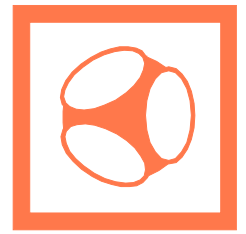


brainbot
TECHNOLOGIES AG

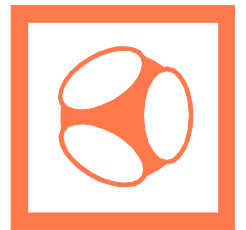
NEXTBOT

KNOWLEDGE FRAMEWORK

HEIKO HEES
13.4.2010



1 Einleitung	4
1.1 Zielsetzung Nutzerakzeptanz.....	5
1.2 Zielsetzung Integration.....	5
1.3 Zielsetzung dezentrales Wissensmanagement.....	6
2 Technologie	7
2.1 nextbot Suche	7
2.1.1 Assoziative Suche.....	7
2.1.2 Natürlichsprachige Suche.....	7
2.1.3 Bool'sche Suche.....	7
2.1.4 Attributeinschränkungen.....	8
2.1.5 Konzeptsuche.....	8
2.1.6 Fehlertolerante Suche.....	8
2.1.7 Spellcheck.....	8
2.1.8 Suche nach ähnlichen Dokumenten.....	8
2.1.9 Relevance Feedback.....	9
2.1.10 Personalisiertes Ranking.....	9
2.1.11 Passage Retrieval.....	10
2.2 Taxonomien	10
2.3 nextbot Klassifikator	11
2.3.1 Training.....	12
2.3.2 Klassifikation.....	12
2.3.3 Performance, Skalierbarkeit, Qualität.....	13
2.4 nextbot Clusterer	14
3 Architektur	16
3.1 Komponenten zur Informationsorganisation	18
3.1.1 Index – Services.....	18
3.1.2 Mediator.....	19
3.1.3 XVFS Service.....	20
3.1.4 Konnektoren.....	20
3.1.5 nextbot-Repository.....	21
3.1.6 Google Index – Service.....	21
3.2 Komponenten zur Systemorganisation	22
3.2.1 Directory – Service.....	22
3.2.2 Authentication – Service.....	23
3.2.3 Certificate – Service.....	23



3.2.4 Proxy Service.....	23
3.3 Komponenten zur Integration in Lösungen.....	24
3.3.1 Nextbot Web-Services / SOAP-API.....	24
3.3.2 WebDAV Server.....	24
3.3.3 Windows Shell Integration / COM-Service.....	24
4 Entwicklungsstandards.....	26
4.1 Programmiersprachen.....	26
4.2 XP als Vorgehensmodell.....	26
4.3 Qualitätsmerkmale nach ISO 9126.....	27
4.4 Plattformen.....	27

1 Einleitung

Dieses Dokument beschreibt das *nextbot – Knowledge Framework (nextbot - KF)* der brainbot technologies AG. Es werden die zugrunde liegenden Technologien sowie die Architektur des Systems beschrieben.

Zielgruppe sind Personen, die die Technologie und deren Integration in Lösungen verstehen und bewerten können wollen.

Das *nextbot - Knowledge Framework* integriert drei brainbot Technologien:

- nextbot Suche*
- nextbot Klassifikator*
- nextbot Clusterer*

Das *nextbot - Knowledge Framework* basiert auf Komponenten, die im Zusammenspiel einen flexiblen und leistungsfähigen Rahmen bilden, um individuelle Lösungen zur Erschließung, Vernetzung und Auswertung von Informationen in Organisationen zu ermöglichen. Insbesondere ermöglicht das *nextbot – Knowledge Framework* den Aufbau dezentraler IT-gestützter Wissensmanagementlösungen.

Die wesentlich bereitgestellten Funktionalitäten sind:

- Einheitliche Sichtweise auf Informationen verschiedenster Quellen
- Intelligente assoziative Volltextsuche
- Ordnerhierarchien (Taxonomien)
- Automatische Einordnung von Dokumenten in vorgegebene Ordner (Kategorisierung)
- Automatisches Bilden von Ordnern mit zusammengehörigen Dokumenten (Clustering)

Hinter der Konzeption des Systems stehen drei Zielsetzungen. Zum einen soll für den einzelnen Anwender die Effizienz beim Umgang mit Informationen maximiert werden. Die zweite Zielsetzung besteht darin, den Aufbau dezentraler Wissensmanagementlösungen zu ermöglichen. Das dritte Ziel ist die einfache Integrierbarkeit der Technologie in gewachsene IT-Infrastrukturen.

1.1 Zielsetzung Nutzerakzeptanz

Die Akzeptanz eines Systems zum Informationsmanagement lässt sich an der Erfüllung folgender Kriterien bewerten:

Verfügbarkeit

Ist die gesuchte Information im System verfügbar? Können Informationen einfach zur Verfügung gestellt werden?

Zugriff

Wie einfach kann die gesuchte Information gefunden und als relevant erkannt werden?

Kontext

Sind Informationen in ihrem Kontext verfügbar? Können sie in Bezug zu anderen Informationen gesetzt werden?

Integrität

Wie verlässlich ist die Information? Kann zwischen seriösen und weniger verlässlichen Informationen unterschieden werden?

Komfort

Wie einfach ist es Informationen zu bekommen und zu teilen?

Die Architektur und Technologie des Frameworks wurde so gewählt, dass obige Kriterien optimal erfüllt werden können.

1.2 Zielsetzung Integration

Unterschiedliche Organisationen haben sehr verschiedene komplexe Anforderungen an ein Informationsmanagement, für welches es viele verschiedene Lösungen gibt, wie: DMS, CMS, Fileserver, Notes, Exchange, etc.

In vielen Organisationen bestehen mehrere solcher Systeme nebeneinander, sind aber nicht zueinander kompatibel und nicht in der Lage miteinander zu kommunizieren.

Um den optimalen Nutzen aus den vorhandenen, aber verteilt vorliegenden Informationen ziehen zu können, benötigt man eine Technologie, welche die Systeme integriert und es erlaubt, Informationen übergreifend zu organisieren, Zusammenhänge zu entdecken und Informationen zu verknüpfen.

Das *nextbot – Knowledge Framework* ermöglicht den Aufbau von Systemen, die eine einheitliche Abfrage über, sowie die Navigation in allen Informationsquellen einer Organisation erlauben.

1.3 Zielsetzung dezentrales Wissensmanagement

Im Gegensatz zu anderen Produkten ermöglicht das *nextbot – Knowledge Framework* neben dem Aufbau klassischer zentraler Wissensmanagement Lösungen auch die Etablierung von dezentralen Systemen, welche dem Nutzer eine transparente einheitliche Sicht auf alle verfügbaren Informationen ermöglichen.

