

SuccessStories

- [Tapestry Success Stories](#)
 - [PaperCut NG](#)
 - [Solis Reports](#)
 - [SOLIS](#)
 - [Alciweb](#)
 - [DocsDB/4](#)
 - [TheServerSide.com](#)
 - [partnermine.com](#)
 - [Intellidos Limited](#)
 - [Our Application](#)
 - [Quick Tour](#)
 - [Technology](#)
 - [Results](#)
 - [Thomson - Compu-Mark](#)
 - [US International Trade Commission EDIS Search Application](#)
 - [Greek Chess Federation](#)
 - [Graviter](#)
 - [Brother International de Mexico](#)
 - [Actualis.com](#)
 - [Seafax.com](#)
 - [www.ArabianDreams.ae](#)
 - [The National Hockey League](#)
 - [Ping Identity - PingFederate 3.0](#)
 - [BestCrosswords.com](#)
 - [Church And People](#)
 - [Kunstbilder Galerie](#)
 - [TeamWork Live](#)
 - [Tree of Life Web Project](#)
 - [PepperJobs](#)
 - [PowerSteering Software](#)
 - [Toplicht](#)
 - [UProm.TV](#)
 - [ViewObit.com](#)
 - [RODANOTECH](#)
 - [Flirtbox](#)
 - [LPSE: Indonesia E-Government Procurement](#)
 - [Indra: Single Sign on](#)

Tapestry Success Stories

Come on you lurkers! Post your stories here!

- [ChoosingTapestry](#)
- [MovingToTapestry](#)

PaperCut NG

<http://www.papercut.com/products/ng/images/screenshots/printer-list-small.jpg>

Here at [PaperCut](#) we write software to monitor, quota and report network printing and internet usage. In early 2005 we had decided that our previous product was being constrained by its architecture (not to mention showing its age), and we needed to start afresh.

We investigated all the usual suspects for web application development at the time and settled on Tapestry 3. Tapestry's component and event based model was fairly unique at the time, and was a real selling point for us. The popularity of this model today is testament to the fact that Tapestry 'got it right'. Our application, [PaperCut NG](#), has been a huge success and we are still using Tapestry 3 to this day (Sep 07).

Tapestry's component based model made it easy for us to develop components once and use them in multiple areas of our application, even though the initial learning curve and development time was quite steep. It also integrated very nicely with the other components we chose for PaperCut NG (Hibernate, Spring, Derby, Jetty, JasperReports and others). Making it look great was also quite easy (at least we think so - [see for yourself](#)). Probably most importantly, our Tapestry code has been easy to maintain.

PaperCut NG (and its Tapestry front-ends) has been installed in thousands of organizations around the world, with individual installations successfully scaling to databases with millions of records, supporting 60,000+ users and many simultaneous logins. This was all without sacrificing the relative simplicity of the application, which works equally well on a 5 user network.

Even with how far the world of web development has come over the last few years, Tapestry 3 still meets our needs and provides us with a solid, attractive and easy to maintain public face of our application. We are keen to upgrade to Tapestry 5... as soon as we find the time 😊

Solis Reports

Solis Reports is a reports and data mining framework. The framework works with any database that has a JDBC interface. The users can dig for data in the reports to arrive to the essential information, and export it in Excel. Reports are configured in xml, using a set of sql queries. The input parameters, the display of columns, the sortable status, the groupings, the totals are fully configurable in xml.

A demo is also available [online here](#).

Tapestry framework was chosen for its ability to generate user friendly, powerful and efficient web interfaces. The code is easily maintainable, and the integration with other services like JExcel API for exporting the reports content is straight-forward. Creation of special components for entering and validating various types of data works like a charm.

SOLIS REPORTS was built by **Nemesis IT**, a software company from Bucharest, Romania. Our Product Manager for Solis Reports is Rodica Balasa (rodica.balasa@nemesisit.ro).

Technologies used: Tapestry Framework, JExcel API, Jakarta Tomcat.

SOLIS

SOLIS is a CRM and sales management system for distributors, which integrates the supply-offering-order-shipment-billing chain of operations. A demo is also available [online here](#).

