IsisProposal

This proposal has now been submitted to the general@incubator.apache.org list for voting and should not be modified

Isis Proposal

The following presents the proposal for creating a new project within the Apache Software Foundation called Isis.

Abstract

Isis will be an extensible standards-based framework to rapidly develop and enterprise level deploy domain-driven (DDD) applications.

Proposal

The Isis project will bring together a collection of open source projects that collectively support the rapid development of domain-driven applications. The heart of Isis is the Naked Objects Framework, an established open source project that has been around since 2002. In addition, it will incorporate a number of sister projects that build on Naked Objects' pluggable architecture and which extend the reach of Naked Objects in several key areas.

In addition, the project will be reorganising the existing projects to logically separate out the components into JSR-299 beans. We believe that the JSR-299 programming model is likely to become widely used for enterprise Java applications; adopting it should make it easier for new contributors to understand how the framework fits together and therefore to develop their own extensions. In turn, we hope this will further extend the reach of the framework to other complementary open source frameworks (either within Apache or outside of it).

Background

Naked Objects is an open source Java framework that was originally developed to explore the idea of enterprise systems that treat the user as a "problem solver, not a process follower". Conceived by Richard Pawson, the first version of the framework was written by Robert Matthews (2002). Richard and Rob also wrote a book, Naked Objects (Wiley, 2002), to explain the idea.

More generally, Naked Objects is an implementation of the naked objects architectural pattern. In its purest form, "all" the developer has to do is develop their domain model as pojos; Naked Objects then provides: a object-oriented user interface by rendering those pojos; persistence by extracting the content of the pojos; security by wrapping access to the pojos; remoting by turning local calls into remote ones; and localisation by adapting all the names used in the metamodel. All of this is done reflectively at runtime so that the developer can concentrate on the most important aspect - the application itself. You can think of Naked Objects' OOUI generation as analogous to Hibernate and other ORMs, but rather than reflecting the pojo into the persistence layer, they are reflected into the presentation layer. A number of other open source frameworks cite it as their inspiration, including JMatter, OpenXava, and Trails.

Over this time Naked Objects has attracted a fair degree of attention among the early adopter crowd, generally splitting opinion as either a very good idea or a very bad one. A common misconception is that naked objects is only appropriate for simple CRUD based applications. While developing CRUD applications is indeed trivial, an important innovation is that the UI generated by NO also renders the pojo's commands/behaviors (we call them actions). Simply stated: any public method that does not represent a property or collection is rendered so it can be invoked, eg with a button, a menu item or a hyperlink. We characterize entities with such behaviors as "behaviorally complete". It's OO as your mother taught it to you.

At the same time that we have been developing our ideas on the naked objects, there has been a resurgent interest in object modelling at the enterprise level, specifically as described by Eric Evans' book, Domain Driven Design. Recognizing that there's a lot of synergy between the two ideas, the NO framework now uses DDD terminology, such as repository, domain service and value.

As mentioned in the proposal section, Isis will consist of both the original NO framework, along with a number of sister projects. These sister projects were written by Dan Haywood to support a book he wrote about the framework, Domain Driven Design using Naked Objects (Pragmatic Bookshelf, 2009). The intent of these projects was to demonstrate the pluggable nature of the framework.

Both Naked Objects and its sister projects are under the ASL v2 license.

Not directly related to this proposal but worth mentioning: Naked Objects has also been ported to the .NET platform, as a commercial product. Richard Pawson, the originator of the naked objects pattern, now devotes his energies to the .NET version and is no longer involved in the open source Java version. Conversely, Rob Matthews, the originator of the framework and a co-author of this proposal, now devotes his energies to the Java version, not the .NET one.

Rationale

We recognize that the key to open source projects long-term success is a large user base, along with a goodly number of diverse active and enthusiastic committers. Being brutally honest, we cannot claim to have either. That said, we are not naive enough to think that entrance into the Apache incubator will automatically bring us these things. Rather, we believe it will give us a platform to more effectively publicize the project so that it can succeed. It will also allow us to take advantage of the collaborative environment that the Apache Software Foundation provides. Attracting a diverse group of developers will also provide the opportunity for significant advancements and improvements to the Isis framework, making it more useful for more people.

There are, then, several reasons for us wanting to contribute the framework to Apache.

First, it helps us legitimize the "naked objects" concept. Notwithstanding the fact that the project has attracted its fair share of nay-sayers, as its developers we remain convinced of its usefulness and contribution to enterprise development in general. Most significantly, (v2.0 of) Naked Objects was used to develop the online application for benefits administration of pensions and other state benefits for the Irish Government. This project went live in 2006, is used by 1500+ users on a day-by-day basis, consists of an enterprise domain model of approximately 500 entities, and pushes out a new release each month. Richard and Dan remain consultants to this project; we would dearly like others to reap the benefit of building enterprise applications in this way.