Bottom-Up Fähigkeit

Zentralistisch organisierte Wissensmanagement-Lösungen erfüllen aus verschiedenen Gründen nicht die in sie gesetzten Erwartungen. Wesentlich ist, dass die Systeme nicht im Kontext des Nutzers arbeiten, sondern Informationen fern vom Erzeuger und Konsument verwalten. Es gibt keine Anbindung der konkreten Informationsobjekte des Anwenders (lokale Dokumente, Emails, Bookmarks) an die offiziellen, autorisierten Informationen im Wissensmanagementsystem der Organisation.

Das *nextbot – Knowledge Framework* unterstützt den Aufbau dezentraler, verteilter Wissensmanagementlösungen, die evolutionär wachsen können. Hierzu können an Individuen gebunden „Personal Memories“ zu einem übergreifenden „Organisational Memory“ zusammengeschlossen werden.

Performance und Skalierbarkeit

Die Performance des Systems skaliert linear mit der Zahl der angebundenen Teilnehmer und verfügbaren Dokumente. In zentralen Szenarien können beliebig große Dokumentkollektionen über mehrere Rechner verteilt werden. In dezentralen Szenarien bringen neue Teilnehmer ihren PC oder Laptop als Hardwareressourcen mit ein.

Sicherheit

Die Kommunikation aller Komponenten im System ist über SSL v3 (TLS v1/ RSA 2048 BIT) verschlüsselt. Die Authentifizierung erfolgt über signierte Zertifikate. Somit wird die Kommunikation von verteilt über Unternehmensgrenzen hinweg arbeitenden Systemen auf Basis des *nextbot – Knowledge Frameworks* den hohen Sicherheitsanforderungen von Organisationen gerecht. Weiterhin ist eine feinkörnige Rechtevergabe an Nutzer oder Gruppen für alle Informationsobjekte möglich.

2 Technologie

2.1 nextbot Suche

Die nextbot Suche ist das Herz des Frameworks. Sie erlaubt die automatische inhaltliche Verknüpfung aller Dokumente, Kategorien und Teilnehmer untereinander.

2.1.1 Assoziative Suche

Klassische Volltextsuchen liefern zu einem gegebenen Suchbegriff alle passenden Dokumente zurück. Die assoziative Suche arbeitet im Standardfall vergleichbar, bietet darüber hinaus jedoch viele weitere intelligente Funktionalitäten: angefangen von unscharfen Suchen über Relevanzbewertung, Ähnlichkeitssuchen bis hin zur kontextabhängigen Gewichtung und Sortierung der Ergebnisse.

Kern der assoziativen Suche ist die Extraktion von Konzepten aus Dokumenten. Diese Konzepte beschreiben die wesentlichen inhaltlichen Merkmale eines jeden Dokuments.

Die assoziative Suche arbeitet sprachunabhängig. Es können Dokumente verschiedener Sprachen im selben Index verwaltet werden.

Die assoziative Suche ist auf die Komponenten *Index – Service* und *Mediator* verteilt.

2.1.2 Natürlichsprachige Suche

Suchanfragen können natürlichsprachig ohne Kenntnis einer speziellen Anfragesprache gestellt werden. Die Suchanfrage „Informationen zu Sicherheitsaspekten in der Spritzgussfertigung“ liefert beispielsweise alle Dokumente zurück, die zu diesem Thema relevant sind.

2.1.3 Bool'sche Suche

Die bool'sche Suche erlaubt eine Stichwortsuche genauer zu spezifizieren in dem die bool'schen Verknüpfungen UND, ODER, NICHT berücksichtigt werden.

Weiterhin kann nach Phrasen, wie z.B. „knowledge management“ gesucht werden.

Beispiel für eine komplexe bool'sche Suchanfrage:

+ („knowledge management“ KM) +consulting –(portal centralized)

Bool'sche Suchen werden vom *nextbot – Knowledge Framework* nur aus Gründen der Vollständigkeit unterstützt. Darüber hinaus werden Verfahren zur Verfügung gestellt, die das Formulieren solcher komplexen Suchanfragen überflüssig machen.

2.1.4 Attributeinschränkungen

Alle Suchanfragen können über eine beliebige Kombination von Attributen, wie z.B. Autor, Datum, Größe, Dateityp sowie eine oder mehrere Kategorien eingeschränkt werden.

2.1.5 Konzeptsuche

Anfragen und Dokumente können sich von der verwendeten Terminologie her vollständig unterscheiden und doch auf konzeptioneller Ebene genau zueinander passen. Dieses Problem wird über eine auf statistischen Verfahren beruhende Konzeptsuche gelöst, die Ähnlichkeiten von Bedeutungen automatisch anhand der verwalteten Dokumente und jeweiligen Anfragen lernt.

2.1.6 Fehlertolerante Suche

Die fehlertolerante Suche ermöglicht es falsch geschriebene Wörter in Dokumenten oder Suchanfragen auf die richtige Schreibweise abzubilden und somit fehlertolerant abzugleichen.

2.1.7 Spellcheck

Der Spellcheck ist verwandt mit der fehlertoleranten Suche und ermöglicht dem Nutzer eine Rückmeldung über potentielle Fehler in der Suchanfrage.

2.1.8 Suche nach ähnlichen Dokumenten

Das *nextbot – Knowledge Framework* erlaubt es Anfragen in Form von Textausschnitten oder Dokumenten zu stellen. Diese Funktionalität kann eingesetzt werden, um zu einem gefundenen Dokument ähnliche Dokumente zu finden, z.B. eine frühere oder neuere Version oder eine Quelle zu einem Zitat. Weiterhin können Dubletten von Dokumenten aufgespürt werden.

Insbesondere kann diese Funktionalität genutzt werden um zu einem Dokument andere im Kontext stehende Dokumente, Kategorien oder Personen zu identifizieren und anzuzeigen.