Tapestry framework was essential in creating a web interface based enterprise application with complex workflow and business logic. Even though the learning curve is steep, the long term benefits of using Tapestry in a complex application worth the initial effort. The framework is robust (more robust than jsps for instance) and maintenance and updates tasks take minimum time. The code is highly reusable and maintainable. Integration with Jasper Reports for accessing documents and reports from the web interface was straight forward and with excellent performance results.

SOLIS was built by **Nemesis IT**, a software company from Bucharest, Romania. Our team leader for Tapestry development is Marilen Corciovei (marilen.corciovei@nemesisit.ro).

Technologies used: Tapestry Framework, Hibernate, Jasper Reports, Jakarta Tomcat, PostgreSQL and Oracle implementations.

Alciweb

Alciweb is a Content Management System (CMS) developed by **Nemesis IT** for **Alcidian Solutions** for medium cityhalls sites in France. The site was developed for a few years starting with Tapestry 3.0 and the first implementation is online for more than 1.5 years for the city of Montigny le Bretonneux near Paris. The CMS is a robust application using Tapestry, Hibernate and a PostgreSQL database. It now holds hundred to thousand of pages.

DocsDB/4

DocsDB/4 is a professional relational document management, workflow and archiving system through a web-based interface. See <http://www.docsdbs.com>

Tapestry is a very powerful, practical and well build framework on which our entire DocsDB/4 web application is based. DocsDB/4 is a real heavy-duty web application, performance, maintainability and concurrency are very important and Tapestry helps in every way. DocsDB/4 consists of many pages with 90% dynamic content, complex business logic and huge forms, nothing like a regular web page.

Developing web applications with Tapestry felt like home for the Java and HTML programmers because of the strict separation of the presentation and the business layers. All other aspects of Tapestry, especially the component-based architecture, make up for clean, reusable and maintainable code! Tapestry is nothing like we ever experienced before developing a web application!

TheServerSide.com

As of a couple of weeks ago, **TheServerSide.com** (as well as TheServerSide.net) are running on Tapestry and **HiveMind**. The work was done by myself, Howard Lewis Ship. The majority of users did not see a difference, beyond a few refinements to the UI here and there.

Performance has been excellent, though a true before and after comparison would not be possible, since the backend database access changed from **WebLogic** CMP entity beans to Solarmetric Kodo JDO at the same time. Additionally, the application runs entirely stateless (no HttpSession), though a good bit of state is stored in hidden form fields.

It was primarily a one-man job; the bulk of the effort was dealing with the backlog of existing JSPs containing content (such as articles, interviews, and so forth). In many cases, these had to be converted into partial HTML files that are included as part of a Tapestry page.

The client is completely thrilled by the end results, in terms of performance and the promise of vastly improved functionality. Expect to see more detailed reports on TheServerSide.com.

partnermine.com

partnermine.com is a new dating site with the goal to stand out of the jungle of bad dating sites that are out there. Due to the nature of our very detailed user profiles and search engine we were faced with rather big HTML forms that even repeated in large parts on three or more pages. We started out with Struts because it sounded like the standard tool for the job, but it was such a pain, and the amount of duplicate and dumb glue code was just not standable. Maintainability would have been near to impossible without introducing errors. When we looked for alternatives Tapestry quickly became our favourite and soon after the winner, especially because of the brilliant component model that helped us to separate our pages into reusable parts, and we are well rid of duplicate code - we edit once and it changes everywhere.

It took us maybe 3-4 weeks to get properly into Tapestry and convert all our old Struts code to Tapestry. It was quite a learning curve (though the Struts one wasn't exactly fun either) and threw us apparently(!) back by a month. But I'm certain that in the end it helped us to reduce total development time. We have such a beautifully clean project now - and it's easily maintainable too! It's just a pleasure to work with it and I don't even want to think of the nightmares that Struts or similar frameworks would have given us if we had gone the full way with them.

We are very happy with the efficiency and productivity Tapestry gives us, and feel that other frameworks would have required considerably more time and /or staff to accomplish what we have now.

