

## The Nooron Pattern Language

Nooron is a comprehensive approach to working with information. It is built to simplify reuse. It accelerates software creation and distribution. It can be understood as the interaction of the following patterns. Warning: The Nooron Pattern Language is a wildly incomplete work in progress. The connectivity of 'RelatedPatterns' is sufficient so that `pattern_language_as_dot` produces an interesting overview. See rendered versions of the graph and the docbook output at: <http://www.noosphere.org/software/nooron/> under 'screenshots'.

## How this document was generated

This document was automatically published as a DocBook XML document by a web-based knowledge management and publishing system called Nooron. The information in this document is stored within Nooron as a knowledge base stored at: `/know/nooron_pattern_language__pattern_language_as_docbook/`. It is based on a reusable 'Pattern Language Ontology'. (An ontology is a succinct machine- and human- understandable definition of the kinds of things which exist in some domain; kind of like a relational database schema, but more flexible.) There is an associated 'Pattern Language Wardrobe' which connects the classes defined in the ontology with presentation templates which can display a particular pattern language in a variety of ways. These templates are called Nooron Page Templates which are based on Zope Page Templates (ZPTs).

These are the other templates which can be used to view this pattern language. The 'dot' format mentioned below is a graph format which shows the connections between patterns.

- `frame_details_as_html`
- `frame_as_html`
- `pattern_language_as_dot`
- `pattern_language_as_docbook`
- `pattern_language_compact_as_html`
- `pattern_language_intents_as_html`

# The Patterns

## Aggressively Free Software

### Intent

Free Software (ala the GPL) as opposed to narrowly Open Source Software is unambiguous about its availability and clearly committed to the creation of a digital commons.

### RelatedPatterns

RadicalOpenness

### References

<http://www.gnu.org/copyleft/gpl.html><sup>1</sup>

## Agile Software Development

### AKA

Extreme Programming

### Intent

Individuals and interactions over processes and tools. Working software over comprehensive documentation. Customer collaboration over contract negotiation. Responding to change over following a plan.

### RelatedPatterns

RadicalOpenness

### References

<http://agilemanifesto.org/><sup>2</sup>

## **Citizen Scientist**

### **Intent**

The professionalization of science interposes scientists between us and our participation in the improvement of knowledge. Citizen Scientists take an active role in creating their understanding of the world.

### **RelatedPatterns**

LiterateProgramming

### **References**

<http://www.amazon.com/exec/obidos/ASIN/0714525103><sup>3</sup>

## **Collective Intelligence**

### **AKA**

Hive Mind

Noosphere

Global Brain

### **Intent**

Maybe together we have enough intelligence to solve some of the really big problems.

### **RelatedPatterns**

CitizenScientist

SelfOrganization

RadicalOpenness

KnowledgeRepresentation

### **References**

<http://bootstrap.org/><sup>4</sup>

<http://www.noosphere.org><sup>5</sup>

<http://pespmc1.vub.ac.be/><sup>6</sup>

## Criteria and Evaluations

### Intent

The implicit criterion in most rating systems is 'goodness'. Every adjectival phrase is a potential criterion and hence a potential dimension along which evaluations may occur. This infinitely extensible space is the space of human opinions. Criteria and evaluations can be used to filter, order, improve and inform rich representations of information. Goodness is not enough.

### KnownUses

PICS

### RelatedPatterns

PeerReview

### References

<http://www.w3.org/PICS/><sup>7</sup>

## Don't Repeat Yourself

### AKA

DRY

### Intent

Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.

### RelatedPatterns

RecombinatoryKnowledge

Versioning

LiterateProgramming

KnowledgeRepresentation

## References

[http://www.pragmaticprogrammer.com/ppbook/extracts/rule\\_list.html](http://www.pragmaticprogrammer.com/ppbook/extracts/rule_list.html)<sup>8</sup>

## Extensible Architecture

### Intent

Build flexibility into the very nature of a system by basing it on well-defined interfaces and simple interactions.

### Applicability

Nooron benefits from the Extensible Architecture approach first by attempting to make as much of the system as possible be knowledge-based and then leveraging everything with a growing set of knowledge-based tools. For example, PyOKBC is the Python knowledge representation API (Application Programmer Interface) used by Nooron to talk to data and by the creators of 'garments' (Nooron Page Templates, nee Zope Page Templates) which portray and interact with content. PyOKBC was based on OKBC, the Open Knowledge-Base Connectivity API because OKBC was specifically designed to interact with diverse knowledge sources simultaneously, interactively and interoperably. Knowledge bases served by one machine from one kind of KRS (knowledge representation system) can do things like inherit from and transitively contain knowledge from other machines using other KRSes. The possibility of using PyOKBC backends to turn legacy datastores into knowledge stores is hence the fundamental mode of extensibility in Nooron.