Second, and as already mentioned, it gives us a platform on which to publicize. The Naked Objects framework did have its moment in the sun about 5~6 years back, but, at that time, it was under a GPL license rather than ASL v2. We were also solely focused in developing the aforementioned benefits system, rather than building an open source community. One could argue that we had an opportunity and we blew it; at any rate what we hope is that Apache will give us an opportunity to build up a new community. At Devoxx 2009 we ran an informal poll to get opinions of Naked Objects, from "best thing since sliced bread", through "fundamentally flawed", to "never heard of it". There were 5x as many votes in "never heard of it" as there were in all of the other columns. That can either be taken as very disappointing, or as an opportunity. We prefer the latter interpretation.

Third, by renaming the project to Isis, it gives us a chance to reposition the framework. While the "naked objects" pattern is important, we also want to emphasize domain-driven design. Alistair Cockburn's hexagonal (or "ports and adapters") architecture is another influence; the plugins that the NO framework supports (see nakedobjects.org/plugins) are either ports/adapters from the presentation layer, or ports/adapters to the persistence layer. Furthermore, the newer UI viewers that we have been developing allow the UI to be customized in various ways and to various extents; so the pojos are not necessarily naked, they are lightly (or heavily!) clad. And also, being blunt, that term "naked", while attracting the "bleeding edge" guys, tends to be a turn-off for the "early majority" who we now want to target.

Fourth, it removes doubt over its direction. Currently the NO framework is ASLv2 but copyright Naked Objects Group Ltd (NOGL), with Richard Pawson still the figurehead of the naked objects movement. As already mentioned, NOGL's energy is in their commercial .NET product. They are happy to donate the relevant rights to this software to Apache because they recognise that the framework is already critically dependent upon the open source community, so this is the best way to encourage greater take up, and ensure its future. Changing the name of the Java version also means it removes confusion in the market place as to what Naked Objects framework is (ie a .NET product only). Meanwhile the rights to the various sister projects that Dan has written would also be donated to ASF. Having a single legal entity - ASF - owning rights for all of this software would be very desirable; we think it might prompt others to explore the framework.

Fifth, the synergies with other Apache projects will help us meet our ambition to make the framework easier to extend. There are two principle extension points of the framework: viewers, and object stores. While we do understand that it isn't a goal of Apache per se to create a portfolio of frameworks, we hope that being part of the Apache family might encourage members of these other communities to help us develop new viewers or object stores. One of the sister projects provides a customizable viewer that uses Wicket; since pre-announcing this proposal on the incubator mailing list we've had one expression of interest to develop a new viewer using Tapestry.

The 'domain services' angle of DDD also means there are opportunities to integrate with frameworks that aren't just about presentation or persistence; in Dan's book he sketches out an integration with [[camel.apache.org|Camel]; there are multiple opportunities here. We also hope to tap into expertise to help us refactor the framework components into JSR-299 beans. Again, we've had an expression of interest from the incubator mailing list along these lines.

Sixth, it isn't finished. As has been pointed out to us, projects whose codebases are finished don't make for good project candidates. Isis, though, will probably never be truly finished. The hexagonal architecture, as we think of it, is about plugging in different presentation and persistence layers. We have several viewers that are in active development (including the Wicket, and a RESTful-based viewer), and object stores too (BerkleyDB, MongoDB, vanilla SQL). But there are lots of UI frameworks we haven't even started on, either Apache's own (eg Click, Tapestry, MyFaces, Pivot, …) or external (eg Vaadin, Portals, Android, JavaFX, NetBeans RCP, Eclipse RCP, Eclipse RAP, FLEX, Silverlight, …). The same is true for persistence technologies, both internal to Apache (eg CouchDB, OpenJPA, Cassandra, Cayenne, HBase, iBATIS, ...) and external (eg neo4j, db4o, BigTable, Amazon S3, JCloud …). And… there are also lots of development tools that could be built, either IDE integrations, or into build tools such as Maven.

In summary: we hope that incubation will allow us to develop Isis into a standards-based framework for building domain-driven apps, appealing both to its user community (who just want to use it "out-of-the-box") and to its contributor community (who want to quickly understand how it works and what is required to extend it).

