

Der Energie Navigator

Thomas Kurpick, Claas Pinkernell, Bernhard Rumpe

Lehrstuhl Software Engineering

{kurpick,pinkernell,rumpe}@se-rwth.de

Abstract. Der Energie Navigator ist ein Softwaresystem für die automatisierte Massendatenerfassung und -aufbereitung von Gebäudesensordaten. Die Software erlaubt es automatische Analysen auf Basis dieser Daten durchzuführen und bietet diverse Visualisierungsarten und Reporting-Mechanismen an. Zusätzlich können Spezifikationen von Gebäuden erstellt und automatisiert überprüft werden.

Nutzer sind Architekten und Energieexperten, die damit Gebäude auf ihren Verbrauch und Einsparpotenzial spezifizieren, analysieren und optimieren. Die Software wird zurzeit mit Partnerunternehmen in einer breit angelegten Feldstudie erprobt.

Es wird ein komplexes (automatisiertes) Buildsystem auf Ant-Basis genutzt. Eine Besonderheit bei dem Projekt ist die Entwicklung einer Backup-Komponente, die eine automatisierte Migration der Produktivdaten nach jedem Release ermöglicht. Die Entwicklung erfolgt agil und in kurzen Release-Zyklen.

Keywords

1 Einleitung

Der Energie Navigator ist ein Softwaresystem für die automatisierte Massendatenerfassung und -aufbereitung von Gebäudesensordaten. Die Software erlaubt es automatische Analysen auf Basis dieser Daten durchzuführen und bietet diverse Visualisierungsarten und Reporting-Mechanismen an. Zusätzlich können Spezifikationen von Gebäuden erstellt und automatisiert überprüft werden.

Die Struktur des vorliegenden Beitrags gliedert sich wie folgt: Zunächst wird das Projekt vorgestellt. Danach wird eine kurze Übersicht über die Architektur des Energie Navigators und die verwendeten Technologien im Projekt skizziert. Im Anschluss wird auf den Entwicklungsprozess im Projekt eingegangen. Als Nächstes wird auf die Herausforderungen auf der menschlichen als auch der technologischen Seite eingegangen und mit einer zusammenfassenden Einschätzung abgeschlossen.

2 Der Energie Navigator

Der Energie Navigator ist ein System für die Spezifikation, Analyse und Optimierung von Gebäuden und technischen Anlagen [1]. Im Mai 2010 wurde die synavision GmbH mit dem Ziel der Weiterentwicklung des Systems zur Produktreife gegründet. Das erste Release (Version 1.0) wurde im Mai 2011 veröffentlicht.

Zum Zeitpunkt der Veröffentlichung dieses Beitrags befindet sich das System bei mehreren Firmen in der Erprobungsphase. Gleichzeitig wird am Release 2.0 des Energie Navigators weitergearbeitet und dieser ständig um neue Konzepte und Funktionen erweitert.



[KPR11] T. Kurpick, C. Pinkernell, and B. Rumpe
Der Energie Navigator
In: Entwicklung und Evolution von Forschungssoftware, Tagungsband, Rolduc, 10.-11.11. 2011,
Shaker Verlag, Aachen, ISBN 978-3-8440-1600-0,
Aachener Informatik-Berichte, Software Engineering Band 14. 2013.
www.se-rwth.de/publications

Die Begleitforschung für das System ist durch Mittel des BMWi und der EU gefördert worden. Weitere Förderanträge für neuartige Konzepte sind bereits in Planung.

Der Energie Navigator wird zusammen mit dem Institut für Gebäude- und Solar-technik der TU Braunschweig und der energydesign braunschweig GmbH entwickelt, die für die fachlichen Konzepte der Anwendungsdomäne verantwortlich sind. Die Planung, die Entwicklung und das Management werden in Aachen vom Lehrstuhl Software Engineering der RWTH und der synavision GmbH durchgeführt. Zum aktuellen Zeitpunkt sind an der Entwicklung ca. 20 Personen (4-5 Assistenten, ca. 13-17 Studenten) beteiligt.

Das System wird von den Nutzern positiv bewertet. Es werden regelmäßig Verbesserungsvorschläge und zusätzliche Anregungen mitgeteilt.

