The Nooron Pattern Language

http://pynooron.dev.semandra.com/know/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.

1. The Patterns

Instances of pattern are the individual patterns in the pattern language constituted by the kb.

1.1. Aggressively Free Software

1.1.1. 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.

1.1.2. RelatedPatterns

RadicalOpenness

1.1.3. References

http://www.gnu.org/copyleft/gpl.html

1.2. Agile Software Development

1.2.1. AKA

Extreme Programming

1.2.2. 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.

1.2.3. RelatedPatterns

RadicalOpenness

1.2.4. References

http://agilemanifesto.org/

1.3. Citizen Scientist

1.3.1. 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.

1.3.2. RelatedPatterns

LiterateProgramming

1.3.3. References

http://www.amazon.com/exec/obidos/ASIN/0714525103

1.4. Collective Intelligence

1.4.1. AKA

Hive Mind

Noosphere

Global Brain

1.4.2. Intent

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

1.4.3. RelatedPatterns

CitizenScientist

SelfOrganization

RadicalOpenness

KnowledgeRepresentation

1.4.4. References

http://bootstrap.org/

http://www.noosphere.org

http://pespmc1.vub.ac.be/

1.5. Criteria and Evaluations

1.5.1. 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.

1.5.2. KnownUses

PICS

http://www.votery.org

1.5.3. RelatedPatterns

PeerReview

1.5.4. References

http://www.w3.org/PICS/

http://www.noosphere.org/background/goodness

1.6. Don't Repeat Yourself

1.6.1. AKA

DRY

1.6.2. Intent

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

1.6.3. RelatedPatterns

RecombinatoryKnowledge

Versioning

LiterateProgramming

KnowledgeRepresentation

1.6.4. References

http://www.pragmaticprogrammer.com/ppbook/extracts/rule_list.html

1.7. Extensible Architecture

1.7.1. Intent

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

1.7.2. 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 basees 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.

1.7.3. RelatedPatterns

KnowledgeRepresentation

REST

1.8. Fully Automatic

1.8.1. AKA

Batteries Included

1.8.2. 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.

1.8.3. RelatedPatterns

FullyConfigurable

KnowledgeRepresentation

1.9. Fully Configurable

1.9.1. AKA

Mass Customization

1.9.2. 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.

1.9.3. KnownUses

SmallTalk

Squeak

1.9.4. RelatedPatterns

FullyAutomatic

NotSoBasicAuthentication

KnowledgeRepresentation

1.10. Knowledge Representation

1.10.1. 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.

1.10.2. KnownUses

OKBC

PyOKBC

Cyc

1.11. Literate Programming

1.11.1. 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.

1.11.2. KnownUses

http://www.literateprogramming.com/ftools.html

1.11.3. RelatedPatterns

CitizenScientist

1.11.4. References

http://www.literateprogramming.com

1.12. Not So Basic Authentication

1.12.1. Intent

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

1.12.2. Method

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

1.12.3. RelatedPatterns

FullyConfigurable

KnowledgeRepresentation

1.13. Peer To Peer

1.13.1. Intent

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

1.13.2. RelatedPatterns

RadicalOpenness

1.14. Peer Review

1.14.1. Intent

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

1.14.2. RelatedPatterns

CriteriaAndEvaluations

RadicalOpenness

1.15. REpresentational State Transfer

1.15.1. 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.

1.15.2. KnownUses

http://www.amazon.com

http://www.yahoo.com

1.15.3. RelatedPatterns

URLsHaveMeaning

1.15.4. References

http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

http://internet.conveyor.com/RESTwiki/moin.cgi/FrontPage

1.16. Radical Openness

1.16.1. 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.

1.16.2. KnownUses

http://c2.com/cgi/wiki?WikiWikiWeb

http://www.everything2.com

1.16.3. RelatedPatterns

PeerReview

NotSoBasicAuthentication

ExtensibleArchitecture

Versioning

REST

1.17. Recombinatory Knowledge

1.17.1. AKA

Modularization

Law Of Demeter

1.17.2. 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.

1.17.3. Variations

transclusion

1.17.4. KnownUses

http://www.nooron.org/know/nooron_app_architecture.pykb

1.17.5. RelatedPatterns

KnowledgeRepresentation

DontRepeat

1.17.6. References

http://www.ccs.neu.edu/home/lieber/LoD.html

1.18. Self Organization

1.18.1. AKA

Fast, Cheap and Out of Control

1.18.2. 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.

1.18.3. KnownUses

Life

Brains

Universe

1.18.4. RelatedPatterns

RadicalOpenness

LiterateProgramming

1.19. Semantic Web

1.19.1. Intent

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

1.19.2. RelatedPatterns

KnowledgeRepresentation

1.19.3. References

http://www.semanticweb.org/

1.20. Text Considered Harmful

1.20.1. AKA

AssertionTheUnitOfSelection

1.20.2. 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 utterences 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 usefullness lies somewhere on the spectrum between screaming incoherently in the wilderness and crafting fully machine-understandable knowledge in Cyc.

1.20.3. RelatedPatterns

KnowledgeRepresentation

1.21. URLs Have Meaning

1.21.1. Intent

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

1.21.2. RelatedPatterns

REST

1.22. Versioning

1.22.1. Intent

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

1.22.2. 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.

1.22.3. KnownUses

http://www.gnu.org/software/rcs/rcs.html

http://www.gnu.org/software/cvs/

http://nodal.sourceforge.net/

1.23. World Views

1.23.1. 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.

1.23.2. RelatedPatterns

PeerReview

CriteriaAndEvaluations