Initial Source

1. Combine the codebases

Both the core Naked Objects framework and the sister projects reside in Subversion trees, hosted on SourceForge:

- nakedobjects.sourceforge.net
- wicketobjects.sourceforge.net
- restfulobjects.sourceforge.net
- jpaobjects.sourceforge.net
- testedobjects.sourceforge.net (FitNesse, Concordion)
- groovyobjects.sourceforge.net

These will need to be moved into a single Subversion tree, hosted on Apache infrastructure.

2. Rationalize the builds

Both the NO codebase and the sister projects are built using Maven 2. It shouldn't be difficult to combine these into a single build.

3. Standardize package names

Naked Objects package names are currently:

- org.nakedobjects.applib.* and org.nakedobjects.service.* for the applib and domain services
- org.nakedobjects.core.* for the core

- org.nakedobjects.plugins.xxx for each plugin

These should move, respectively, to

- org.apache.isis.application.*
- org.apache.isis.core.* and
- org.apache.isis.alternatives.xxx (we expect that plugins will become alternatives under JSR-299).

The sister projects package names are currently:

- org.starobjects.wicket.* (for wicket objects)
- org.starobjects.restful.* (for restful objects)

etc.

Because these are all just plugins/alternatives, they should just move to org.apache.isis.alternatives.*.

4. Move the version number down.

To emphasize the fact that this is a new project not yet considered complete, we will move the number back down to < 1.0, eg v0.1. This will allow us to work on a number of releases, hopefully getting to 1.0 as and when we graduate from the incubator.

5. Establish continuous integration

The Naked Objects framework currently builds on its own Hudson server; we would move this over to run on Apache infrastructure.

6. Rationalize documentation

The documentation for the sister projects is reasonably up-to-date, but the documentation for Naked Objects needs rationalizing, aligning with the core component and the various plugins. This will help make the framework more digestible to new users/would-be committers; they can focus on the core, or a bit of the core (say, the metamodel), or work on just one plugin.

7. Rationalize the Maven sites

Related to above, we need to "tell the story better" so that would-be users can see what benefits using the framework will bring (and, conversely, what freedom they give up in adopting a framework).

8. Review/copy over outstanding tickets.

There are a number of tickets in the Naked Objects TRAC wiki. These should be either moved over, or fixed.

Initial Goals

The following outlines some of the goals we have set ourselves during incubation. Of course, these may change as we proceed and learn more.

- Prepare ground by defining the 3 area of Isis: Application; Framework; and Plugin.
- Address (either fix or transfer) all tickets from Naked Objects TRAC wiki.
- Ensure existing documentation (of which there is a reasonable amount) is correctly related to each project now that the documentation has been separated out.
- v 0.1 - source code combination and rationalization (as per above).
- v 0.2 - refactor components to JSR-299, while maintaining backwards compatibility for bootstrapping.
- v 0.3 - JPA persistor ported from Hibernate to Apache OpenJPA.
- v 0.4 - integrate with JMX for runtime management; provide profiling of client/server and webapps (eg serialization vs domain logic vs domain services vs object store timings).
- v 0.5 - write contract tests for all major plugin APIs (object stores, authentication, authorization, remoting).

We also have a number of overarching goals:

- steadily improve the code coverage
- clean up the APIs. Some of the code dates back to Java 1.1 (at one point in time the code was cross-compiled into J# code); so there is opportunity to use more generics and remove use of arrays
- steadily reduce the amount of proprietary code, and the code size in general; use newer libraries such as google-collections more extensively.

As well as the work going on to create the Isis project there are a number of components that are in the works, and that will be released as they are ready:

- Scimpi web application release.
- Introduce dynamic view design into the DnD viewer.
- Wicket viewer release.
- NOSQL persistor release (using CouchDB, MongoDB and BerkeleyDB).
- SQL persistor release.
- CLI viewer release.
- Portal integration: Examine and implement support for compatible portals. Under consideration: WebSphere Portal Server.

Whether these are part of incubation or not will depend on whether we feel we have reached a self-sustaining community (but it's more likely than not that they will be released during incubation). Equally, there may be other viewers/persistors using other technologies that might be implemented during incubation.

Current Status

Naked Objects 4.0.0 was released at the end of 2009, broadly corresponding to the release of Dan's book.This is released into the Maven central repo, along with an application archetype for quick-start. The three sister projects mentioned in Dan's book (restful, tested, jpa) are at 1.0-beta-3, but not formally released into the Maven central repo. The remaining sister projects are in alpha status.

