Technische Universität
Kaiserslautern

o1o1seda o1o1oo

software engineering dependability

Software Entwicklung 2

Bindung und Kopplung

Inhalt

- Kriterien zur Bewertung von Architekturen
  - Bindung
  - Kopplung

© Prof. Dr. Liggesmeyer

Entwurfskriterien
Generelle Entwurfskriterien

- Architekturen von Softwaresystemen werden – neben ggf. existierenden speziellen Zielsetzungen – anhand der Kriterien *Bindung* und *Kopplung* bewertet.
- Bindung und Kopplung sind methodenunabhängige Kriterien, d.h. sie gelten z.B. für nicht objektorientierte Architekturen genauso wie für objektorientierte Architekturen.

Bindung und Kopplung

- Die **Bindung** (cohesion) innerhalb einer Systemkomponente und die Kopplung der Systemkomponenten untereinander bestimmen die Struktur eines Software-Systems. Bindung (cohesion) ist ein qualitatives Maß für die Kompaktheit einer Systemkomponente. Es werden dazu die Beziehungen zwischen den Elementen einer Systemkomponente betrachtet.
- **Kopplung** (coupling) ist ein qualitatives Maß für die Schnittstellen zwischen den Systemkomponenten. Es werden der Kopplungsmechanismus, die Schnittstellenbreite und die Art der Kommunikation betrachtet.

© Prof. Dr. Liggesmeyer

Bindung und Kopplung

- Die **Bindung** (cohesion) innerhalb einer Systemkomponente und die Kopplung der Systemkomponenten untereinander bestimmen die Struktur eines Software-Systems. Bindung (cohesion) ist ein qualitatives Maß für die Kompaktheit einer Systemkomponente. Es werden dazu die Beziehungen zwischen den Elementen einer Systemkomponente betrachtet.
- Die Beziehungen zwischen den Elementen einer Systemkomponente sollen möglichst ausgeprägt sein. Die Systemkomponenten einer guten Architektur besitzen eine **starke** Bindung.
- Die Bindung (B) ist ein Maß, das sich auf eine einzelne Systemkomponente (M) bezieht, d.h. **B = f(M).**

© Prof. Dr. Liggesmeyer

Bindung und Kopplung

- **Kopplung** (engl. coupling) ist ein Maß für die Stärke der Beziehung zwischen zwei betrachteten Systemkomponenten.
- Es werden der sogenannte Kopplungsmechanismus, die Schnittstellenbreite und die Art der Kommunikation betrachtet.
- Die Kopplung soll möglichst **schwach** sein.
- Die Kopplung (K) bezieht sich auf zwei betrachtete Systemkomponenten (M1, M2), d.h. **K = f(M1, M2)**.

© Prof. Dr. Liggesmeyer

Bindung und Kopplung

- Das Ziel des Entwurfs ist es eine Architektur so zu konzipieren,
  - dass die Bindung der Systemkomponenten möglichst hoch ist und
  - die Kopplung zwischen den Systemkomponenten möglichst gering ist.
- Kopplung und Bindung sind zwei Entwurfskriterien, die einander unterstützen. Eine Stärkung der Bindung führt zwangsläufig zu einer Schwächung der Kopplung. Es ist daher im Grunde nicht erforderlich einen Kompromiss zu erreichen.
- Verletzungen der Bindungs- und Kopplungskriterien können jedoch fachliche Gründe besitzen (z.B. bessere Verständlichkeit der Struktur).
- Eine starke Bindung und schwache Kopplung begünstigt z.B. die Wartbarkeit und Wiederverwendbarkeit.
- Die Überprüfung von Bindung und Kopplung sollte in jedem Fall Bestandteil der Qualitätssicherung einer Architektur sein.