So, if you're in the process of evaluating what tool is the best for you, come to partnermine.com, have a look around, and enjoy the wonderful world of Tapestry.

Intellidos Limited

Our Application

IntellidosTM have produced a web served data integration tool that compiles a dossier containing all the known data about a specific entity on a web page and provides hyperlinks to other related entities. **HyperDossierTM** is designed to allow users to browse and explore large collections of corporate data through an intuitive, easy-to-use web interface. In addition, **QueryConstructorTM** allows users to create complex, cross-domain ad-hoc queries against this space to search for entities with specific properties. **HyperDossier** groups properties into reports and uses a file browser-like tree to allow uses to select which combination of properties to show.

Quick Tour

The first demo gives a general tour of **HyperDossier** (and **QueryConstructor**) and located at [*http://www.intellidos.com/viewlets/HyperDossierGeneral_viewlet_swf.html*](http://www.intellidos.com/viewlets/HyperDossierGeneral_viewlet_swf.html). It shows how collated information about small molecules, assays, proteins, people and teams is displayed and cross linked.

The second demo gives you an insight on how we used **Tapestry** to implement wizards by showing how a new report is created by an administrator. This demo is located at [*http://www.intellidos.com/viewlets/ReportCreation170804_viewlet_swf.html*](http://www.intellidos.com/viewlets/ReportCreation170804_viewlet_swf.html).

Technology

Before undertaking this development we evaluated a number of Java technologies to determine which one(s) suited our needs best. Our previous experience was in developing large, complex, component-based systems with Swing GUIs and a strong MVC architecture. JSP was rejected very early and then Struts, Turbine and **Tapestry** were evaluated by writing quick prototypes. As might be expected Struts produced working results very quickly but the code was found to be harder to maintain in larger systems due to the lack of components and a clear separation of MVC. While Turbine provided a good controller model it was found to be a little bit too low level for our needs and did not provide its own view layer.

Tapestry was found to be quite difficult to get into at first and the 'quick' prototype took longer than expected. However, this was before Howard M. Lewis Ship published **Tapestry in Action** which makes learning much easier. Once the initial learning curve had been overcome we found that we could increasingly apply our Swing experience and the development speed accelerated. **Tapestry** offers a high degree of flexibility and functionality which helped us in our aim to build a rich Swing-like web GUI. The code was nicely structured using components and the familiar MVC pattern which greatly aided understanding, maintainability and the use of automated testing.

Results

While **Tapestry** does have a fairly steep learning curve we have been delighted with the results. As you can see from the links the product looks good and has a high degree of GUI interactivity. What is not so obvious is that the GUI has gone through several evolutions and the code is still maintainable despite this. This alone should be enough to recommend **Tapestry** for any serious ongoing software engineering project.

A demo of our products can be found on our website at www.intellidos.com

Thomson - Compu-Mark

At Compu-Mark, we started using Tapestry as our default web framework for all our on-line services in August 2003. We first did an evaluation on the available/popular frameworks, and the refreshing new concept (using components) of Tapestry made us decide to use it.

The project on which we used it first, was composed out of 3 developers of which none had prior experience with Java nor web-development. They were all experienced C++ developers. Learning to do web-development in Java was expected to be a challenge, but in fact Tapestry lowered the barrier: there is HTML to do the layout, and there is Java to do the logic. Of course, some experience needed to be built up in wiring it together.

By now, 4 Tapestry-based B2B applications are launched with success. If we had to choose a web-framework again, it would again be Tapestry.

Thanks Howard!

US International Trade Commission EDIS Search Application

I had known of Tapestry for over a year when I found myself in the lead position on the ground floor of a project for the USITC. The two sub-projects were

- A back-end content feed system for moving public documents to a Web server and indexing them along with their meta-data in a search engine
- A public Web front-end for the above-mentioned search engine

Initially, the assumption was that another experienced developer would handle the back-end system alone and I would work with a junior (I don't like such titles, but it's easier than typing *less-experienced*) developer to create the Web app. It is worth mentioning that this would have been a mentoring effort for me as the junior developer had **absolutely no professional Java experience**.