The main committers for the codebases to date have been Robert Matthews and Dan Haywood. Both Rob and Dan work on the NOF core, and each also works independently (reflecting their individual interests) on their respective plugins. Much work was done on the core by both Rob and Dan leading up to the release of NOF 4.0.0, and we are now reasonably happy with it. Much work remains (see above) in the area of plugins/alternatives; there is work to complete and improve the existing ones and many opportunities to develop new ones.

We readily support users on the NO forum (on SourceForge) and also on the forum for Dan's book (on pragprog.com). As a consequence of Dan's book, a GWT-based viewer (non open source) has been developed separately, and we have provided support for this (and hope it will be contributed back to the framework in the future).

Over the years we have received some patches for the framework, which we have incorporated, but not many. Part of the reason for this, we believe, is that until NOF 4.0.0 it had a monolithic architecture, making it difficult for would-be contributors to provide small patches. We think that NOF 4.0.0 improves in this area, but a move to JSR-299 would be a major step forward to help bring up participation.

Community

We recognize that the lack of a large (or at least, vocal) user community is the weakest part of our proposal. That said, we do have a steady trickle of queries on both the Naked Objects forum, and on the forum for Dan's book. Getting NOF 4.0.0 released has rekindled interest in at least one long-time user who is helping Rob to test one of the object store plugins, while we've also picked up commitment to help with this Apache proposal from a couple of people via the book forum.

To help build up our community we intend to:

- ensure that the website and documentation is first-rate (see initial goals, above)
- make sure that the Isis code can be easily used and understood
- court other open source projects with compatible technologies to work on integrations with Isis
- write a series of articles for leading web journals, eg theserverside.com, javaworld.com, artima.com. We would want to point out that we were in the Apache Incubator, and actively looking for help
- submit sessions to Devoxx and similar, Java-focused, conferences; again we'd trade on the Apache Incubator status.

We also hope that some of the newer members of our community will help us identify what the roadblocks are to adoption, so that we can address them.

Core Developers

The core developers are:

- Robert Matthews, UK-based independent consultant. Original author of the Naked Objects framework, committer to the NOF core and primary developer of the NOF plugins (DnD viewer, HTML viewer, Scimpi viewer, in-memory ObjectStore, XML ObjectStore, !BerkeleyDB ObjectStore, SQL ObjectStore, !MongoDB ObjectStore). Until recently, worked for Naked Objects Group Ltd on the commercial .NET version. Is now independent and working on apps built using the open source Java version.
- Dan Haywood, UK-based independent consultant. Contributor to the Naked Objects framework since 2005; took lead in much of the restructuring of the NO architecture for NOF 4.0.0. Also primary developer for sister projects plugins (RestfulObjects viewer, WicketObjects viewer, JPA ObjectStore, TestedObjects "viewer", Groovy support). Part-time consultant/advisor to the Irish Government project (since 2004); also a trainer /consultant in agile, Java, TDD etc.

Additional committers are:

- Kevin Meyer, South Africa-based freelance developer and business analyst. Kevin has been working primarily in a testing role, both on the SQL Object Store with Rob and on the Wicket viewer with Dan. Kevin has recently started contributing fixes to both.
- Dave Slaughter, US-based developer/consultant who is the Lead of the Software and Specialty Engineering group at SM&A. Dave has spent his career in the development of enterprise applications for companies such as Siemens, Sprint and Lockheed Martin. He has started a SWT viewer and has also started improving code coverage of the XML ObjectStore.
- Alexander Krasnukhin, a Swedish-based developer who has spent more than a year developing different applications on Naked Objects v3.0.3 and spent six months developing a closed-source GWT viewer for Naked Objects v4.0 for his former employer in Russia (Novosoft). Alexander is interested in developing a new viewer for Android.

As a result of a correspondence on the incubator mailing list, we have also had interest from:

- Mohammad Nour El-Din, Egypt-based committer to Apache OpenEJB. Nour has helped us with this proposal relating to JSR-299.
- Ulrich Stark, committer to Apache Tapestry. Uli has expressed an interest in developing a Tapstry-based viewer.

We also have had interest (off list) in developing a Vaadin viewer, and we know of a student masters project that has developed a (different) Android viewer for Naked Objects 4.0, which we're keen to incorporate if we can. We are also hoping that we might persuade Alexander's previous employer to donate their GWT viewer.

Alignment

The current codebase makes heavy use of Apache projects, including: Maven, log4j, Apache Commons Codec/Collections/CLI/Lang/HttpClient and Wicket.

