

Template can be submitted online at <http://www.itee.uq.edu.au/~patterns/submit.html>

Complex Systems Pattern Template

The following pattern format is adapted from Gamma, E., R. Helm, et al. (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley.

Submitted by

Name:

Affiliation:

Date:

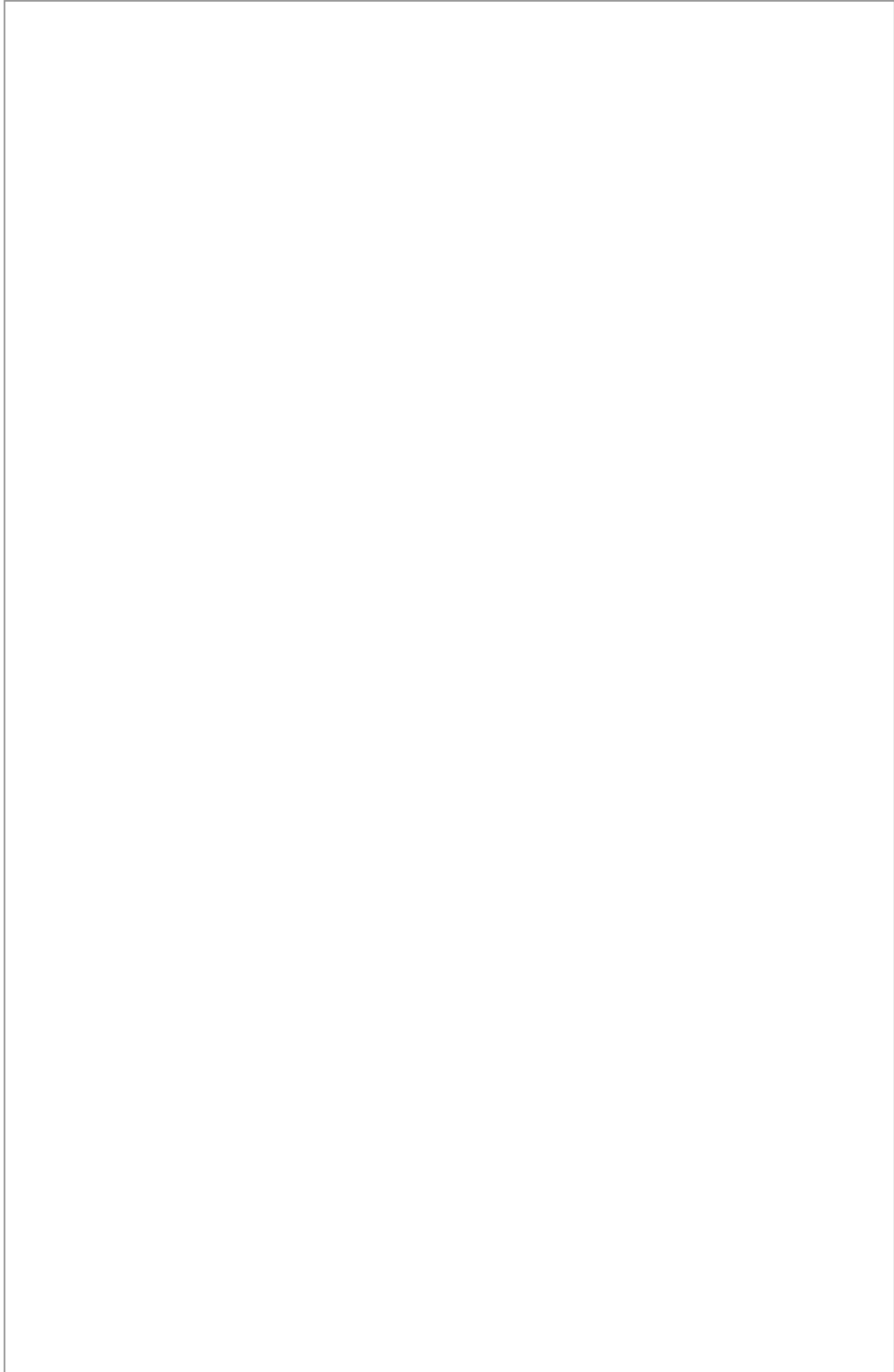
1a. Pattern name. The pattern's name conveys the essence of the pattern succinctly. A good name is vital, because it will become part of the design vocabulary.

1b. Classification. The pattern's classification should reflect its scope and use (see classifications page <http://www.itee.uq.edu.au/~patterns/repository/>).

2. Intent. A short statement that answers the following questions: What does the pattern do? What is its rationale and intent? What particular issue or problem does it address?