3 Architektur der Energie Navigator Plattform

In der Abbildung 1 ist die Architektur des Energie Navigators auf hoher Abstraktionsebene zu sehen. Das System teilt sich in die folgenden Komponenten:

- *Data Import*: Importieren der Massendaten in das System
- *Preprocessing*: Bereinigen der Massendaten von Fehlern
- *Specification*: Anlegen und Verwalten der Gebäudespezifikation
- *Analysis*: Auswertung der Massendaten auf Basis der Spezifikation
- *Reporting*: Erstellen von Berichten mit den Ergebnissen der Analyse

Weitere Informationen über die Energie Navigator Plattform finden sich unter [1, 2, 3, 4].

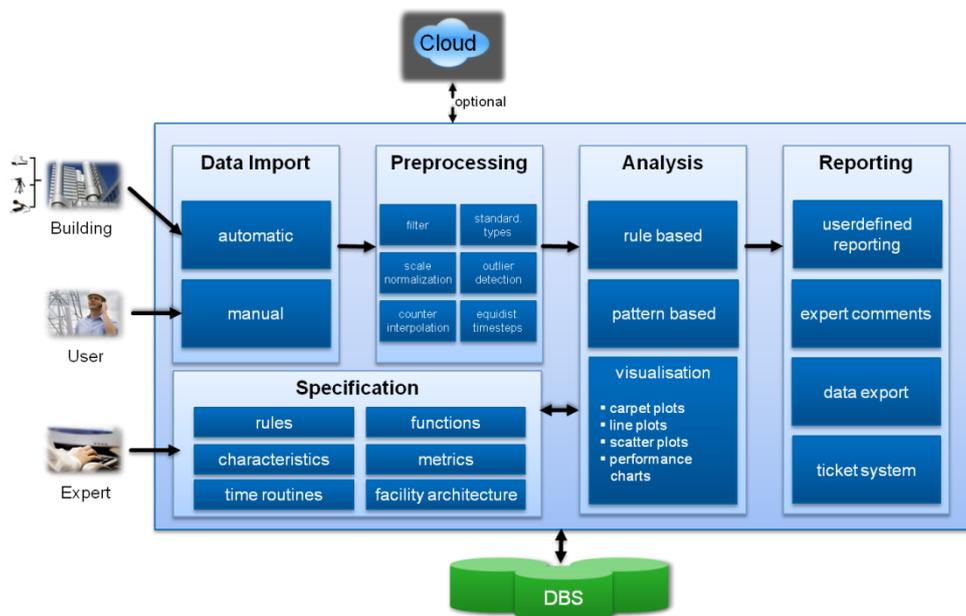


Abbildung 1: Architektur des Energie Navigators

4 Technologie

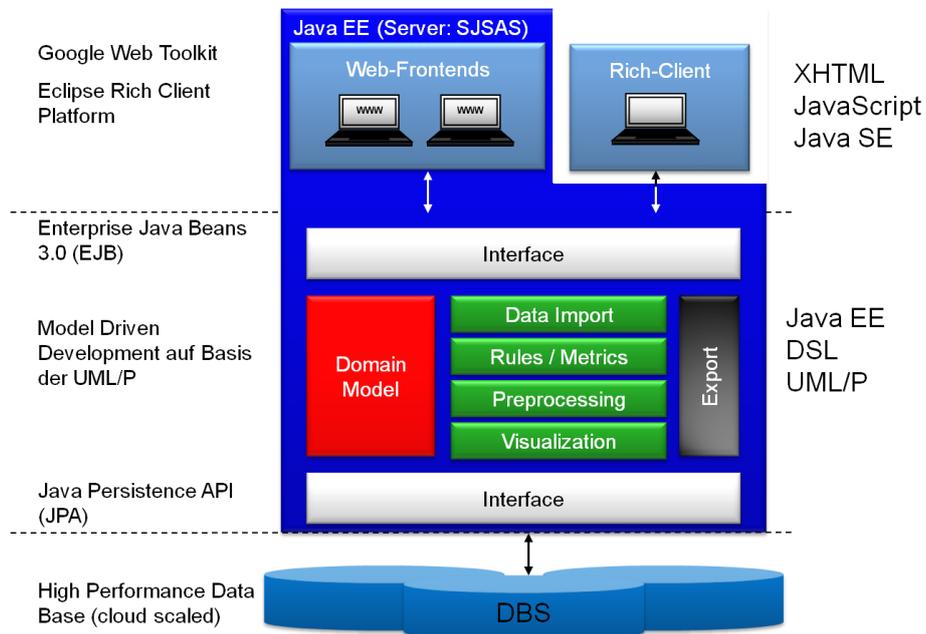


Abbildung 2: Eingesetzte Technologien

In der Laufzeitumgebung des Energie Navigators werden viele verschiedene Technologien eingesetzt. In der Abbildung 2 sind die wichtigsten aufgeführt.

Auf der Serverseite werden Java EE, PostgreSQL und massendatenfähige Datenbanken eingesetzt. Auf der Frontend-Seite werden neben einer Rich Client Plattform (RCP) Anwendung auch Web-Applikationen mit dem Google Weg Toolkit (GWT) entwickelt. Die Software wird modellbasiert auf Grundlage eigener DSLs entwickelt, die mit Hilfe des DSL-Frameworks Monticore [5] erstellt wurden.

5 Infrastruktur

Die Entwicklungs- und Erprobungsinfrastruktur des Energie Navigators besteht aus virtuellen Maschinen. Es gibt zwei Server für Produktivsysteme, zwei Server für die kontinuierliche Integration sowie Nightly-Builds und ca. 25 virtuelle Entwicklungsumgebungen. Die virtuellen Entwicklungsumgebungen sind vorkonfiguriert, so dass neue Entwickler sich keine eigene Umgebung für das System erstellen müssen. Der Zugriff erfolgt entweder über einen Remote-Zugang auf den Maschinen oder die virtuelle Maschine wird von den Entwicklern lokal ausgeführt.