2.1.9 Relevance Feedback

Aus Studien über die Nutzung von Internetsuchmaschinen ist bekannt, dass Suchende mit der Formulierung komplexer Suchanfragen häufig überfordert sind. Dies führt zu wenigen oder keinen gefundenen passenden Dokumenten bzw. dem vorzeitigen frustrierten Abbruch der Suche wegen zu vieler unpassender Dokumente. Neben den gewohnten Suchmechanismen muss dem Suchenden daher eine Möglichkeit angeboten werden, ohne die Formulierung komplexer Suchanfragen seinen Informationswunsch in einem iterativen Verfahren exakt zu beschreiben. Das *nextbot – Knowledge Framework* stellt hierzu das *Relevance Feedback* Verfahren bereit. Ausgangspunkt für das *Relevance Feedback* ist eine herkömmliche Suche über eine Suchanfrage. Nachdem dem Benutzer die Ergebnisliste angezeigt wurde, hat er die Möglichkeit, anhand der Kurzbeschreibung eine negative oder positive Bewertung für jedes Suchergebnis vorzunehmen. Die Bewertung der Dokumente wird daraufhin für eine sukzessive Verbesserung der Ergebnismenge herangezogen. Das System ist in der Lage, Präferenzen des Suchenden zu erkennen und diese in die Suche einfließen zu lassen. Dadurch kann der Nutzer sehr schnell und ohne manuelle Modifikation von Suchbegriffen optimal passende Dokumente ermitteln.

2.1.10 Personalisiertes Ranking

Die Suche in großen Dokumentkollektionen kann Ergebnislisten mit vielen thematisch verschiedenen Dokumenten liefern.

Das personalisierte Ranking bewertet die Dokumente bezüglich ihrer inhaltlichen Nähe zu den dem Nutzer bereits bekannten Dokumenten und kann so automatisch eine Umsortierung der Ergebnismenge vornehmen.

Die Anschlussfähigkeit der gefundenen Dokumente an die bekannten Informationen des Nutzers wird dadurch sichergestellt, dass vorrangig Ergebnisse präsentiert werden, die sich im Kontext des Nutzers befinden.

2.1.11 Passage Retrieval

Die Suche von Informationen entspricht nicht der Suche nach Dokumenten. An die Suche *nach* einem Dokument schließt sich häufig die Suche *in* einem Dokument an.

Das Passage Retrieval kombiniert beide Schritte, indem es als Ergebnis direkt die besten Passagen aus Dokumenten extrahiert und zurückliefert.

2.2 Taxonomien

Eine Taxonomie bezeichnet die Einteilung von Dingen, insbesondere Organismen, in Taxa (Gruppen). In der Biologie erfolgt diese Einteilung traditionell in einen bestimmten Rang, wie Art, Gattung oder Familie. In der Linguistik beschäftigt sich die Taxonomie mit der Segmentierung und Klassifikation sprachlicher Einheiten, um mit diesen ein Sprachsystem zu beschreiben. Auch andere Forschungsdisziplinen verwenden den Begriff der Taxonomie allgemein für ein Klassifikationssystem, eine Systematik oder den Vorgang des Klassifizierens.

Im *nextbot – Knowledge Framework* bezeichnet das Konzept der Taxonomien die Möglichkeit Dokumente explizit strukturiert in Kategorien (Ordnern) abzulegen. Taxonomien stellen hier eine Erweiterung der vom Filesystem (Explorer) bekannten Baumstruktur dar, die darin besteht, dass Dokumente wie auch Kategorien Unterelemente mehrerer anderer Kategorien sein können.

In der Praxis bieten Taxonomien den Vorteil der Mehrfachzuordnung, ohne Dubletten erstellen zu müssen. In einem klassischen Filesystem ist es z.B. nicht immer klar, ob ein *Protokoll C* zum *Projekt B* beim *Kunden A* unter */projekte/B/protokolle/C* oder */kunden/A/protokolle/C* abgelegt werden soll. In einer Taxonomie ordnet man es einfach den Kategorien *kunden/A/* UND *protokolle/* UND *projekte/B/* zu. Über die Taxonomie sind die wesentlichen Aspekte des Dokuments genau beschrieben.

Über eine Selektion mehrerer verschiedener gesuchter Aspekte kann ein Dokument ganz einfach beschrieben und in der Taxonomie gefunden werden.

In Kombination mit der assoziativen Suche spielen Taxonomien ihre volle Leistungsfähigkeit aus. So kann z.B. eine Suchanfrage über die Auswahl mehrerer Kategorien eingeschränkt werden. Als Ergebnis erhält man Dokumente, die der Suchanfrage entsprechen und in der Schnittmenge der Kategorien enthalten sind. Darüber hinaus wird die Treffermenge um Dokumente erweitert, die den kombinierten Konzepten aus Anfrage und gewählten Kategorien entsprechen.

Taxonomien sind über das Netzwerk transportabel und ermöglichen zusammen mit der automatischen Klassifikation, fremde Dokumentkollektionen aus Sicht der eigenen Taxonomie zu betrachten und umgekehrt.

2.3 nextbot Klassifikator

Klassifikation ist ein Prozess, bei dem Zuordnungsvorschläge für Dokumente zu Kategorien in einer gegebenen Taxonomie generiert werden.

Einsatzmöglichkeiten sind:

- Information Routing (z.B. Eingangspost)
- Vorschläge für Ablageorte
- Abbildung von externen Informationen in bestehende Taxonomien

Das *nextbot – Knowledge Framework* unterstützt eines der leistungsfähigsten Verfahren zur Dokumentklassifikation. Das Verfahren ist selbstlernend, d.h. dass die *Klassifikatoren* anhand von Beispieldokumenten in einer vorgegebenen Taxonomie gelernt werden (Training).

Weitere Leistungsmerkmale sind:

- Automatische Zuordnung von neuen Dokumenten zu Ordnern in der Taxonomie
- Zuordnung zu mehreren Kategorien
- Erzeugung von Vorschlägen für eine Zuordnung unter Angabe von Konfidenzwerten bezüglich der Verlässlichkeit von Klassifikationsentscheidungen
- Bidirektionaler Abgleich
- Transportable *Klassifikatoren*
- Automatisches Training der *Klassifikatoren* über Beispieldokumente
- Hierarchische Generierung von *Klassifikatoren*
- Möglichkeit *Klassifikatoren* zu analysieren und über Änderungen an der Menge der Beispieldokumente zu verbessern
- Sehr hohe Verarbeitungsgeschwindigkeit

2.3.1 Training

Das System trainiert einen Klassifikator für jede Kategorie einer Taxonomie. Basis sind die in der Kategorie enthaltenen Dokumente und Unterkategorien, aus denen die wesentlichen Konzepte extrahiert und zu einem Klassifikator zusammengeführt werden, welcher die Beispieldokumente von den restlichen Dokumenten die außerhalb der Kategorie liegen inhaltlich abgrenzt.

Zu jedem Klassifikator wird über ein so genanntes „leave-one-out-Verfahren“ ein Schwellenwert trainiert und gleichzeitig Qualitätswerte evaluiert.