By using knowledge-based techniques to generate XML output, Nooron makes the generation of a vast range of useful portrayals of knowledge to be possible with a tremendous amount of generality, reusability and customizability. Reusability will be the reward of careful and diligent authors of NooronPageTemplates.

### RelatedPatterns

KnowledgeRepresentation

REST

## Fully Automatic

### AKA

Batteries Included

### **Intent**

For new or casual users the utility of software is grounded in its clarity, predictability and reliability. Making every aspect of a system 'good enough' makes it easy for new users to get started. Having 'good enough' features trickle down from power users automatically is one way to provide usability for everyone.

### **RelatedPatterns**

FullyConfigurable

KnowledgeRepresentation

## **Fully Configurable**

### **AKA**

Mass Customization

### **Intent**

For technical or 'power' users the utility of software is grounded in its openness, automatability, and customizability. Making every aspect of a system understandable, configurable and replaceable makes it possible for the needs of serious users to be satisfied indefinitely. People who are 'power users' in some contexts can still appreciate simplicity during more mundane tasks.

### **KnownUses**

SmallTalk

Squeak

### **RelatedPatterns**

FullyAutomatic

NotSoBasicAuthentication

KnowledgeRepresentation

## Knowledge Representation

### Intent

Represent knowledge in the form of conceptual categories (called classes) and instances of those classes (called individuals) in such a way that computers can manipulate the knowledge by storing it, retrieving it and performing inferences on it.

### KnownUses

OKBC

PyOKBC

Cyc

## Literate Programming

### Intent

Generate the documentation for a system from the source code for the system so programmers are supported by their tools to create and learn from better documentation, and probably create better and more understandable code.

### KnownUses

<http://www.literateprogramming.com/ftools.html><sup>9</sup>

### RelatedPatterns

CitizenScientist

### References

<http://www.literateprogramming.com><sup>10</sup>

## Not So Basic Authentication

### Intent

Users own and maintain their identity and preference information at a location of their choosing and minimize registration effort at sites they visit.

**Method**

Users enter an URL instead of a username at a BasicAuthentication prompt.

**RelatedPatterns**

FullyConfigurable

KnowledgeRepresentation

**Peer To Peer****Intent**

Facilitate system scalability and self-organization by permitting or requiring that the computing resources of the users are contributed to the overall system.

**RelatedPatterns**

RadicalOpenness

**Peer Review****Intent**

Filter and improve human intellectual efforts by having fellow lovers of quality provide feedback and rating.

**RelatedPatterns**

CriteriaAndEvaluations

RadicalOpenness

**REpresentational State Transfer****Intent**

Make URLs a powerful part of the user interface for a web-based application by ensuring that clearly understandable parts of the URL are equivalent to subject, verb and object.

### **KnownUses**

<http://www.amazon.com><sup>11</sup>

<http://www.yahoo.com><sup>12</sup>

### **RelatedPatterns**

URLsHaveMeaning

### **References**

<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm><sup>13</sup>

<http://internet.conveyor.com/REStwiki/moin.cgi/FrontPage><sup>14</sup>

## **Radical Openness**

### **Intent**

Permit a system to benefit from the contributions of all comers by erring in the direction of open write permissions and relying on trust, peer review, and ratings to put control over the signal to noise ratio into the hands of the viewer rather than having to be the responsibility of a moderator or administrator.

### **KnownUses**

<http://c2.com/cgi/wiki?WikiWikiWeb><sup>15</sup>

<http://www.everything2.com><sup>16</sup>

### **RelatedPatterns**

PeerReview

NotSoBasicAuthentication

ExtensibleArchitecture

Versioning

REST

## **Recombinatory Knowledge**

### **AKA**

Modularization

Law Of Demeter

### **Intent**

Keep different kinds of knowledge properly separated into containers which can be 'composed' with other knowledge to support multiple uses, and diverse purposes unimagined by their creators.