Neben den virtuellen Maschinen wird das Projektportal SSELab für die Entwicklung verwendet [6]. Von den angebotenen Services des SSELabs werden die Services Subversion, Trac, Wikis, Mailing-Liste und Feedback-System [7] eingesetzt.

Da die Plattform des Energie Navigators in mehrere Komponenten gegliedert ist, wurde auch viel Aufwand in die Build-Skripte des Systems gesteckt. Mit Hilfe der Build-Skripte ist es möglich, automatisiert die IDE zu installieren sowie das Produktiv-System und das Nightly-Build-System zu erstellen.

Die Buildskripte dienen auch dazu, die Plattform auf mehreren virtuellen Maschinen auszuliefern und zu installieren. Ziel dieser Build-Infrastruktur ist es, den Entwicklern zu ermöglichen, die Plattform zu bauen, auszuliefern und auszuführen, ohne Wissen über das ganze System zu haben.

6 Entwicklungsprozess

Der Entwicklungsprozess des Energie Navigators ist iterativ. Im Moment werden zwei unterschiedliche Entwicklungszweige gepflegt. Der Zweig für die Version 1.0, die schon bei Kunden produktiv eingesetzt wird, und der Zweig 1.1 der neue Funktionen beinhaltet und von einigen Kunden getestet wird. Jeden Monat wird ein neues Release für die Version 1.0 und 1.1 erstellt und ausgeliefert (*Release often*). Der Release-Prozess ist automatisiert, so dass für ein Release des Systems nur ca. eine Stunde gebraucht wird (*Release fast*).

Die Planung des Systems erfolgt über die Releasezyklen. Dafür wird das Ticketsystem Trac als Managementwerkzeug eingesetzt. Im Projekt gibt es verschiedene Rollen, die in der Entwicklung für Aufgaben oder Komponenten verantwortlich sind, z.B. eine studentische Hilfskraft für die Tests des Systems. Dieser Student überwacht die Testausführung im Continuous Integration System, erstellt neue Test-Datensätze für die Integrationstests und erhöht die Testfallabdeckung in den einzelnen Komponenten.

Jede Komponente des Systems hat einen wissenschaftlichen Mitarbeiter als Komponentenverantwortlichen. Er überwacht die Qualität der Komponente und fällt Designentscheidungen für die Implementierung und Weiterentwicklung.