Funktionale *Klassifikatoren* können schon ab der geringen Zahl von zwei Beispieldokumenten pro Kategorie erzeugt werden. Mit einer steigenden Zahl an Beispielen steigt die Qualität der erzeugten *Klassifikatoren*.

Es können globale Minimalwerte für die zu erzielenden *Precision* und *Recall* Werte gültiger *Klassifikatoren* konfiguriert werden.

Klassifikatoren lassen sich optimieren, indem man die Menge der Beispieldokumente manipuliert. Dies geschieht indem von einem Klassifikator diejenigen Beispieldokumente erfragt werden, die signifikant zu einer gegebenen Fehlklassifikation beigetragen haben. Weiterhin können Beispieldokumente erfragt werden, die den Klassifikator inhomogen gestalten oder die Trennschärfe zu anderen *Klassifikatoren* einschränken.

2.3.2 Klassifikation

Werden zu einem gegebenen Dokument Klassifikationsvorschläge erfragt, wird das Konzept des Dokuments extrahiert und gegen die Menge der verfügbaren *Klassifikatoren* geprüft.

Alle trainierten *Klassifikatoren* werden in eine invertierte Darstellung gewandelt. So ist es möglich, dass die Klassifikation von Dokumenten auch bei sehr großen Taxonomien innerhalb von Millisekunden abläuft.

Das Ergebnis ist eine Liste aller Kategorien, deren Klassifikator innerhalb eines gegeben Toleranzbereiches auf das Dokument passt. Teil des Rückgabewertes ist die Konfidenz, mit der jeder Klassifikator das Dokument als zugehörig erkannt hat, so wie die Güte (*Precision*, *Recall*) des Klassifikators.

Die *nextbot* - *Klassifikatoren* erlauben im Vergleich zu anderen Lösungen einen performanten bidirektionalen Abgleich. D.h., dass nicht nur zu einem Dokument alle passenden Kategorien gefunden werden können, sondern auch zu einer Kategorie

alle passenden Dokumente mit einer einzigen Klassifikationsanfrage zurückgeliefert werden.

Dies ist insbesondere in Verbindung mit der Möglichkeit interessant, *Klassifikatoren* über das Netzwerk zwischen verschiedenen Kollektionen transportieren zu können. *Klassifikatoren* erlauben die im Netzwerk verteilten Taxonomien verschiedener Nutzer abzugleichen und Kategorien mit ähnlichen inhaltlichen Konzepten zu identifizieren.

2.3.3 Performance, Skalierbarkeit, Qualität

Das System wurde in Szenarien mit 70.000 Kategorien und über 1.000.000 Beispieldokumenten erfolgreich eingesetzt. Bei einer Teststellung mit dieser Kollektion (Archiv eines großen Verlags) schlug die Lösung auf Basis von *nextbot – Knowledge Framework* alle drei Mitbewerber in den Kriterien Performance und Qualität mit deutlichem Abstand.

Der Klassifikationsdurchsatz des Systems ist abhängig von der Zahl der Kategorien. Auf aktueller Hardware und einer Kollektion mit 70.000 Kategorien können über 3.000 Dokumente / Minute klassifiziert werden.

Das Training der *Klassifikatoren* ist linear abhängig von deren Anzahl und der Kollektionsgröße. Auf dem oben beschriebenen System konnten ca. 3000 *Klassifikatoren* pro Stunde trainiert werden.

Trotz zahlreicher Optimierungen hinsichtlich der Performance entspricht die Qualität der *Klassifikatoren* dem, was in der Wissenschaft derzeit als machbar gilt¹.

¹ Sebastiani, Fabrizio: Machine Learning in Automated Text Categorization]

2.4 nextbot Clusterer

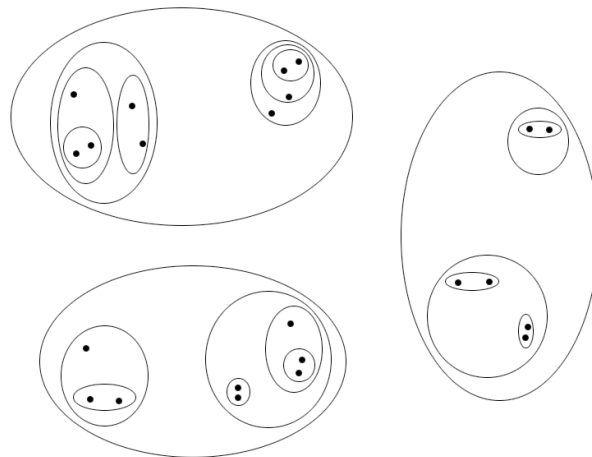
Clustering ist ein vollautomatischer Prozess, der Dokumente in Gruppen (Cluster) von inhaltlich verwandten Dokumenten zusammenfasst.

Einsatzmöglichkeiten sind:

- Gruppierung von Suchergebnismengen
- Automatischer Aufbau von Taxonomien
- Erschließung von unbekanntem Dokumentkollaktionen
- Automatische Aufteilung großer inhomogener Kategorien (ermöglicht bessere *Klassifikatoren*)
- Dublettenkontrolle

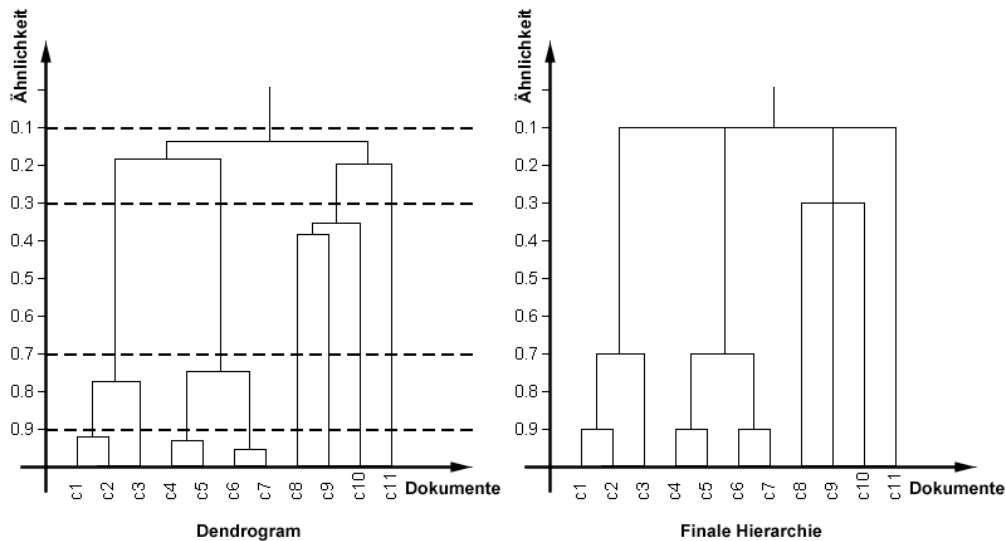
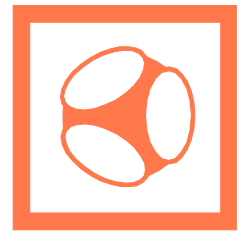
Das *Nextbot – Knowledge Framework* unterstützt ein effizientes hierarchisches Clusteringverfahren.

