

# Stücklistenbasiertes Komponenten- Konfigurationsmanagement

Steffen Becker<sup>+</sup>, Sven Overhage<sup>\*</sup>

+ *Georg-Spengler-Straße 20b, D-64291 Darmstadt, E-Mail: stbecker@rbg.informatik.tu-darmstadt.de*

\* *An der Steinmauer 2, D-61191 Rosbach, E-Mail: sven@overhage.org*

**Zusammenfassung.** In diesem Beitrag wird ein Konzept für das Konfigurationsmanagement komponentenorientierter Anwendungen dargestellt. Dabei wird zunächst der Begriff „Konfigurationsmanagement“ näher erläutert und anschließend die Stücklistenorganisation als eine geeignete Methode für das Konfigurationsmanagement beschrieben. Der Beitrag konzentriert sich auf die Entwicklung einer Vorgehensweise zur (automatisierten) Unterstützung der Komponentenauswahl, die auf Stücklisten, einem einheitlichen Spezifikationsrahmen und einer multiattributiven Entscheidungsfindung basiert. Abschließend wird das Änderungsmanagement beschrieben, das ebenfalls zum Konfigurationsmanagement zu zählen ist.

**Schlüsselworte:** Konfigurationsmanagement, Auswahlmanagement, Änderungsmanagement, Stückliste, Variantenstückliste, Spezifikation, Multiattributive Entscheidungsfindung

## 1 Einleitung

Die Einführung der Komponentenorientierung in die betriebliche Anwendungsentwicklung verspricht eine Reihe von Vorteilen, darunter neben einer kürzeren Entwicklungszeit der zu entwickelnden Anwendungen vor allem eine verbesserte Wartbarkeit, Skalierbarkeit und Qualität [Szyp1998:3ff]. Durch den Einsatz weitgehend vorgefertigter Komponenten reduziert sich zunächst vor allem der Aufwand während der Implementierung, die nunmehr lediglich die Neuentwicklung einiger weniger Komponenten sowie den Zusammenbau (die Konfiguration) von Komponenten zu Anwendungen umfasst. Darüber hinaus konzentriert sich die Wartung bzw. Skalierung einer komponentenorientierten Anwendung nur noch auf die jeweils von Änderungen betroffenen wenigen Komponenten, die noch dazu in der Regel von fachkundigen Herstellern realisiert wurden und durch zahlreiche Wiederverwendungen auch bereits entsprechend ausgereift sein dürften.

Trotz dieser vielen Vorteile konnte sich die seit langem in der Literatur empfohlene komponentenorientierte Anwendungsentwicklung [Mcil1968] in der Praxis bislang jedoch nicht auf breiter Front durchsetzen. Als Gründe für diese Entwicklung werden häufig fehlende fachliche und technische Standards für die Entwicklung und Spezifikation von Komponenten genannt, wodurch das Auftreten von Inkompatibilitäten bzw. Heterogenitäten [Over2002b] zwischen diesen begünstigt wird. Des Weiteren werden die bislang mangelhaft entwickelten Komponentenmärkte dafür verantwortlich gemacht, die jedoch eine weitere wichtige Voraussetzung für den Erwerb von Komponenten zur Anwendungsentwicklung darstellen [Hahn2002]. Darüber hinaus ist festzuhalten, dass die Einführung der Komponentenorientie-

rung in die Anwendungsentwicklung nicht automatisch zu besser strukturierten und beherrschbaren Anwendungen führt. So bleibt einerseits die Zerlegung einer zu entwickelnden Anwendung in geeignete Komponenten dem Geschick der jeweiligen Beteiligten überlassen. Andererseits führt die Zerlegung einer Anwendung in eine Vielzahl von Komponenten auch zu einer höheren Komplexität in der Anwendungsentwicklung, die zu beherrschen und durch entsprechende Methoden zu unterstützen ist. Als Beispiele hierfür sind vor allem Methoden für das Auswahlproblem im Hinblick auf einen geeigneten Komponenten-Mix bei der Entwicklung einer Anwendung sowie Methoden für die effiziente Ermittlung der in Anwendungen verbauten Komponenten zur Unterstützung von Wartungsarbeiten bzw. Skalierungsarbeiten an bestehenden Systemen zu nennen.