Als zusätzliche Aufgaben fallen in dem Projekt das Management der Entwicklungstätigkeiten, das Sammeln und Bewerten von Rückmeldungen der Kunden und das Zuweisen von offenen Tickets an die Entwickler an.

7 Herausforderungen

Das Projekt bietet zwei unterschiedliche Herausforderungen. Auf der einen Seite ist das Management der beteiligten Personen und des Teams eine Herausforderung und wird durch den universitären Kontext zusätzlich erschwert; auf der anderen Seite führt die Evolution des Systems und des Projekts zu technischen Problemen, die aktiv angegangen werden müssen.

7.1 Menschen

Neben den wissenschaftlichen Mitarbeitern, die lange im Projekt arbeiten, sind auch studentische Hilfskräfte, Auszubildende und Studenten, die ihre Bachelor-, Master- und Diplom-Arbeit im Rahmen des Projekts erstellen, beteiligt. Gerade die Studenten, die ihre Studienarbeiten schreiben sind nur sehr kurz im Projekt involviert (ca. 4-6 Monate im Durchschnitt). In dieser Zeit muss sich der Student mit dem System fachlich und technisch vertraut machen, um anschließend seine Aufgabe zu lösen und geeignet in die Plattform zu integrieren.

Um eine möglichst effektive Entwicklung zu gewährleisten, hat das Projekt zwei Projektträume eingerichtet. In diesen Räumen ist Platz für 12 Entwickler. Jeder Platz

hat einen eigenen PC, bietet aber auch Anschlussmöglichkeiten für Laptops. Während der Woche gibt es drei feste Zeiträume in denen sich Mitarbeiter des Lehrstuhls und Studenten in diesen Räumen treffen und zusammen entwickeln oder die nächsten Schritte besprechen. Von jedem Beteiligten wird erwartet mindestens einmal pro Woche an einem Treffen teilzunehmen.

Die gemeinsame Zeit wird hauptsächlich genutzt, um die Architektur der verschiedenen Komponenten und Implementierungsaufgaben zu besprechen. Zusätzlich sollen durch die Treffen technische Probleme schnell behoben werden, die während der Entwicklung entstehen und mit der komplexen Infrastruktur zusammenhängen. Neben den wöchentlichen Treffen finden auch gelegentlich interne Vorträge zu speziellen Themen statt. Als Beispiel sind hier Vorträge zum „Einsatz von Sonar“, „Coding Guidelines“ und Architektur Reviews genannt. Neben den fachlichen Treffen finden aber auch Treffen für das Teambuilding statt; es wurde u.a. eine große Release-Party für das Release 1.0 gefeiert. Außerdem findet jährlich ein Sommergrillen mit allen Entwicklern statt.

7.2 Evolution des Projekts

Die Evolution des Systems und des Projekts muss aktiv gemanagt werden. Am Anfang dauerte ein Release des Systems ca. einen Arbeitstag. Durch mehrere studentische Arbeiten wurden die einzelnen gewachsenen Buildskripte der unterschiedlichen Komponenten auf eine gemeinsame Basis gestellt, und so der Release-Prozess auf ca. eine Stunde reduziert. So wurde auch der Release-Zyklus von 4-5 Monaten zwischen zwei Releases auf einen Monat verkürzt.

Kontrovers wurde die Wahl zwischen Ant und Maven als Buildtool diskutiert. Im Rahmen des Projekts wurde sich für Ant entschieden, da hier die Anpassungen an den Projektkontext leichter zu bewerkstelligen waren als der Umstieg auf Maven.

Durch die Erhöhung des Release-Taktes wurde es nötig, den Kunden auch die Möglichkeit der Datenmigration zu geben. Gerade am Anfang des Projekts wurden noch häufig Änderungen am Datenmodell vorgenommen, so dass Daten nicht ohne weiteres übernommen werden konnten. Deshalb wurde eine Komponente für die Datenmigration zwischen mehreren Releases erstellt, die eine automatische Migration der alten Daten erlaubt.

Durch die verteilte Entwicklung mit den Entwicklern in Aachen und den fachlichen Experten in Braunschweig wurde auch die Feedback-Komponente [7] integriert. Sie ermöglicht dem Endbenutzer im Energie Navigator, Rückmeldungen für die Entwickler zu erstellen. Für die entfernte Kommunikation werden Skype und Teamviewer eingesetzt.