© Prof. Dr. Liggesmeyer

Bindung

- Eine gute Bindung liegt vor, wenn nur solche Elemente zu einer Einheit zusammengefasst werden, die auch zusammen gehören .
- Die folgenden Bindungen werden unterschieden (in aufsteigender Stärke):
  1. zufällige Bindung
  2. logische Bindung
  3. zeitliche Bindung
  4. prozedurale Bindung
  5. kommunikative Bindung
  6. sequentielle Bindung
  7. funktionale Bindung

Entwurfskriterien Bindung

Bindungsarten

- Eine **sequentielle Bindung** liegt vor, wenn mehrere Teilfunktionen, die hintereinander ausgeführt werden, zu einer Systemkomponente zusammengefasst werden. Sequentiell gebundene Systemkomponenten können offensichtlich nicht wiederverwendet werden, wenn eine andere Ausführungssequenz erforderlich ist.
- Eine **kommunikative Bindung** liegt vor, wenn Teilfunktionen aufgrund von Kommunikationsbeziehungen zu einer Systemkomponente zusammengefasst werden.
- Bei einer **temporalen Bindung** ist der gemeinsame Ausführungszeitpunkt entscheidend für die Bildung der Systemkomponente (z.B. Realisierung einer Komponente, die alle Initialisierungen durchführt).
- Einer **zufälligen Bindung** liegt kein erkennbares Prinzip zugrunde.

© Prof. Dr. Liggesmeyer

Bindung

- Ziel: Erreichen einer funktionalen Bindung
  - Alle Elemente sind an der Verwirklichung einer einzigen, abgeschlossenen Funktion beteiligt.
  - Komplexe Funktionen werden realisiert, indem importierte Funktionen verwendet werden, die selbst funktional gebunden sind.
- Kennzeichen einer funktionalen Bindung
  - Alle Elemente tragen dazu bei, ein einzelnes spezifisches Ziel zu erreichen.
  - Es gibt keine überflüssigen Elemente.
  - Die Aufgabe kann mit genau einem Verb und genau einem Objekt beschrieben werden.
  - Austausch gegen anderes Element, welches denselben Zweck erfüllt, leicht möglich
  - Hohe Kontextunabhängigkeit, d.h. einfache Beziehungen zur Umwelt

© Prof. Dr. Liggesmeyer

Bindung

- Vorteile einer funktionalen Bindung
  - Hohe Kontextunabhängigkeit (die Bindungen befinden sich innerhalb der Prozedur, nicht zwischen Prozeduren)
  - Geringe Fehleranfälligkeit bei Änderungen
  - Hoher Grad der Wiederverwendbarkeit
  - Leichte Erweiterbarkeit und Wartbarkeit, da sich Änderungen auf isolierte, kleine Teile beschränken
- Konzept der Bindung verallgemeinert die Regeln für „guten Code“ zu Regeln für „guten Software-Entwurf“
- Die Bindungsart einer Prozedur lässt sich nicht automatisch ermitteln, sondern nur durch manuelle Prüfmethoden

Funktionale Bindung

- Bei einer funktionalen Bindung tragen alle Elemente dazu bei, ein einzelnes spezifisches Ziel zu erreichen. Diese Funktionalität kann daher mit genau einem Verb und genau einem Substantiv beschrieben werden:
  - „Drucke Bericht“,
  - nicht „Drucke und versende Bericht“,
  - ebenfalls nicht „Drucke Bericht und Deckblatt“.
- Sicherheitskritische Funktionen können geeignet gekapselt und vom Rest des Systems getrennt werden => aufwändige Techniken für den Umgang mit sicherheitskritischer Funktionalität müssen nur auf jene Komponenten angewendet werden, die dieses zwingend erfordern. Unkritischere Komponenten können abgetrennt und mit konventionelleren Techniken behandelt werden.

© Prof. Dr. Liggesmeyer

Bindung

- Bindung von Datenabstraktionen/Klassen
  - Beschreibt das Zusammenwirken verschiedener Funktionen, welche derselben Datenabstraktion oder Klasse zuzuordnen sind. Voraussetzung: Alle Methoden sind funktional gebunden
  - Gute Bindung einer Klasse (model cohesion) liegt vor, wenn
    - sie ein einzelnes semantisch bedeutungsvolles Konzept repräsentiert
    - die Klasse keine verborgenen Klassen enthält und
    - keine Operationen enthält, die an andere Klassen delegiert werden können
  - Wird in der Literatur auch als Kohärenz bezeichnet
  - Für Klassen ist weiter die Bindung innerhalb von Vererbungsstrukturen wesentlich.