3. Also known as. Other well-known names for the pattern, if any.

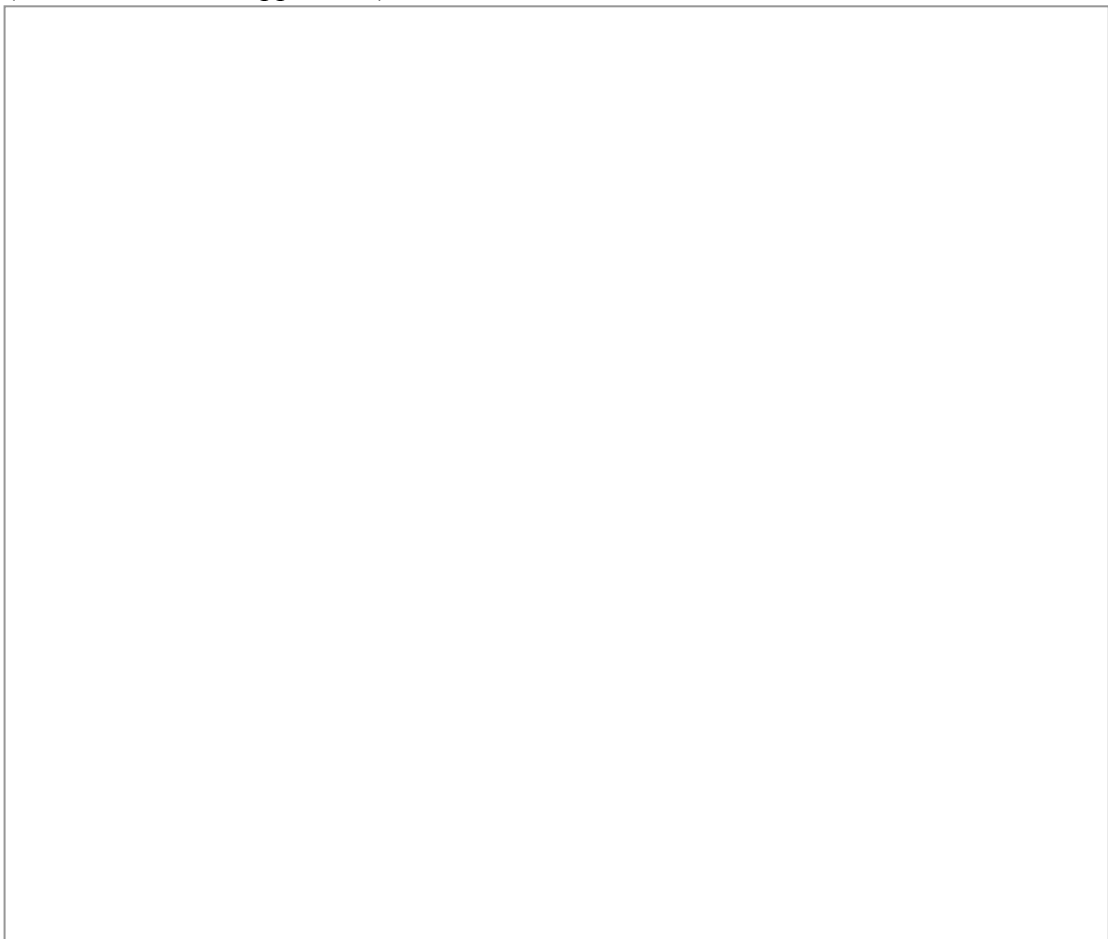
4. Motivation. A scenario that illustrates a problem and how the pattern solves the problem. The scenario aids understanding the more abstract description of the pattern that follows.



5. Applicability. What are the situations in which the Complex Systems pattern can be applied? What are examples of Complex Systems problems that the pattern can address? How can you recognize these situations?