Am Anfang wurden Modelle im Projekt nur in der Analyse genutzt. In den gemeinsamen Treffen wurden Klassen-, Sequenzdiagramme oder Automaten an der Tafel erstellt und diskutiert, um anschließend die Implementierung von Hand zu programmieren. Im weiteren Verlauf des Projekts wurde dann auf eine modellbasierte Entwicklung umgestellt, die aus den Modellen die Implementierung generiert.

Dabei wurde festgestellt, dass der Austausch von handgeschriebenem Code durch generierten Code nicht ohne weiteres möglich ist. Die Generatoren lassen sich gut für neue Komponenten des Systems einsetzen, aber schlecht für die Migration. Grund

sind Sonderfälle, die bei der manuellen Implementierung umgesetzt wurden, nicht aber bei der Generierung.

8 Zusammenfassende Einschätzung

Das Projekt des Energie Navigators erlaubt mit seinem Entwicklungsprozess eine schnelle Entwicklung von neuen Features. Die festen Arbeitszeiten/-treffen führen im Projekt zu einer guten Kommunikation zwischen allen Beteiligten. Das agile Vorgehen ermöglicht Entscheidungen schnell zu treffen und auf geänderte Anforderungen der Kunden zu reagieren.

Es gibt aber noch einige negative Punkte in dem Projekt, die zu verbessern sind. Die mittlerweile komplexe Infrastruktur des Projekts erfordert einen hohen Einarbeitungsaufwand. Die Qualität der Entwicklung leidet unter den vielen studentischen Entwicklern, die selbst nur für kurze Zeit in dem Projekt mitarbeiten. Die Nutzung des Continuous Integration Systems für jeden Commit ist nicht sinnvoll, da mittlerweile die Generierungs-/Kompilierungszeit des ganzen Systems länger als 20 Minuten dauert. Im Moment sind deshalb weitere Arbeiten für die Build-Infrastruktur geplant, die beide Punkte erheblich verbessern sollen.

9 References

- 1 M. N. Fisch, T. Kurpick, C. Pinkernell, S. Plesser, and B. Rumpe. The energy navigator-a web based platform for functional quality mangement in buildings. In Proceedings of the 10th International Conference for Enhanced Building Operations (ICEBO 10), Kuwait City, Kuwait, 2010.
- 2 M. N. Fisch, M. Look, S. Plesser, C. Pinkernell and B. Rumpe, Der Energie-Navigator - Performance-Controlling für Gebäude und Anlagen. In Technik am Bau (TAB) - Fachzeitschrift für Technische Gebäudeausrüstung, 04/2011:36-41, bau verlag, March 2011.
- 3 M. N. Fisch, S. Plesser, C. Pinkernell and B. Rumpe, Software für die energieoptimierte Betriebsführung von Gebäuden. In BINE Informationsdienst Projektinfo (ISSN 0937-8367), 14/10:1-4, October 2010.
- 4 M. N. Fisch, S. Plesser, C. Pinkernell and B. Rumpe, Virtueller Prüfstand für Gebäude: Der Energie Navigator. In XIA Intelligente Architektur (ISSN 0949-2356), Vol. 73, 10-12/10:68-70, October 2010.
- 5 H. Krahn, B. Rumpe, and S. Völkel. MontiCore: a Framework for Compositional Development of Domain Specic Languages. International Journal on Software Tools for Technology Transfer (STTT), 12(5):353-372, September 2010.
- 6 C. Herrmann, T. Kurpick, B. Rumpe SSELab: A Plug-In-Based Framework for Web-Based Project Portals In: Proceedings of the 2nd International Workshop on Developing Tools as Plug-Ins (TOPI 2012) at ICSE 2012, June 3, Zurich, Switzerland, 2012.
- 7 C. Herrmann, T. Kurpick, and B. Rumpe, Agile User-Feedback and its Management through the SSELab Feedback System, RWTH Aachen, Tech. Rep. AIB-2012-11, 2012. [Online]. Available: <http://aib.informatik.rwthachen.de/2012/2012-11.pdf>