© Prof. Dr. Liggesmeyer

Informale Bindung

- [Myers 78] fordert für abstrakte Datenobjekte informale Bindung.
  - Diese liegt vor, wenn mehrere, in sich abgeschlossene, funktional gebundene Zugriffsoperatoren, die zu einer Datenabstraktion gehören, auf einer einzigen Datenstruktur operieren.
  - Idee: hinter der gemeinsamen Funktionalität liegt auch ein gemeinsames Datenmodell
- Merkmale
  - Unterstützt das Geheimnisprinzip, d.h. die Datenstruktur gehört nur zu einer Datenabstraktion
  - Änderungen der Datenstruktur tangieren nur eine Datenabstraktion
  - Problem der Vermischung von Zugriffsoperationen, da alle auf derselben Datenstruktur operieren

© Prof. Dr. Liggesmeyer

Informale Bindung

- Kann auch als funktionale Bindung betrachtet werden
- Dies ist dann gegeben, wenn der Zweck der Datenabstraktion die Bereitstellung einer einzelnen, definierten Dienstleistung ist – z.B. „Lese Antriebsparameter“.
- Es kann funktionale Bindungen auf allen Abstraktionsebenen eines Systems geben.
- Funktional gebundene Systemkomponenten auf höheren Abstraktionsebenen sollten funktional gebundene Komponenten enthalten.
- Zusätzlich können Daten hinzutreten, die zur Erledigung der Aufgabe nötig sind.

© Prof. Dr. Liggesmeyer

Entwurfskriterien
Bindung

- Beispiel: Die Komponente „Lese Antriebsparameter“ enthält die funktional gebundenen Komponenten „Lese Temperatur“, „Lese Öldruck“, „Lese Ladestrom“ und die Daten „Antriebsparameter“:
  - Einerseits verhindert es nicht die funktionale Bindung, dass „Lese Antriebsparameter“ mehrere Funktionen und zugehörige Daten enthält, falls „Lese Antriebsparameter“ ansonsten eine abgeschlossene Funktion erbringt.
  - Andererseits ergibt nicht jede Zusammenfassung von funktional gebundenen Komponenten ein funktional gebundenes übergeordnetes System:
    - So wird „Lese Antriebsparameter und stelle diese dar“ nicht funktional gebunden sein, obwohl die beteiligten Komponenten durchaus funktional gebunden sein können.
    - Eine geeignete Lösung könnte die Auftrennung in die zwei funktionalen Bindungen „Lese Antriebsparameter“ und „Stelle Antriebsparameter dar“ sein.

© Prof. Dr. Liggesmeyer

Bindung in Vererbungsstrukturen

- Die ganze Vererbungshierarchie muss untersucht werden
- Starke Vererbungsbindung liegt vor, wenn die Hierarchie eine Generalisierungs-/Spezialisierungshierarchie im Sinne der konzeptuellen Modellierung ist.
- Schwache Vererbungsbindung liegt vor, wenn die Hierarchie nur zum "code sharing" verwendet wird.
- Das Ziel jeder neu definierten Unterklasse muss sein, ein einzelnes semantisches Konzept auszudrücken.

© Prof. Dr. Liggesmeyer

Entwurfskriterien
Kopplung

- Kopplung ist ein Maß für die Stärke der Interaktion zweier betrachteter Komponenten.
- Die Forderung zu einer starken Bindung führt zwangsläufig zu einer schwachen Kopplung.
- Eine schwache Kopplung gewährleistet eine hohe Unabhängigkeit der Systemkomponenten voneinander. Dies begünstigt die Wartbarkeit, den Austausch von Komponenten aber auch alle Qualitätssicherungsaktivitäten, z.B. den Test.
- Die schwächste und daher beste Kopplung ist die so genannte Datenelementkopplung. Diese liegt dann vor, wenn die Schnittstelle zweier Komponenten nur elementare Daten (d.h. keine Datenstrukturen) und keine Steuerinformation in Vorwärtsrichtung enthält.
- Daten werden selbst verarbeitet, während Steuerinformationen in Vorwärtsrichtung die Verarbeitung steuern, d.h. nicht selbst verarbeitet werden:
  - Einer Komponente „Drucke oder zeige Öldruck an“ muss mitgeteilt werden, ob der Öldruck gedruckt oder angezeigt werden soll. Dieser Schnittstellenparameter ist eine Steuerinformation in Vorwärtsrichtung, weil er die Verarbeitung steuert, d.h. in der Komponente die auszuführenden Teile bestimmt. Dieser Steuerparameter ist erforderlich, weil die Komponente nicht funktional gebunden ist. Durch Auftrennung der Komponente in die zwei Komponenten „Drucke Öldruck“ und „Zeige Öldruck an“ entstehen funktionale Bindungen. Jede Komponente kann für sich aufgerufen werden. Der Steuerparameter ist nicht erforderlich. Steuerinformationen in Rückwärtsrichtung ist akzeptabel insofern es sich um Status-Rückmeldungen handelt. So kann die Komponente „Drucke Öldruck“ z.B. eine Statusrückmeldung (ok, Papier, Toner, ausgeschaltet) zurückliefern.