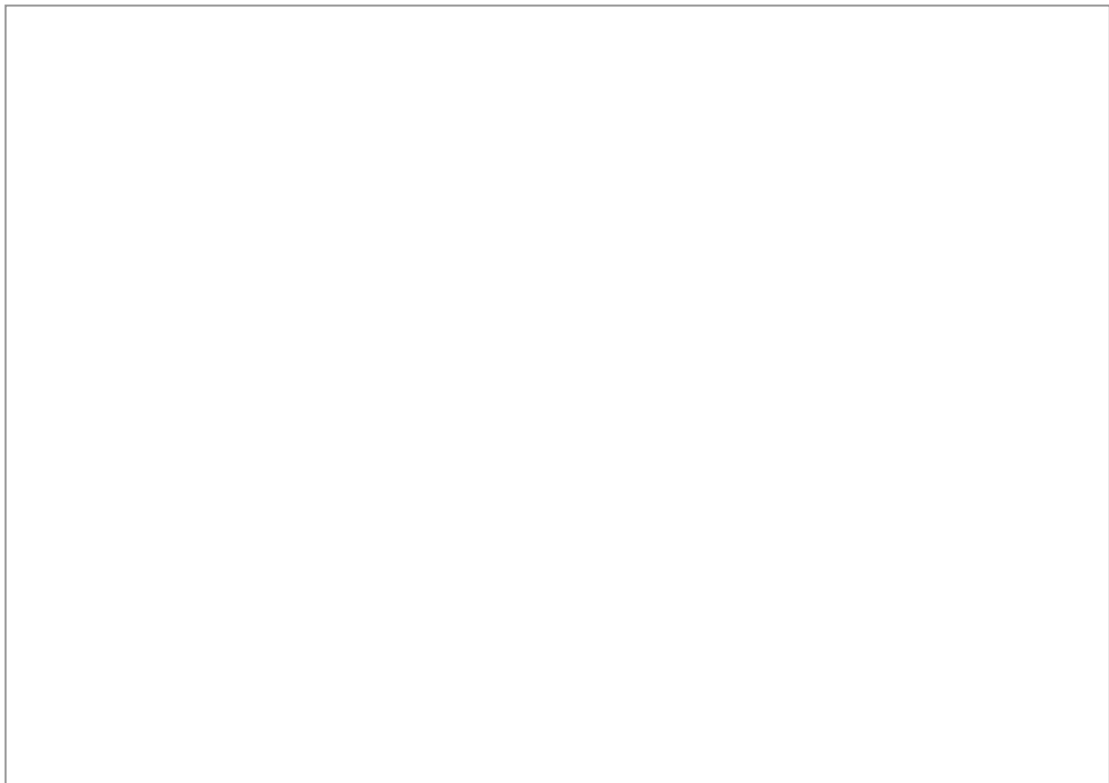
6. Structure. A graphical representation of the components of the pattern if possible (leave blank if not applicable).



7. Participants. The components of the pattern and their responsibilities (leave blank if not applicable).



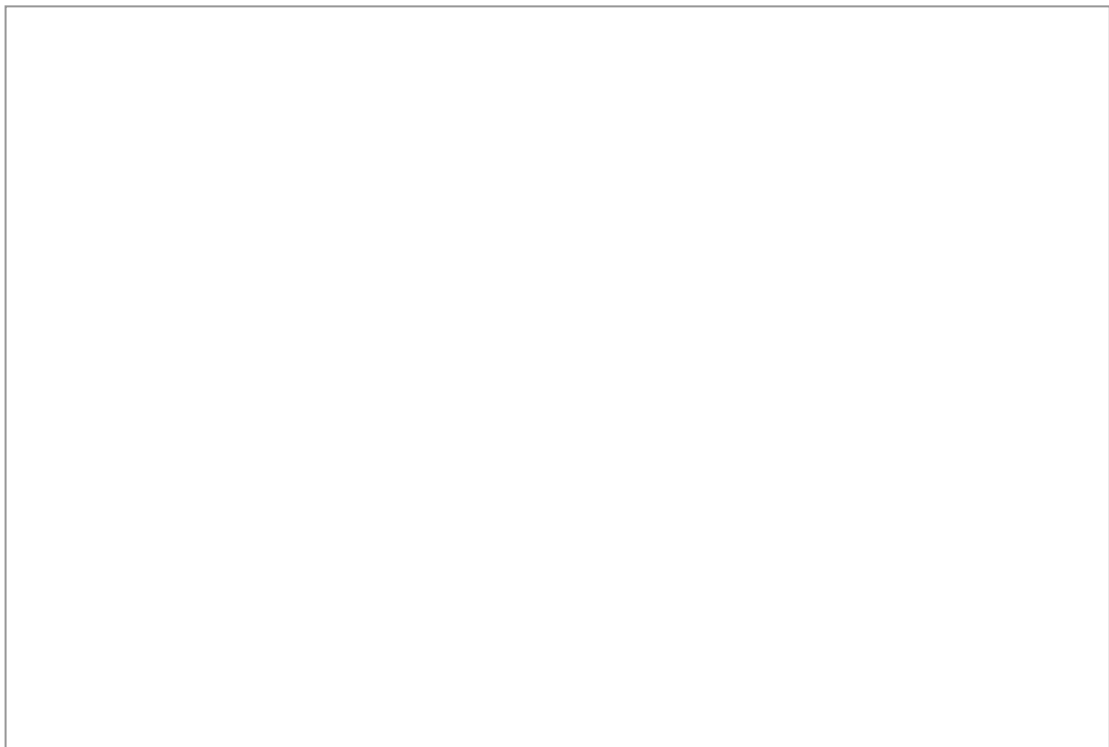
8. Collaborations. How the participants collaborate to carry out their responsibilities (leave blank if not applicable).