There is a particular opportunity to integrate nicely with both Wicket and Tapestry. Both Wicket and Tapestry are great way of building web UIs, but have little to say about the "back-end". Naked Objects, meanwhile, provides a full runtime environment with pluggable persistence layers, and exposes a metamodel to allow generic or customisable UIs to be built rapidly. The currently in-development WicketObjects viewer brings Wickets and Naked Objects together, and (as noted above) there has been interest in writing a Tapestry viewer.

Another ongoing integration project is the ongoing-development of an object store using MongoDB; the intent is to make this codebase a good basis for other similar object stores, such as Apache CouchDB.

There are no Apache projects that we are aware of that compete with Naked Objects. At its heart, NO is (a) a metamodel, and (b) a container that acts as an abstraction over a persistence layer, using the identity map pattern.

Known Risks

The biggest risk is that we fail to build a diverse community during incubation, opening up the possibility that the project could be orphaned.

That said, there is little risk that either Rob or Dan will move onto other interests; we are both independent consultants and have the resources and inclination to continue working on the codebase. Indeed, with Rob now working only on the Java version (and not the .NET one) and Dan having finished his book, we have more resources now than at any time in the last couple of years.

Inexperience with Open Source

Although Naked Objects is an open source project, the number of committers is so small then we cannot claim great experience with open source. Neither Rob nor Dan are committers to any other open source project, though both have submitted occasional patches to the various open source projects that we use.

We are, however, comfortable users of open source projects. We also appreciate that there are lots of open source projects out there and that most developers will form an impression of a project without necessarily ever trying it out. This is one of the reasons why we feel we need to bring the two different codebases together, and create a standard message about what Apache Isis is about ("rapid development", "domain-driven design", "standard, extensible architecture", "customizable UIs").

Homogeneous Developers

The two main developers, Rob and Dan, are based in the UK. Although we have collaborated on the framework over the years, we do not work for the same company and are independent.

The other developers mentioned in this proposal are based in South Africa, US, Sweden, Egypt and Germany.

Reliance on Salaried Developers

There are no salaried developers working on the projects. The main developers, Dan and Rob, are both independent consultants. We use non-billable time to work on the codebase, with the view to developing consultancy/services from it.

Documentation

- Richard Pawson's PhD Thesis, with foreword by Trygve Reenskaug
- Books:
    - Domain Driven Design using Naked Objects, Dan Haywood
        - pragprog.com/titles/dhnako
    - Naked Objects, Richard Pawson and Rob Matthews book Naked Objects
        - full text available online at nakedobjects.org/book
- nakedobjects.org - current website
- danhaywood.com - Dan's blog to accompany his book
- starobjects.org - parent to Dan Haywood's sister projects; references the various SF websites for the sister projects

Source and IP Submission Plan

As mentioned earlier, the NO framework is ASLv2 but copyright belongs to Naked Objects Group Ltd. NOGL is happy to donate the relevant rights to Apache, while Dan is also happy to donate the various sister projects that he has written. Having a single legal entity - ASF - owning the relevant rights to all this software would be very desirable.

All the existing committers to the Naked Objects framework have formally granted their contributions as the copyright of NOGL; there have been no committers to Dan's sister projects other than Dan himself.

According to our checks in email archives and the SVN log, there have in addition been patches to the Naked Objects framework from 4 other individuals in the community. None of these patches is significant, and we don't believe that any infringe any other existing IP, and were provided in good faith to be the copyright of NOGL. That said, we have e-mailed these individuals in order to verify this. Worst comes to worst, we can back out their patches (based on svn diffs) and reimplement the patches as required. These steps will be performed during incubation, before our first release.

External Dependencies

Other than the Apache dependencies, all other open source projects used all have ASL v2.0 (eg Google Collections, cglib, objenesis), BSD (eg Hamcrest, ASM), MPL (eg javassist) or similarly permissive licenses. We do also have a soft dependency on an LGPL-licensed library (Hibernate) but during migration would look to migrate to the Apache equivalent (OpenJPA).

Required Resources

- Subversion
- Jira
- Hudson CI server
- Wiki
- Website space

Mailing Lists

- isis-private
- isis-dev
- isis-commits
- isis-user

Subversion Repository

https://svn.apache.org/repos/asf/incubator/isis

Issue Tracking

Jira; project known as 'isis'

Initial Committers

- Robert Matthews
- Dan Haywood
- Kevin Meyer
- Dave Slaughter
- Alexander Krasnukhin

Affiliations

Alexander is employed as a software engineer by Zenterio AB. The other committers are independent consultants.

Champion

- Mark Struberg

Sponsors: Nominated Mentors

- Mark Struberg
- Benson Margulies
- Siegfried Goeschl
- James W. Carman
- Vincent Massol

Sponsor

Apache Incubator