

BAF – Building Automation Framework

Thomas Jonitz
jonitz@iaib.de

30. September 2013

Zusammenfassung

Entwurf und Konzeptionierung eines hardwareabstrahierenden Frameworks für die Gebäudeautomation.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Über das Institut	1
1.2	Hintergrund und Notwendigkeit	2
2	Copyright	3
3	Anforderungsanalyse	3
3.1	Allgemeine Framework-Anforderungen	3
3.2	Anforderungen aus Anwendersicht	4
3.3	Funktionale Anforderungen	5
4	Konzept	5
4.1	Entwurfsskizze	5
4.2	Grobentwurf	5
4.3	Feinentwurf	5
5	Auswertung der Ergebnisse	5
5.1	Betrachtung des Konzepts	5
5.2	Realisierungsvorschläge	5

1 Einleitung

1.1 Über das Institut

Das *Institut für angewandte Informatik im Bauwesen e.V.* (IAIB) ist seit 1997 auf den Gebieten Gebäudeenergieeffizienz, Facility Management (FM) sowie in der Weiterbildung etabliert. Das Institut kann auf intensive Partnerschaften mit nationalen und internationalen Verbänden, Aktivitäten in verschiedenen Initiativen und überregionale Kooperationen mit Unternehmen im ganzen Bundesgebiet verweisen.

Zahlreiche durch das IAIB e.V. durchgeführte und geleitete Initiativen und die im Bereich Energiemanagement, Facility Management und Weiterbildung durchgeführten Projekte haben unserem Institut einen sehr guten fachlichen Ruf erbracht. Als gemeinnütziges, unabhängiges Forschungs- und Entwicklungsinstitut und Kooperationspartner der Hochschule Wismar (Aninstitut) genießt unser Institut hohes Vertrauen in Punkto Unabhängigkeit und Seriosität. Dieses spiegelt sich ebenfalls in unserer absoluten Unabhängigkeit von wirtschaftlichen Institutionen (beispielsweise Produktherstellern) wieder.

Der gemeinnützige Verein trug durch diverse Forschungs- und Entwicklungsarbeiten sowie Aus- und Weiterbildungsmaßnahmen in den vergangenen Jahren maßgeblich zur Entstehung internationaler Patente, innovativer Produkte und Erkenntnisse bei.

1.2 Hintergrund und Notwendigkeit

Im Zuge der *Energiewende*, vorallem vor dem Hintergrund der Energie-Einspar-Verordnung (EnEV), dem Energie-Einspar-Gesetz (EnEG) und dem Erneuerbare-Energien-Gesetz (EEG) in Deutschland, müssen viele Betreiber von Gebäuden die energetischen Parameter ihrer Liegenschaften anpassen. Den hohen Investitionskosten stehen Fördermittel der EU, dem Bund, den Ländern und den Kommunen gegenüber. Einen ökonomischen Sinn ergeben energetische Baumaßnahmen zwar nur, sofern sie in den Lebenszyklus der Gebäude und dessen technische Ausstattung (TGA) passen. Dennoch ist in dieser Branche ein wachsender Markt zu beobachten, was an dem Preisanstieg zu beobachten ist und auf die gestiegene Nachfrage an energetischen Maßnahmen zurückgeführt werden kann.

Der Klimaschutz stellt jedoch nicht den einzigen Vorteil dar. So stehen den Investitionen in „moderne“ Anlagen auch hohe Einsparpotentiale gegenüber; sowohl Energieeinsparungen, als auch –daraus resultierend– Kosteneinsparungen. Diese Einsparpotentiale werden nicht nur durch die Leistungsfähigkeit und/oder den geringeren Energiebedarf definiert, sondern können durch effiziente und intelligente Softwaresysteme weiter ausgeschöpft werden. Dies zeigen Fallstudien und Pilotprojekte, wie etwa die Implementierung von „Heat-Off“, einem am IAIB entwickelten Algorithmus, mit dem an der Hochschule Wismar Energieeinsparungen von bis zu 70% zu verzeichnen sind.

Nachteilig am Einsatz von TGA-unterstützenden Softwaresystemen sind die (relativ) hohen Entwicklungskosten. Aufgrund fehlender Schnittstellen und Standards wird das Platzieren von „Standardsoftware“ auf dem Markt verhindert. Stattdessen finden i.d.R. maßgeschneiderte Systemlösungen Anwendung, die eine tiefe Kenntnis über die Liegenschaften und deren Ausstattung des Kunden erfordern. Gerade dieser Aspekt der fehlenden *Allgemeingültigkeit* machen die Systementwicklung für die TGA kostspielig.

Mit Hilfe des BAF-Projekts soll ein erster Schritt in Richtung einer offenen und vereinfachten Softwareentwicklung gegangen werden.

Ziel dabei ist es, die Kommunikation zwischen Software und der vielfältigen Möglichkeiten angeschlossener TGA-Systeme derart zu abstrahieren, dass einem Softwareentwickler die konkrete I/O-Implementierung verborgen bleibt und so der Fokus auf die Entwicklung des Dienstes gelegt werden kann. Dieses Ziel soll mit Hilfe eines Frameworks erreicht werden.

2 Copyright

Das nachfolgende Konzept wurde am Institut für angewandte Informatik im Bauwesen e.V. in Zusammenarbeit mit der Hochschule Wismar erstellt. Eine Weitergabe und oder Reproduktion –insbesondere im digitalen Sinne– sind – auch auszugsweise– ohne ausdrückliche, schriftliche Einwilligung des Autors und des IAIB nicht erlaubt.