Über die mittlere Ähnlichkeit der Konzepte von Dokumenten zueinander, werden Cluster gebildet und die jeweils zwei (im Mittel) ähnlichsten Cluster sukzessive zu einem übergeordneten Cluster zusammengefasst.



[Sukzessives hierarchisches Clustering nach inhaltlicher Nähe]

Um eine nutzerfreundliche Darstellung und brauchbare Taxonomie zu generieren, können Schnitte bezüglich bestimmter Clusterqualitäten definiert werden - womit die Anzahl der Cluster reduziert wird. Der Sonderfall mit nur einem Schnitt erzeugt ein flaches, nicht hierarchisches Clustering.



[Schnitte zur Reduktion der Clusteranzahl]

Um dem Endanwender eine Bewertung der Cluster zu ermöglichen, wird diesen eine Liste der wichtigsten Begrifflichkeiten der enthaltenen Dokumente als Name gegeben. Weiterhin ist es möglich die enthaltenen Dokumente nach ihrer mittleren Ähnlichkeit zu den restlichen Dokumenten im Cluster zu sortieren und das für das Cluster repräsentativste Dokument anzuzeigen.

Da Dokumente mehrere Aspekte enthalten können und entsprechend auch in einer Taxonomie in mehreren Kategorien existieren sollten, kann optional in einem weiteren Schritt jedes Dokument auf passende Zugehörigkeit zu weiteren Clustern geprüft und ggf. automatisch zugeordnet werden.

Für online Applikationen kann ein genaues Clustering zu zeitaufwendig sein. Daher bietet das *nextbot – Knowledge Framework* die Möglichkeit, Clusteringprozesse innerhalb einer limitierten Zeitspanne ablaufen zu lassen und dafür auf eine maximale Präzision der Clusterbildung zu verzichten.

3 Architektur

Das System besteht aus mehreren Komponenten, die zu einer individuellen Lösung kombiniert werden können.

Kern des Systems sind ein oder mehrere *Index – Services*, welche einen Index über die Dokumente in den angebundenen *Repositories* verwalten und intelligente Such- und Klassifikationsfunktionalitäten bereitstellen. Zudem werden im *Index – Service* die Rechte, sowie der Kontext der Dokumente verwaltet.

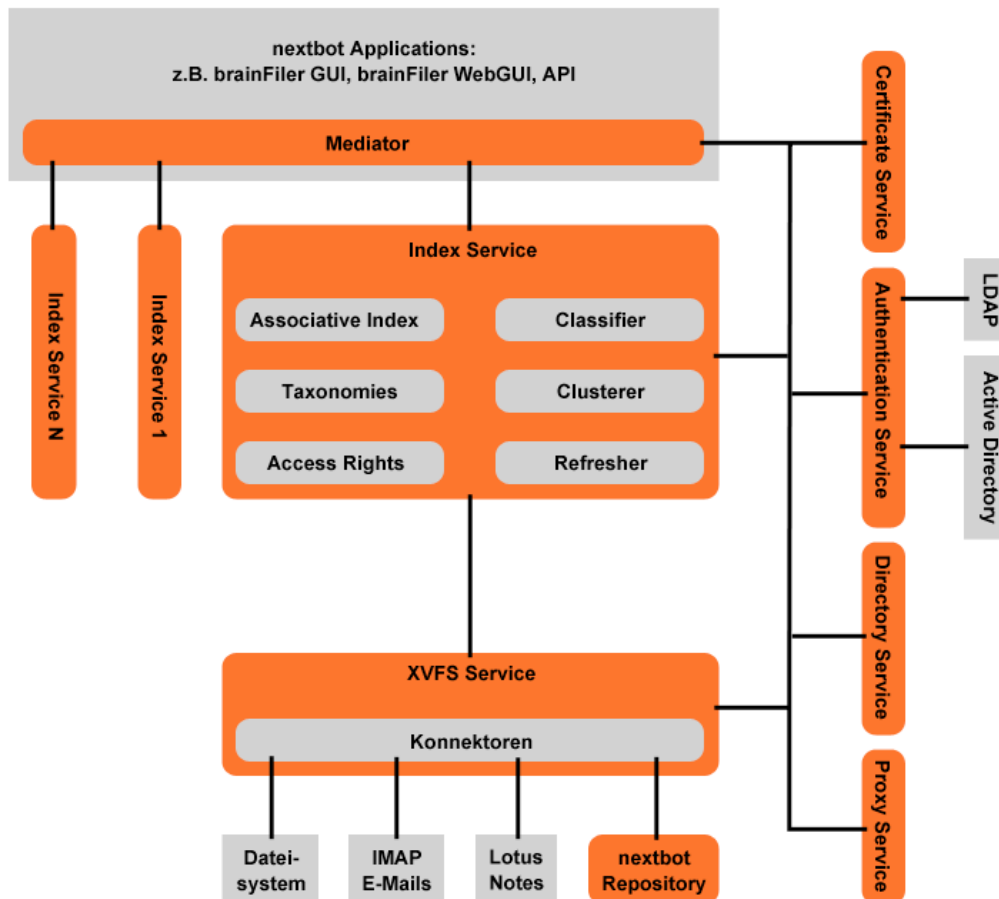
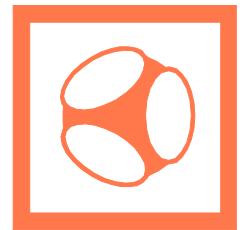
Die Architektur bietet eine hohe Flexibilität in der mehrere *Index – Services* auf verschiedenen Rechnern zu einem System mit transparentem Zugriff kombiniert werden können.

In zentralen Szenarien ist die Verteilung der verwalteten Dokumente auf mehrere *Index – Services* möglich, um:

- sehr große Datenbestände auf mehrere Rechner zu verteilen
- die Performance zu erhöhen, indem Anfragen über ein Load-Balancing auf mehrere *Index – Services* aufgeteilt werden
- hochverfügbare Lösungen aufzubauen

In dezentralen Szenarien bringt jeder Teilnehmer seine Informationen über einen eigenen *Index – Services* mit. Aus deren Gesamtheit ergibt sich ein System auf welches transparent zu gegriffen werden kann, sofern in den *Index – Services* entsprechende Freigaben existieren.