I had initially dismissed Tapestry due to the apparent steep learning curve. However, after an unsuccessful attempt at using [WebWork](#), I decided to experiment with Tapestry by asking the junior developer to go through Kent Tong's excellent [Getting Started with Tapestry](#) Tutorial. Not only did he get through the tutorial just fine, but I made some encouraging observations:

1. **He was having fun.** I frequently caught this otherwise highly-composed imperturbable man exclaiming "Yes!" upon a successful mouse-click.
2. **He took personal initiative in his work.** He figured out how to set up a Tapestry environment in Eclipse using Spindle and began development of our production app without a break after working on the tutorial for only 2 or 3 days.
3. **He became resourceful.** After purchasing [Tapestry in Action](#), he actually created his own custom component and got on with form field validation with little to no assistance.
4. **He cleanly integrated with the search engine.** We exposed the search engine's API calls as [Spring](#) beans. The distinction between the controller and model could not be clearer. I cannot even think of how the line between these could be blurred.

As it happened, in less than 3 weeks the back-end task turned out to be a great deal more complex and demanding than initially anticipated. So, I had to focus almost 100% of my time on it and the junior developer was tasked with making a solo attempt at creating the front end.

He did it. The project has been a resounding success.

Tapestry components just work - no questions asked. I disagree with the statements alluding to a steep and time-consuming learning curve. Tapestry is markedly different from [Struts](#) or [WebWork](#), which are perhaps more intuitive as they resemble extensions of the application-level protocol (HTTP) of the TCP stack. However, we have found great advantage in using a framework which is divorced from this paradigm. We did not have to come up with any unexpected or unnatural extensions, and a single consistent development pattern facilitated productivity and developer communication.

It would be unfair to credit our success exclusively to Tapestry and leave the brilliance, talent and work ethic of our junior developer unrecognized. However, I sincerely believe that the choice of Tapestry was catalytic in activating these attributes. This is invaluable to someone like me, to whom software development is a people-centric endeavor.

Our client is sold on Tapestry as well, and it will be a key tool in our organization's efforts going forward. You may have a look at our application [here](#).

Regards,

Atif Faridi

Greek Chess Federation

As of 1/7/2005, the managing and rating system of Greek Chess Federation is based on Tapestry.

It's a system responsible for >30000 chess players and >400 clubs and for calculating their ratings. It manages the >200 tournaments and >13000 chess games played in Greece per semester.

It's been almost a year since I've first learned about Tapestry, but only since Nov 2004 have I decided to use it in my projects. At first, I did a thorough investigation of the framework, writing lots of test pages. My biggest initial concern was whether it could cover all my development needs. Would there be a case when something is not feasible in Tapestry? I then had a student (at University of Athens) do his final project on it. It was very pleasant to see how this (inexperienced in Java) developer came to understand and use Tapestry. It ended up becoming the fastest final project I've ever supervised. I almost feel sorry for the other (brighter) students who had to learn Struts (again, after my proposal).

Finally, I just started using it in real projects and it just worked. All my fears went away!

The only thing that I now regret is of not having known Tapestry from the very first days of its existence. It feels as if I've wasted so much time in this pre-Tapestry era!

Andreas Andreou

Graviter

Graviter is the assisted product search for [Gravit](#), an online electronic shop in Brazil (think about this as a guidester knock-off). Gravit itself is hosted on a canned "e-commerce" site, but the smart, computer-inclined shop manager wanted to provide search capabilities that go far beyond what the e-commerce site currently offers.

So I've designed a "mirror" database, that gets data exported from the e-commerce site (thanks goodness they provide an automated XML data export), and adds what we need to provide smart, consumer-oriented searches. Every product fits in a *category*, and each category has many *attributes*. The shop manager can use some administrative pages to edit product descriptions and attribute values; customers then use the search pages to find the products they want.