3 Anforderungsanalyse

Folgende Anforderungsanalyse betrachtet sowohl die Anforderungen an das zu entwerfende Framework, als auch die Anforderungen, die an die Umgebung gestellt werden.

3.1 Allgemeine Framework-Anforderungen

Ein gutes Frameworkdesign ist durch diverse Indikatoren gekennzeichnet:

Einfachheit: Neben der hohen Leistungsfähigkeit muss ein Framework vor allem einfach gehalten werden. Immer neue Features und Funktionalitäten, die in das System eingeflochten werden sollen, überladen das Design und machen es instabil. Es ist daher wichtig ein ausgewogenes Verhältnis zwischen Leistungsfähigkeit zu finden.

Erweiterbarkeit: Ob ein Framework erweiterbar sein soll, hängt von den Entwicklern ab. Spätere Änderungen erhöhen den Funktionsumfang, oder verbessern vielleicht die Performance, jedoch leidet oftmals, durch die steigende Komplexität, die Abwärtskompatibilität zu früheren Versionen.

Integration: Ein Framework selbst ist kein lauffähiges Programm. Daher ist es essentiell notwendig, dass sich ein gutes Framework in die jeweilige Entwicklungsumgebung integriert. Dazu zählt auch, dass Konzepte der zu verwendeten Programmiersprache berücksichtigt werden. Aber auch die Entwicklungswerkzeuge, die beim Entwickeln von Anwendungen eingesetzt werden, müssen mit dem Framework arbeiten können.

Konsistenz: Frameworks müssen widerspruchsfrei sein. Finden sich logische Fehler im Design, so muss es zwingend überdacht werden, bevor es ausgeliefert wird. Wie bei anderen Produkten ist die Konsistenz eines der wichtigsten Merkmale guter Software.

Gründung auf Bewährtem: Viele Frameworks nutzen bewährte Techniken aus anderen Produkten, die lediglich eine Anpassung an die jeweiligen Bedürfnisse erfahren. Dennoch ist es wünschenswert – wenn es auch mit Vorsicht zu genießen ist – neue Lösungsansätze zu entwerfen.

Weiterhin werden ein teurer Entwurfsprozess und eine hohe Kompromissbereitschaft im Entwurf gefordert. Da diese Eigenschaften sich jedoch nicht auf das Design selbst beziehen seien sie an dieser Stelle vernachlässigt und nur der Vollständigkeit genannt.

3.2 Anforderungen aus Anwendersicht

Der Anwender ist in diesem Fall ein Softwareentwickler, der das Framework zur Implementierung von TGA-Software nutzt. Da das Framework bestandteil von TGA-Software ist und sein designiertes Einsatzgebiet im Bereich der eingebetteten Systemen liegt, muss es die Grundsätzlichen Anforderungen an Software für Embedded-PCs erfüllen. Dabei muss ein besonderes Augenmerk auf den Ressourcenverbrauch, die Portabilität, die Zuverlässigkeit und Stabilität, sowie die Flexibilität und Skalierbarkeit gelegt werden.

Ressourcenverbrauch: Grundsätzlich sollte der Ressourcenverbrauch so gering wie möglich gehalten werden. Moderne Embedded-PCs verfügen in der Regel über relativ viel Hauptspeicher (zwischen 64 MiB und 512 MiB). Im Hinblick auf die Erweiter- und Skalierbarkeit darf das Framework nicht den Großteil der verfügbaren Ressourcen belegen.

Zuverlässigkeit: Im Betrieb muss das System autonom handeln können. Ein eingreifen des Nutzers zur Laufzeit ist nicht vorgesehen. Im Störfall muss das System den Fehler automatisch erkennen und beheben. Fraglich ist, ob diese Aufgabe an das Framework delegiert werden sollte.

Flexibilität: Frameworks bieten einer Software einen vorgefertigten Funktionsumfang. Sie kennen allerdings nicht das System, in dem sie eingesetzt werden. Es ist daher notwendig, sich nicht auf einen Einsatzzweck zu konzentrieren, sondern ein definiertes Verhalten so allgemein und abstrakt wie möglich zu definieren.

Skalierbarkeit: Sofern das eingebettete System es zulässt, sollte das Framework keine Einschränkungen bezüglich der Skalierbarkeit besitzen. Die Skalierbarkeit ist demnach abhängig von der Einsatzhardware und dem Einsatzsystem.

Portabilität: Wünschenswert ist es das Framework auf verschiedene Systeme zu portieren, was in dieser Version noch nicht geplant ist. Das Framework wird aufgrund der weiten Verbreitung und der Offenheit, primär für Linux-, respektive unixähnliche -Systeme entworfen. Ein Einsatz auf einer – beispielsweise – Microsoft Windows[®]-Plattform wäre in späteren Versionen ebenso denkbar.

3.3 Funktionale Anforderungen

4 Konzept

Die Konzeptionierung des Frameworks erfolgt in drei Schritten. Dabei wird zunächst eine *Skizze* angefertigt, wie das Framework zu realisieren ist. Im Anschluss wird ein *Grobentwurf* erstellt, der die beteiligten Objekte und deren Zusammenwirken beschreibt. Im letzten Schritt, dem *Feinentwurf*, werden Verhaltensweisen beschrieben, die für die korrekte Arbeit des Frameworks ausserhalb der Objektsicht wichtig sind. So zum Beispiel das Instanzieren der Objekte.

4.1 Entwurfsskizze

4.2 Grobentwurf

4.3 Feinentwurf

5 Auswertung der Ergebnisse

5.1 Betrachtung des Konzepts

5.2 Realisierungsvorschläge