In der Regel wird eine Applikation auf Basis der *nextbot – API* entwickelt. Diese wiederum ist an einen *Mediator* gebunden, der die Anfragen transparent an alle betroffenen *Index – Services* weiterleitet. Die *Index – Services* synchronisieren Ihre Inhalte, über *Konnektoren* angebundenen *Repositories*. Ein *Directory – Service* vermittelt zwischen den Komponenten, welche sich gegenseitig über einen *Authentication – Service* authentifizieren.



[Schematische Darstellung der Architektur]

Die Komponenten tauschen Nachrichten und Arbeitsaufträge über ein verschlüsseltes Netzwerkprotokoll asynchron aus.

Der Ausfall oder das Abmelden einer Komponente hat keinen Einfluss auf die Verfügbarkeit des verbliebenen Systems.

3.1 Komponenten zur Informationsorganisation

3.1.1 Index – Services

Der *Index – Services* ist das Herzstück des Frameworks in dem folgende Funktionalitäten zusammengefasst sind:

- Ein assoziativer Volltextindex welcher die nextbot Suche und die Strukturierte Speicherung der Attribute zu Dokumenten zur Verfügung
- Verwaltung der Taxonomien und Dokumentzuordnungen
- Verwaltung der Objektrechte
- nextbot Klassifikator (wie in Abschnitt 2.3 beschrieben)
- nextbot Clusterer (wie in Abschnitt 2.4 beschrieben)
- Konzeptextraktion, Training der *Klassifikatoren*

Der *Index – Service* wird in der Regel nicht direkt, sondern über den *Mediator* oder die Synchronisation des *XVFS – Service* angesprochen.

Sicherheit

Alle Objekte sind mit dezidierten Zugriffsrechten ausgestattet. Es kann feinkörnig gesteuert werden welcher Anwender welche Operationen auf Objekten durchführen darf. Das System unterscheidet zwischen Lese-, Schreib- und Besitzerrechten. Rechte können auf untergeordnete Objekte vererbt werden.

Effizientes Trainieren

Die nötigen Rechenschritte zur Indexierung, zur Konzeptextraktion und zum Training der *Klassifikatoren* sind in kleine Zeitscheiben unterteilt, die nur ausgeführt werden, wenn das System keine externen Anfragen bearbeitet. Es kann eingestellt werden, welchen Anteil an den CPU-Ressourcen die internen Services nutzen dürfen.

Effizientes Speichermanagement

Der *Index – Services* ist darauf ausgelegt große Datenmengen effizient zu verwalten. Um mit großen Dokumentmengen (> 100.000 Dokumente) arbeiten zu können, nutzen der Index und die eingebettete Datenbankbibliothek *Memory Mapped Files*. Dies bedeutet, dass der Service den verfügbaren Hauptspeicher optimal nutzt und dessen dynamische Ressourcenzuteilung durch das Betriebssystem gleichzeitig ein kooperatives Verhalten bezüglich anderer laufender Applikationen sicherstellt².

² Eine detaillierte Beschreibung befindet sich unter folgender Webadresse:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngenlib/html/msdn_manamemo.asp

Konkret bedeutet das, dass der *Index – Services* nur 10 MB zum Betrieb benötigt. Dies ist theoretisch auch für große Dokumentmengen ausreichend - allerdings steigt die Verarbeitungsgeschwindigkeit an, wenn das Betriebssystem große Teile der Datenbank im Hauptspeicher hält. Für 1.000 Dokumente werden je nach Dokumentkollektion ca. 5 - 15 MB Festplattenplatz für die Datenbank benötigt.

Performance

Auf aktueller Hardware (2GHz, 512MB RAM) und mittlerer Kollektionsgröße (ca. 10.000 Dokumente) kann der *Index – Services* ca. 20 Dokumente pro Sekunde indexieren. Suchanfragen werden, abhängig von ihrer Komplexität, in 5-20ms beantwortet.

3.1.2 Mediator

Der *Mediator* stellt einen transparenten Zugriff auf alle *Index – Services* bereit.

Alle Objekte bekommen einen eindeutigen Global Objekt Identifier (GID), welcher die Domäne, den Namen des *Index – Service* sowie die Objekt ID in diesem eindeutig beschreibt. Die GID ist unabhängig von der IP Adresse des *Index – Service* oder der physischen Ablage des bezeichneten Objekts. Die Migration eines *Index – Service* auf einen anderen Rechner oder der Betrieb auf einem Laptop mit wechselnden IP-Adressen ist somit möglich.

Applikationen, die auf dem *Mediator* basieren, sehen nur noch einen großen *Index – Service*, der alle Objekte transparent zugreifbar macht, unabhängig davon, wo auf der Welt der zugehörige *Index – Service* oder die referenzierten Dokumente liegen.

Ein Teil der intelligenten Retrievalverfahren wie Clustering, Relevance Feedback, Konzeptsuche, etc. benötigen die Umrechnung von Konfidenzwerten und Konzepten im *Mediator* bevor Anfragen an ausgewählte *Index – Services* gestellt werden.

Endanwendungen auf Basis des *Mediators* können so entwickelt werden, dass sie Teilergebnisse (z.B. zu einem frühen Zeitpunkt, wenn noch nicht alle *Index – Services* geantwortet haben) verarbeiten, darstellen und so eine sehr gutes Antwortverhalten aufweisen.

Der *Mediator* verfügt über ein intelligentes Caching, welches vermeidet, dass häufig benötigte Objekt-Attribute mehrfach vom *Index – Service* angefordert werden müssen. Über einen Listener-Mechanismus wird gewährleistet, dass Änderungen an einem Objekt unmittelbar allen Interessenten mitgeteilt werden.

Der *Mediator* an sich ist kein Service, sondern eine Bibliothek, die von Anwendungen genutzt werden kann um auf die vom *nextbot Knowledge Framework* bereitgestellten Funktionen zuzugreifen.

Eine genaue Beschreibung existiert in Form einer Entwicklerdokumentation.

3.1.3 XVFS Service

Der *XVFS – Service* (eXtended Virtual File System) synchronisiert die Inhalte zwischen *Repositories* und zugeordneten *Index – Services*.

Der Service definiert eine Schnittstelle auf deren Basis *Konnektoren* zu beliebigen *Repositories* entwickelt werden können.