### **Variations**

transclusion

### **KnownUses**

[http://www.nooron.org/know/nooron\\_app\\_architecture.pykb](http://www.nooron.org/know/nooron_app_architecture.pykb)<sup>17</sup>

### **RelatedPatterns**

KnowledgeRepresentation

DontRepeat

### **References**

<http://www.ccs.neu.edu/home/lieber/LoD.html><sup>18</sup>

## **Self Organization**

### **AKA**

Fast, Cheap and Out of Control

### **Intent**

Self-organization is the basis for the most complex, powerful systems known. Brains, life, and the universe itself exhibit the properties of spontaneous complexity generation, information manipulation and replication which underlie self-organization. Software systems which embrace these design principles can similarly be sources of surprising novelty.

### **KnownUses**

Life

Brains

Universe

### **RelatedPatterns**

RadicalOpenness

LiterateProgramming

## **Semantic Web**

### **Intent**

Transform the experience of users of the World Wide Web from surfing-between-interlinked-statically-formatted-pages, to interacting-meaningfully-with-meaning.

### **RelatedPatterns**

KnowledgeRepresentation

### **References**

<http://www.semanticweb.org/><sup>19</sup>

## **Text Considered Harmful**

### **AKA**

AssertionTheUnitOfSelection

### **Intent**

When people make the effort to express themselves in knowledge rather than text, they make it easier for machines (without AI) to process their utterances usefully. In turn, this means computers can do a much better job of helping us reuse, improve and (through richly 'typed' representations) understand our collected knowledge. Writing text is a form of expression whose usefulness lies somewhere on the spectrum between screaming incoherently in the wilderness and crafting fully machine-understandable knowledge in Cyc.

### **RelatedPatterns**

KnowledgeRepresentation

## References

### URLs Have Meaning

#### Intent

Enable users to request output in particular file formats simply by tacking appropriate sequences of file extensions onto the ends of URLs.

#### RelatedPatterns

REST

### Versioning

#### Intent

Keep track of the distinct, successive (possibly branching) states that information goes through.

#### Motivation

Digital information is typically frequently copied and revised. It is often the case that useful information is lost in this process or that the very history of successive versions is itself of interest. Keeping each successive significant revision of a digital artifact supports important goals such as restoration of previous state, discovery of who performed what changes and understanding of the significance of changes. It is particularly useful when there are a large number of people involved with the data in question and also when there are a large number of digital objects under management.

#### KnownUses

<http://www.gnu.org/software/rcs/rcs.html><sup>20</sup>

<http://www.gnu.org/software/cvs/><sup>21</sup>

<http://nodal.sourceforge.net/><sup>22</sup>

## World Views

### Intent

An information system which specifically supports the creation and improvement of caricatured 'Points of View', 'Perspectives' or 'World Views' makes possible a new kind of intellectual tourism where we view the familiar in new ways or find new things which are invisible from our habitual perspectives. These every-colored-glasses may accelerate the evolution of the traveller just as they may accelerate the evolution of knowledge, the identification of the roots of conflict and the creation of consensus.

### RelatedPatterns

PeerReview

CriteriaAndEvaluations

## Notes

1. <http://www.gnu.org/copyleft/gpl.html>
2. <http://agilemanifesto.org/>
3. <http://www.amazon.com/exec/obidos/ASIN/0714525103>
4. <http://bootstrap.org/>
5. <http://www.noosphere.org>
6. <http://pespmc1.vub.ac.be/>
7. <http://www.w3.org/PICS/>
8. [http://www.pragmaticprogrammer.com/ppbook/extracts/rule\\_list.html](http://www.pragmaticprogrammer.com/ppbook/extracts/rule_list.html)
9. <http://www.literateprogramming.com/ftools.html>
10. <http://www.literateprogramming.com>
11. <http://www.amazon.com>
12. <http://www.yahoo.com>
13. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
14. <http://internet.conveyor.com/RESTwiki/moin.cgi/FrontPage>
15. <http://c2.com/cgi/wiki?WikiWikiWeb>
16. <http://www.everything2.com>
17. [http://www.nooron.org/know/nooron\\_app\\_architecture.pykb](http://www.nooron.org/know/nooron_app_architecture.pykb)
18. <http://www.ccs.neu.edu/home/lieber/LoD.html>
19. <http://www.semanticweb.org/>
20. <http://www.gnu.org/software/rcs/rcs.html>
21. <http://www.gnu.org/software/cvs/>

22. <http://nodal.sourceforge.net/>