wesley, 1998.
- Other references:
 - Laplante, P.A. and Neill, C.J., Antipatterns: Identification, Refactoring, and Management. CRC Press, 2005