Die *XVFS – Services* und *Index – Services* können auf verschiedenen Rechnern laufen und kommunizieren über das Netzwerk miteinander.

Es werden verschiedene Synchronisationsstrategien angeboten die entweder Dokumente und Struktur (oder Teile davon) aus einem *Repository* in den *Index – Service* importieren oder diesen adaptiv aktualisieren.

Neue, geänderte oder gelöschte Dokumente werden erkannt und nach einer Konvertierung in ein einheitliches XML-Format an den Index Server zur weiteren Verarbeitung übergeben.

Der Konverter wandelt die wichtigsten Formate wie Office, PDF, HTML, TXT, Emails, etc.. Weitere Konverter können dem System einfach hinzugefügt werden.

3.1.4 Konnektoren

Konnektoren implementieren ein einheitliches Interface, können als plug-in im *XVFS – Service* registriert werden und erlauben diesem direkt auf die *Repositories* zuzugreifen.

Derzeit existieren folgende *Konnektoren*:

- Windows / Unix (Netzwerk) Filesystem
- IMAP / Exchange
- Outlook
- Bookmarks (IE, Mozilla)
- History (IE, Mozilla)
- Lotus Notes
- WebDAV
- nextbot-Repository*
- HTTP (Intranet, Internet)
- XML / RDF

Weitere Konnektoren können auf Basis der Interfacespezifikation einfach entwickelt und dem System hinzugefügt werden.

3.1.5 nextbot-Repository

Das *nextbot - Knowledge Framework* stellt ein eigenes Repository bereit in dem Dokumente versioniert und komprimiert gespeichert werden können.

Basis ist das Versionsmanagementsystem *Subversion*, welches intelligente Verfahren zur Analyse von Änderungen zwischen verschiedenen Versionen eines Dokuments implementiert.

Da Dokumente und die Änderungen zwischen Versionen komprimiert gespeichert werden, benötigt dieses Repository in der Regel nicht mehr Speicherplatz für ein Dokument als ein herkömmliches Filesystem.

3.1.6 Google Index – Service

Der *Google Index – Service* ermöglicht es Suchanfragen an die Internetsuchmaschine Google ® weiterzuleiten und so Suchergebnismengen im Netzwerk um Ergebnisse von Google ® zu erweitern.

Hierzu implementiert der GIS die Schnittstelle eines *Index – Service* mit der Einschränkung, dass nur lesende Zugriffe und keine Modifikationen erlaubt sind.

Analog zu diesem Service können andere Komponenten entwickelt werden, die weitere in Organisationen vorhandene Suchsysteme einbinden.

3.2 Komponenten zur Systemorganisation

Die Architektur des *nextbot - Knowledge Framework* ist so angelegt, dass die Komponenten autark, asynchron und dezentral agieren können. Das System ist vergleichbar mit P2P-Netzwerken, in denen Services direkt und ohne den Umweg über eine zentrale Instanz kommunizieren.

Als Kommunikationsprotokoll wird *PerspectiveBroker* von *twistedmatrix* verwendet. Die Kommunikation ist über SSL v3 (TLS v1/ RSA 2048 BIT) verschlüsselt. Die Authentifizierung erfolgt über signierte Zertifikate.

Nutzer, Service Namen, Domänen und Zertifikate

Jeder Service ist über einen eindeutigen Namen bei einem oder mehreren *Directory – Services* registriert. Servicenamen müssen den Namen des Nutzers der den Service betreibt enthalten (z.B. „maier::index14“). Services können in mehreren verschiedenen Domänen registriert sein. Für jede Domäne benötigt der Betreiber des Service ein eindeutiges Zertifikat, welches ihn ausweist. Nutzer können beliebig viele Services von jedem Typ starten.

Services sind nicht an eine bestimmte IP-Adresse gebunden, sondern können überall in der Welt laufen sofern sie sich beim *Directory – Service* anmelden und ihren neuen Verbindungs-Endpunkt mitteilen.

Im Folgenden werden diejenigen Komponenten beschrieben, die es ermöglichen, dass die Services sich finden, gegenseitig authentifizieren und über Firewallgrenzen hinweg arbeiten können.

3.2.1 Directory – Service

Der *Directory – Service* verwaltet die Information darüber, welche *nextbot- KF* Services im Netzwerk verfügbar sind und über welche IP-Adressen und Ports (Endpunkte) diese angesprochen werden können. Der *Directory – Service* vermerkt zu jedem registrierten Service:

- Servicetyp (z.B. „*Index – Service*“)
- Nutzernamen (z.B. „maier“)
- Domäne (z.B. „brainbot.com“)
- Servicenamen (z.B. „maier::index17“)
- Verbindungs-Endpunkt (z.B. 131.78.93.12:5601)

Um einen *Single Point of Failure* zu vermeiden, können mehrere *Directory – Services* auf unterschiedlichen Rechnern laufen. Die *Directory – Services* synchronisieren sich untereinander und gewährleisten, dass beispielsweise bei einem Hardware Defekt das Netzwerk weiterhin verfügbar ist.

3.2.2 Authentication – Service

Der *Authentication – Service* verwaltet die Benutzer und Gruppenzugehörigkeit jeder *nextbot – KF* Domäne. Diese Informationen können von einem LDAP-Server oder aus dem Active Directory eingelesen und zur Authentifizierung von Benutzern genutzt werden.

3.2.3 Certificate – Service

Um Nutzer und Services eindeutig authentifizieren zu können werden im Netzwerk signierte Zertifikate verwendet. Weiterhin werden sie zur Verschlüsselung der Kommunikation eingesetzt.

Zertifikate besitzen eine Gültigkeitsdauer, die allerdings auch unbeschränkt sein kann. Dadurch ist es beispielsweise möglich, einen Gastzugang zu erzeugen, der nach dem gewünschten Zeitraum seine Gültigkeit verliert. Zertifikate müssen von einer zentralen Stelle auf ihre Gültigkeit geprüft werden - dies geschieht durch den Certificate Service.

Weiterhin erlaubt der Service Nutzern Zertifikate zu beantragen, welche nach Angabe von Namen und Passwort und erfolgreichem Abgleich mit dem *Authentication – Service* generiert und signiert werden.

3.2.4 Proxy Service

Die Kommunikation mehrerer Services ist standardmäßig nicht über Netzwerkgrenzen hinweg möglich, wenn die verschiedenen Netze durch Firewalls voneinander getrennt sind. Um zu ermöglichen, dass ein Service im Netzwerk von Standort A über das Internet einen Service im Netzwerk an Standort B durchsuchen kann (wobei Standort A und B jeweils durch eine Firewall vom Internet getrennt sind) muss auf beiden Firewalls der Proxy Service installiert sein. Der Proxy Service übernimmt das Portforwarding, welches interne Services auch außerhalb der Firewall sichtbar macht. Hierbei synchronisiert er sich mit dem internen *Directory – Service*, der sowohl interne als auch externe Endpunkte zu jedem Service verwalten kann.