Graviter was done using PostgreSQL as the backend database, Hibernate 3 for object-relational mapping, Tapestry 4.0 Beta 3 for the presentation layer, and finally Spring 1.2 glueing everything together. While developing Graviter I've found and patched [a few bugs](#) in Tapestry, and contributed [some translations](#) as well. These bugs were easy to find and fix, thanks to the open source nature of Tapestry (and [IDEA's](#) amazing code inspections).

— [Marcus Brito](#)

Brother International de Mexico

The site for [Brother International de Mexico](#) was developed using [MySQL](#), OpenCMS 5, Tapestry 3 and Hibernate 2. The site includes an online store.

[Brother International de Mexico](#) is one of the premier providers of multifunction centers, printers and facsimile machines. Brother was established in Mexico on 1992 and markets many products manufactured by its parent company, Brother Industries, Ltd., of Nagoya, Japan.

Actualis.com

We are one of the top 10 (computers, parts for home users and professionals) resellers in France. Our new website is completely based on Tapestry 3. Behind Tapestry 3 we are using Hibernate 3, Spring and JiBX. We have more than 200000 customers and more than 10000 products online.

Our choice of Tapestry was done after the difficulty to evolve our old php website. Many of our internal applications were done in Java, and then Tapestry had the best solution for form management, reusing components and management. Also the separation of code/presentation convinced the management.

The really nice thing about tapestry is that the framework pushes you to write component based. At the beginning of the project it did not make that much sense, but after 8 months of development, we have a large library of components that are saving an incredible amount of development time.

– Henri Dupre

Seafax.com

The site for <http://www.seafax.com> was developed using Oracle, Cayenne, and Tapestry 3 in 2005. It was converted from an older [WebObjects](#) 4.5 implementation and we were very happy with the conversion to open-source Java platforms. Tapestry and Cayenne were similar in concept to [WebObjects](#) (making the learning curve minimal for us) but better implemented... easier to develop, maintain and debug, faster and more portable.

The site is designed primarily for access by our clients and doesn't offer much information to non-client visitors. It provides credit reports, collection services and other information on the seafood, meat and poultry industries.

– Bryan Lewis

www.ArabianDreams.ae

Arabian Dreams is real estate company based in Dubai, U.A.E. It wanted a web site to allow customers to be able to search properties for rent and for purchase. The usual criteria: price or rent range, location, number of bedrooms, view, were to be provided to narrow down the searches. Behind the scenes, Arabian Dreams wanted a web-based administrative interface that would allow its agents to add, modify and delete property listings, manage landlords' and owners' profiles, generate various reports, generate rental and sale contracts as PDF files and a number of other utility functions. I had used Tapestry and Hibernate in a number of prototype projects when I worked for a Lahore, Pakistan-based software company. The company had been relying on JSP/EJB/J2EE frameworks and the higher management were always apprehensive about new technologies, especially, open source. The project was not as high-profiled as some of the other projects the company was working on. I decided to take on the project and use Tapestry and Hibernate. The business logic was not complex; most of the web pages in the application were straight forward - get the data based on certain criterion and display it, get user input in a form and persist it. Hibernate made the persistent part easy while Tapestry made the web-page management a breeze.

The project took about 5 months to develop and deploy. The web site <http://www.arabiandreams.ae> is hosted at webhosting service in Canada. The environment is Tapestry 3.0.3, Hibernate 2.1.6, MySQL 4.1 and Tomcat 5.

The one feature of Tapestry that I found invaluable was the ability to work in HTML designing tools like Frontpage and Dreamweaver in designing and editing HTML pages with jwcid tags. I had a graphics designer check out HTML pages with embedded jwcid tags from CVS repository, make necessary changes and commit the changes. Never had the graphics designer done something similar in other projects she had worked on, i.e., the HTML file the developer and the designer work with are the same.

My thanks to Howard for designing a framework that has made web development convenient and enjoyable.

– Sohail Aslam

The National Hockey League

The NHL <http://www.nhl.com> has been developing web applications using Tapestry since its 2.3 version. In conjunction with Cayenne, Tapestry is the foundation for the league's online presence including:

NHL.com (schedules, scores, standings, news, etc.), Statistics, Highlight Machine, Media/Video User Registration, NHL Media Site, and a handful of internal intranet apps.