© Prof. Dr. Liggesmeyer

Entwurfskriterien
Kopplung

- Eine **Datenstrukturkopplung** liegt vor, wenn strukturierte Daten übergeben werden. Dies bezieht sich nicht auf vordefiniert strukturierte Datentypen – z.B. „string“. Diese sind erlaubt und werden als elementare Daten betrachtet. Vielmehr führen selbst definierte strukturierte Daten zu einer Datenstrukturkopplung. Es ist erforderlich, dass die Definition der Datenstruktur den interagierenden Komponenten bekannt ist. Falls Änderungen dieser Definition erforderlich sind, so müssen diese konsistent in allen betroffenen Komponenten durchgeführt werden. Die interagierenden Komponenten verlieren daher an Unabhängigkeit. Fehler werden begünstigt. Die Wartung und Weiterentwicklung wird erschwert.
- Eine **Hybridkopplung** liegt vor, wenn zusätzlich zu den zu verarbeitenden Daten Steuerinformationen in Vorwärtsrichtung übergeben werden (siehe Beispiel „Drucke oder zeige Öldruck an“). Hybridkopplungen treten oft als Folge einer schlechten Bindung auf, weil mehrere Funktionen in einer Komponente zusammengefasst wurden, zwischen denen mit Hilfe von Steuerinformation ausgewählt werden muss.

© Prof. Dr. Liggesmeyer

Entwurfskriterien
Kopplung

- Eine **Externe Kopplung** liegt vor, wenn Komponenten über globale Daten miteinander kommunizieren. Jede Änderung der globalen Daten muss konsistent in allen Komponenten berücksichtigt werden. Derartige Strukturen begünstigen Fehler und führen im Extremfall zu kaum noch wartbaren Systemen.
- Eine **Inhaltskopplung** liegt vor, wenn zur korrekten Funktion einer Komponente Details der Realisierung einer anderen Komponente notwendig sind. Durch Modifikationen von Komponenten können Fehlverhalten an weit entfernter Stelle verursacht werden. Auswirkungen von Modifikationen können bei dieser Art der Kopplung systemweit auftreten.

Qualitätssicherung

- Übergeordnete Ziele: Erreichung einer starken Bindung und einer schwachen Kopplung => Überprüfung, ob die Komponenten auf jeder Abstraktionsebene funktional gebunden sind und ob ausschließlich Datenelementkopplungen vorliegen.
- Falls das nicht erfüllt ist, so ist die Notwendigkeit der vorliegenden Abweichungen zu diskutieren. Ein typisches Beispiel ist die Schaffung einer Initialisierungskomponente, die alle Systemfunktionen in definierte Startzustände versetzt. Diese zeitliche Bindung mag der funktionalen Bindung an dieser Stelle aufgrund des Wunsches, alle Initialisierungen an einer Stelle durchzuführen, vorgezogen werden.

© Prof. Dr. Liggesmeyer

Qualitätssicherung

- Die folgenden Aspekte sollten bei der Beurteilung des Entwurfs beachtet werden:
  - Sind die Komponenten funktional gebunden, d.h. kann das Verhalten jeder Komponente durch ein Verb und ein Substantiv beschrieben werden? Gegebenenfalls ist durch Auftrennung und neues Zusammenfassen eine funktionale Bindung erreichbar. Gibt es wichtige Gründe für nicht funktional gebundene Komponenten?
  - Liegen ausschließlich Datenelementkopplungen vor? Hybridkopplungen können oft durch eine Auftrennung der dienstanbietenden Komponente beseitigt werden. Externe Kopplungen können durch Realisierung einer Datenabstraktion beseitigt werden. Inhaltskopplungen erfordern i.allg. umfassende Änderungen.

© Prof. Dr. Liggesmeyer