Die strenge Authentifizierung über Zertifikate stellt sicher, dass nur autorisierte Nutzer Zugang zu den Services hinter der Firewall erhalten.

3.3 Komponenten zur Integration in Lösungen

3.3.1 Nextbot Web-Services / SOAP-API

SOAP ist ein vom W3C standardisiertes und von vielen Herstellern (insb. Microsoft .NET) unterstütztes Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. SOAP stützt sich auf die Dienste anderer Standards: wie beispielsweise XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht (vgl. TCP/IP-Referenzmodell) zur Übertragung der Nachrichten.

Eine SOAP-Nachricht ist nach dem Head-Body Pattern modelliert. Im Head-Bereich der Nachricht werden die Metainformationen der Nachricht untergebracht. Diese können Informationen über das Routing der Nachricht, über eine eventuelle Verschlüsselung und / oder über die Zugehörigkeit zu einer Transaktion umfassen. Im Body der Nachricht sind die Nutzdaten untergebracht. Diese Daten müssen vom Empfänger der Nachricht interpretiert werden, mögliche Zwischenstationen können diese auch ignorieren.

Die *nextbot SOAP-API* stellt die Funktionalität des *Mediators* als synchronen (blocking) oder asynchronen Service im Netzwerk zur Verfügung.

Eine genaue Beschreibung der API existiert in Form einer Entwicklerdokumentation.

3.3.2 WebDAV Server

WebDAV (WWW Distributed Authoring and Versioning) ist eine Erweiterung des HTTP-Protokolls, und wird von einer Vielzahl von Applikationen, insbesondere CMS-Systemen unterstützt.

DAV Searching and Locating (DASL) ist eine Erweiterung des Standards um Ressourcen über Attribute oder Volltextsuchen zu finden.

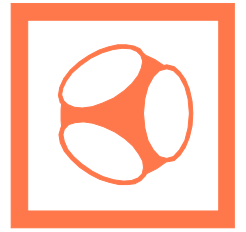
Das *nextbot - KF* bietet bestehenden Applikationen die Möglichkeit über den WebDAV-Standard auf das System zuzugreifen.

3.3.3 Windows Shell Integration / COM-Service

Die Windows Shell Integration bildet die Taxonomien mit den enthaltenen Dokumenten der *Index – Services* im Microsoft Windows Explorer ab. Sie erlaubt einen Zugriff und die Manipulation der Strukturen und Dokumente in den *Index – Services* direkt über den Explorer.

NEXTBOT

KNOWLEDGE FRAMEWORK



brainbot
TECHNOLOGIES AG

Hierzu nutzt sie eine DLL, welche einen COM-Service zur Verfügung stellt. Diese Bibliothek kann eingesetzt werden um einfache Windows Applikationen zum Browsen und Manipulieren der Taxonomien und Dokumente in den *Index – Services* zu entwickeln.

4 Entwicklungsstandards

4.1 Programmiersprachen

Das System ist überwiegend in der dynamische High-Level Programmiersprache Python implementiert. Performancekritische Teile sind in ANSI C++ geschrieben und werden von Python aus eingebunden.

Die Hauptvorteile von Python sind:

Effizienz in der Entwicklung: Programmierer können sich auf die Applikationslogik konzentrieren, statt auf Sprachartefakte und Low-Level-Aspekte. Es gibt kaum Grenzen in der Leistungsfähigkeit der Sprachmittel bei gleichzeitig großer Tendenz zu einfachen Formulierungen.

Plattformunabhängigkeit: Python läuft auf allen bekannten Rechner-Plattformen, auch mit kompatibelem Byte-Code. Reine Python Programme können ohne Modifikation auf andere Plattformen übertragen werden

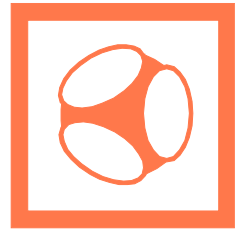
Stabilität & Robustheit: Überragende Fehlerbehandlungsmethoden, besondere Eignung zur Test-Automatisierung & Fehlerprotokollierung, selbst-stabilisierende Fähigkeiten.

Teile der windowsspezifischen Komponenten (vornehmlich die brainFiler-GUI) sind in Borland Delphi entwickelt.

4.2 XP als Vorgehensmodell

Extreme Programming (XP) ist ein relativ neues Grundvorgehen in der Softwaretechnik. Dabei wird auf einen strikten Anforderungskatalog des Kunden verzichtet und es werden auch Kundenwünsche berücksichtigt, die sich während der Entwicklung noch ergeben. Statt des klassischen Wasserfallmodells (bzw. einer Model driven architecture) durchläuft der Entwicklungsprozess immer wieder die Zyklen von Implementierung eines kleinen Schrittes, Tests, und eventuellen Änderungen der Anforderungen (ständig verbesserte Prototypen). Nur die im aktuellen Iterationsschritt benötigten Features werden implementiert.

Dieser Methode liegt die Erfahrung zugrunde, dass der Kunde die wirklichen Anforderungen zum Projektbeginn meist noch nicht komplett kennt. Er fordert Features, die er nicht braucht, und vergisst solche, die benötigt werden.



Es handelt sich um ein Konglomerat aus verschiedenen Ideen, insbesondere

- Pair-Programming (Zwei Programmierer teilen sich eine Tastatur und Monitor - einer codiert, einer denkt mit)
- Integration in kurzen Abständen
- Ständiger Test (alle, auch Kunden testen laufend)
- Laufende Refaktorisierung, ständige Architektur-Verbesserung

4.3 Qualitätsmerkmale nach ISO 9126

Für Software-Produkte werden nach DIN ISO 9126 folgende Qualitätsmerkmale definiert.

- Funktionalität
- Zuverlässigkeit
- Benutzbarkeit
- Effizienz
- Änderbarkeit
- Wartung
- Portierbarkeit

Für alle Software der brainbot Technologies AG ist ein Qualitätsmodell mit Qualitätsmerkmalen definiert aus welchem sich eine Bewertung der Softwarequalität nach Qualitätsindikatoren ableitet.

4.4 Plattformen

Das Framework arbeitet unter Microsoft Betriebssystemen ab Windows 2000 sowie unter den Unix-Derivaten Linux, FreeBSD oder OS X.