Environment: Tapestry (2.3/3.0.3/4.0), Cayenne 1.2, Oracle 9i, Sun One Application Server 7.0

Development Team: Neil Pierson, Marc Smith, & Eric Schneider

– Eric Schneider (I plan to add more info when time permits)

Ping Identity - [PingFederate](#) 3.0

We released [PingFederate](#) 3.0 which was built entirely on top of Tapestry. Our [PingFederate](#) 2.0 was built on Struts, but when we started 3.0, we decided to look into alternatives. Ultimately, we decided on Tapestry. We completely re-designed and re-coded the entire application GUI, using, basically, two developers. We feel that we have a platform for future releases, whereas, 2.0 was not.

We built it on top of JBoss using Jetty. Our first product was built on JBoss using Tomcat and we decided to switch. We hired Howard to come out and bring us up to speed and that was very beneficial.

[PingFederate](#) 3.0 is an advanced federation server for SAML 2.0 web single sign-on, single log-out, and attribute exchange

[PingFederate](#) is an enterprise-grade identity federation (SAML) server for organizations that require a flexible and cost-effective way to integrate, manage, and secure disparate users and applications across internal and external security domains. It provides authentication, attribute, and authorization portability across autonomous security domains. It features a GUI administration tool, built-in key management, SSL and signing certificates, built-in JDBC and LDAPv3 adaptors for advanced attribute management, setup wizards, online help, a user manual, an SDK, and a demo.

It is a commercial product, but the first 100,000 transactions are free.

<http://www.pingidentity.com/products/download>

— Mark Stang

BestCrosswords.com

[BestCrosswords.com](#) is an online crossword puzzle website. It was originally written in perl but was eventually ported to java to ease the development process. Recently, the site has been migrated from Tapestry 3 to Tapestry 4 and the transition proved smooth.

For more info on the site, visit the [company site](#).

Church And People

The [Church and People](#) web site provides churches with community building and administration tools. It includes online event registration, credit card processing, resource management, membership directories, private forums, an event calendar, e-mail tools, group management, member information, and more. The Church and People application works with a church's existing web site and accounting package, rather than trying to replace them.

The application has nearly 200 tapestry pages (including administration pages), many of which have complex view and/or associated business logic, and many of which change appearance based on the login privileges of the user. Three of Tapestry's most important strengths for this project were its ability to use a prototype HTML page as the template, the ease with which components can be constructed, and the high quality of the error messages that allowed problems to be resolved quickly.

Custom components included a *role* component that conditionally included or excluded content based on a user's security privileges; a dynamic *tab* component that is security-aware; link components such as *PersonLink* that act similar to a [DirectLink](#)--but take business objects as parameters and build the visible link text and the destination page based on object data and user roles; and several components that help integrate the application to the credit card processing engine. Creating a new component is no more difficult than creating a new page.

The application itself maintains session state related only to the user's identity, memberships, and permissions. Otherwise, the application is stateless. Application context is maintained through identifiers in hidden fields or URL parameters. These identifiers typically refer to Hibernate objects on which a page is operating.

Some *redirect after post* functionality was written in an application-specific way (that does not use the session) by providing a redirect service that could cooperate with a framework-wide base page class to initialize the page based on URL parameters. (In the current Church and People release, this is not yet a complete solution because posts with error messages are not redirected. A *flash message* based solution is probably required here.)

The application uses Tapestry and Hibernate, with [JasperReports](#) for a reporting engine. It is developed partly using a domain-specific language created with JavaCC and code generation tools written with Velocity. It is deployed on Linux against a MySQL database.

Kunstbilder Galerie

The [Kunstbilder-Galerie](#) web site is a german online shop that sells oilpaintings and art prints. The web site is base on tapestry 3 but will be migrated to tapestry 4 soon.