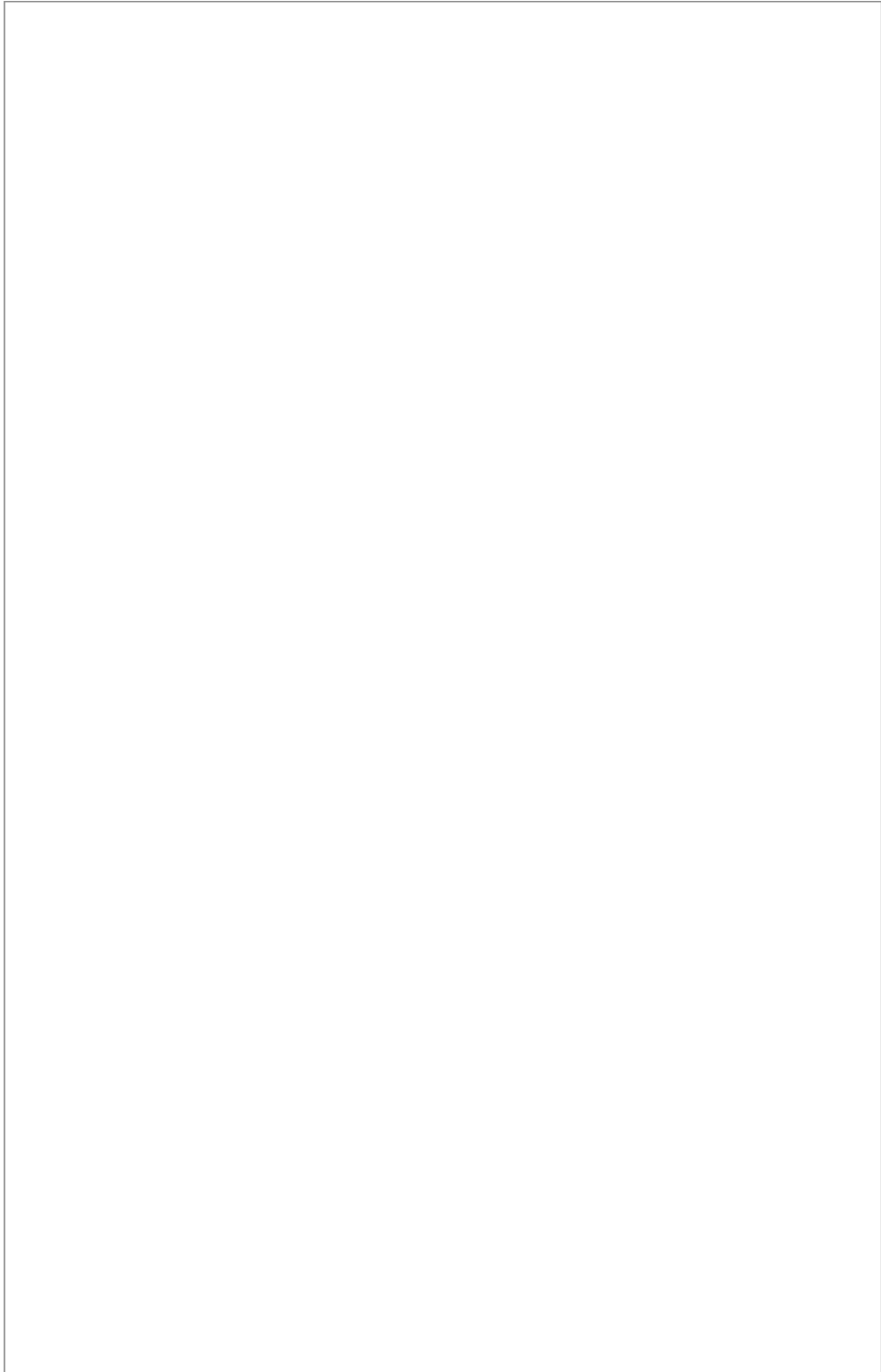
9. Consequences. How does the pattern support its objectives? What are the trade-offs and results of using the pattern?



10. Implementation. What pitfalls, hints or techniques should one be aware of when implementing the pattern? Are there platform-specific issues?



11. Sample code. Code fragments or pseudo-code that illustrate how you might implement the pattern.



12. Known uses. Examples of the pattern found in real systems. Preferably at least two examples from different domains. Give references.



13. Related patterns. Which patterns are closely related to this one? What are the important differences? With which other patterns should this one be used?