TeamWork Live

[TeamWork Live](#) is a web-based team collaboration and project management tool that combines many of the best features of Basecamp, 30Boxes, and wikis into an integrated application. [TeamWork](#) Live makes it easy for you to communicate, collaborate, and share information with your co-workers and clients. The application is designed to be easy to use and it is hosted so there's no software to download, install, or maintain.

The project was developed using Tapestry 4.0. Tapestry 3.1 was used in a previous project and much of that code was migrated to [TeamWork](#) Live. The platform is based on the Tapestry/Spring/Hibernate triad and features an AJAX-rich interface.

Tree of Life Web Project

The [Tree of Life Web Project](#) is a collaborative effort of biologists from around the world. On more than 4000 World Wide Web pages, the project provides information about the diversity of organisms on Earth, their evolutionary history (phylogeny), and characteristics. The ToL has been powered by Tapestry by over two years now, so it's about time it made its way to the wiki =) Since adopting Tapestry, the ToL has experienced rapid growth in functionality and increased stability. While the current version of the public site is on Tapestry3, things are currently being moved to Tapestry4.

PepperJobs

[PepperJobs](#) is small job board entirely developed with tapestry 4.0.2 for the frontend. Each tomcat instance runs on top of apache with mod_jk, and the http requests are load balanced with LVS (Linux Virtual Server). It was a real pleasure to develop the application with this framework, thanks to the developers and the users community.

PowerSteering Software

[PowerSteering](#) is a software vendor that developed and provides enterprise-class, On-Demand software for Lean Project Portfolio Management (Lean PPM), with a particularly strong presence in companies that are running large Six Sigma or other Operational Excellence type programs. Founded in 1998, the front-end of the application was originally built in JSPs. The functionality had grown over the years, and the navigation had become a bit circuitous and confusing, to say the least. Driven by the usability problems, as well as a desire to easily support multiple languages, we made a decision in early 2006 to re-implement the entire UI. Several technologies were investigated, with Tapestry coming out on top. Howard came over and helped us kick things off with a training session for our development team. Our initial "Version 5.0" release using the new Tapestry UI was made public in June 2006. In the October /November 2006 issue, [iSixSigma Magazine](#) published a 'Test Lab' review of four competing software products, and our Tapestry-powered one came out on top! You can see the full review [here](#). Thanks Howard!

Toplicht

The [Toplicht](#) web site is a german online shop for sailing equipment, especially for traditional boats and yachts. It was build with Tapestry 4.0.2. Right now, it's only available in German, but English texts are in the making right now and will be available soon.

UProm.TV

The [UProm.TV](#) web site is a video community websites based on Tapestry (4.1.2,) Hivemind and Postgres. Europe's first video platform with its own TV station, receivable via selected cable networks and satellite (Astra digital, Transponder 103, orbital position 19,2). Videos uploaded by the users can be voted to be aired on satellite TV and be watched by millions of viewers anywhere in Europe.

Tapestry has been chosen by [Fivecast](#) as it offers an object-oriented approach to web-programming as well as a clear separation between templates and the source code. Tapestry also allows to overwrite or implement own services or to replace any default Tapestry service. For example, the locale of the website is determined by its subdomain - which was possible using a simple Hivemind service.

Other video platforms have been launched on the same framework:

- [UFinance.TV](#)
- [UMotor.TV](#)
- [UFlirt.TV](#)

Regards,

Tobias Marx, Fivecast GmbH, Munich (Germany)

ViewObit.com

The [ViewObit.com](#) web site is a completely dynamic portal for Funeral Home clients of our Answering Service, [MyASD.com](#), to publish their obituary information online. We also allow creation of micro-sites with additional custom pages, e.g. [ViewObit.com/AlbrightFH](#). All of the data comes from a database, including mapping URLs to the appropriate site.

The site started as a Tapestry 4.0.2 site and is currently being migrated to 4.1.2, using Hibernate for database access. We appreciated the clear separation of display/template and Java code, the component architecture (so common page elements are written once) and the very rapid development. The entire site was developed in a little over a week with periodic updating for new features. We are currently adding Tapestry-based virtual hosting, which allows a domain name to be pointed to the site and have map to the appropriate pages without using any virtual hosting capabilities of the application server. This is being done using a Servlet filter and normal Tapestry external pages.

-Norman Franke

RODANOTECH

[Rodanotech](#) is a software and services company mainly dedicated to the healthcare industry. One of our mandates was to design and manage a new version of an online clinical database: [MSBase](#). We decided to use Tapestry 5 after successfully using Tapestry 4.1 on other projects. The ease of development with version 5 allowed us to deliver the project in just a few months. Compared to the previous version, written in a "classical" J2EE manner by another company, we managed to enhance performance while significantly reducing the amount of code. The back end runs with Cayenne 2 for database persistency in a MySQL database. Currently deployed in a Tomcat server with Java 6, we are considering a cluster solution with Jetty. We can say Tapestry is bringing back the fun to development, while allowing us to concentrate on business logic instead of "glue code". Also, performance is really outstanding and seeing a constant progress between Tapestry versions.

Alexander Lamb, Eric Bianchi, Geneva / Switzerland

Flirtbox

The [Flirtbox](#) web site is a free dating website that has been developed in PHP back in 1999. In 2005, the website was migrated to Tapestry 4 by Adam Greene (www.professionalopensource.ca). However, due to issues with different URLs, the front end has never been deployed and only the admin interface. With the latest stable Tapestry5 version and the publication of Igor Drobiazko's Tapestry book, I have now started a migration from Tapestry4 to Tapestry5 which will also create "artificial" .php URLs that will keep the old URL structure and sitemap. The admin interface of Flirtbox is already based on Tapestry5 as well as various external interfaces used for SMS communication or administration.

After evaluating Wicket and the latest Struts, Tapestry5 was chosen for the second relaunch due to its simplicity and scalability as well as the possibility to have 100% free control over URL structures. The IOC container also allows the outsourcing of different parts of the website to external companies and freelancers without exposing the whole source code.

Regards,

Tobias Marx, Freelancer, Munich (Germany)

LPSE: Indonesia E-Government Procurement

LPSE (Layanan Pengadaan Secara Elektronik) or *Electronic Procurement Service* is official e-procurement system used by government agencies in Indonesia. In 2006, system designer choosed Tapestry 3 as framework along with iBatis and Spring. Choice comes to Tapestry 3 because it has strict separation between *View* and *Code* elements. We have high employee turnover so we should have a solid framework.

LPSE is implemented as decentralized system. Each government agency (ministry, province, city, district, public university, stated-owned company) has single LPSE installation. Internet connection cost is still high. Moreover, Indonesia is an achipellago with more than 13,000 island. It's impossible to provide single system in capital to serve all agencies. Currently, LPSE has served total transaction [10 trillion rupiah or almost 900 million USD](#) since 2007. This number is less than 1% of total government procurement.

Until July 2010, there are more than 55 agencies have installed the system. This number continues to increase up to 300 agencies by end of 2012. Tapestry 3 will be still used. We do not yet have plan to migrate to latest version.

In newer modules, we use Tapestry 5 to take all of its benefit. The module is part of whole e-procurement system and have interface with Tapestry 3-based module.

Regards

Andik Yulianto - <http://lpse.blogdetik.com/files/2009/02/andikyuliantoyahoom.com.jpg> Jakarta - Indonesia (South East Asia)

Indra: Single Sign on

We have use T5 for Single Sign On (SSO) solution for telecommunication company Telefonica. We based our SSO solution on [OpenId](#) protocol. We had to extend [OpenId](#) protocol to be able to support user's distributed session support for centralized logout and session timeout. There where also requirements on integration with already existing system via WS and REST interafaces.

For our application we choose mix of various technologies: Hibernate, JBoss Cache, Quartz, Restlet, XJC, Maven, Tomcat and obviously T5 to put it together. Our application is running in highly concurrent clustered environment. On project where working quickly changing team up to 15 members. New team members where able to quickly join and help.

Regards

Jan Jirout, Prague (Czech Republic)

Note: LPSE installation: [LPSE of Ministry of Finance](#), LPSE of [Province of West Java](#). You can found other LPSE links there.