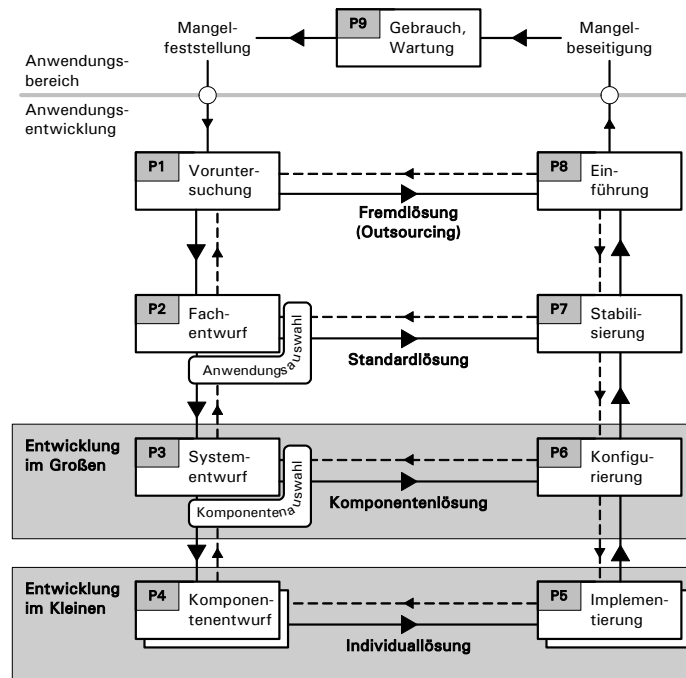
Hieraus lässt sich entnehmen, dass mit der komponentenorientierten Anwendungsentwicklung auch ein effizientes Komponenten-Konfigurationsmanagement entwickelt werden muss, das den Entwurf sowie die Verwaltung von Konfigurationen (also Anwendungen) aus Komponenten unterstützt. Im Rahmen dieses Beitrags wird als Grundlage für das Komponenten-Konfigurationsmanagement in der Anwendungsentwicklung die Stücklistenorganisation verwendet, die in der betrieblichen Produktionswirtschaft in ähnlicher Funktion bereits seit langem eine zentrale Rolle spielt [OrLK1999], [Ortn2001]. Im folgenden Kapitel wird zunächst der Begriff „Komponenten-Konfigurationsmanagement“ näher definiert und in die Anwendungsentwicklung eingeordnet. Daran anschließend werden verschiedene Arten von Stücklisten vorgestellt und deren Eignung für das Komponenten-Konfigurationsmanagement untersucht. Im vierten Kapitel wird auf Basis einer speziellen Stücklistenform ein möglicher Algorithmus für die Unterstützung bei der Auswahl von Komponenten vorgestellt, der die einzelnen Ebenen des standardisierten Komponentenspezifikationsrahmens des GI-Arbeitskreises Komponentenorientierte betriebliche Anwendungssysteme (im Folgenden als GI-Arbeitskreis bezeichnet) [ABC+2002] als Parameter für die Entscheidungsfindung einbezieht. Abschließend werden einige mögliche Einsatzgebiete für Komponenten-Stücklisten im Rahmen der Wartungsarbeiten an bestehenden Anwendungen skizziert und weitere Forschungsaufgaben in diesem Themengebiet diskutiert.

## **2 Komponenten-Konfigurationsmanagement**

Für die Wartung und Weiterentwicklung komponentenorientierter Anwendungen (bzw. Baugruppen) benötigen die Entwickler eine angemessene Dokumentation der Innensicht. Zwar lässt sich nach dem Komponentenmodell von Shaw und Garlan [ShGa1996:196ff], [OvTh2002] und der Komponentendefinition des GI-Arbeitskreises [ABC+2002] eine Konfiguration (also eine Baugruppe bzw. eine Anwendung) wiederum als komplexe Komponente nach dem Black-Box-Prinzip (ggf. angereichert um eine Spezifikation) in die weitere Anwendungsentwicklung einbeziehen. Insbesondere für den Fall, dass in der bestehenden Anwendung bzw. Baugruppe einzelne Komponenten (im Falle eines Versions- oder Herstellerwechsels) auszutauschen sind, reicht jedoch diese Sichtweise für den Hersteller nicht aus.

Die Anforderungen, die sich im Hinblick auf die Dokumentation komponentenorientierter Anwendungen und Baugruppen aus Herstellersicht ergeben, werden vom Komponenten-Konfigurationsmanagement [Szyp1998:334], [LaCr2000] aufgegriffen, das hierfür geeignete Methoden und Werkzeuge bereitzustellen versucht. Es unterstützt schwerpunktmäßig den effizienten Zusammenbau, die Spezifikation der Innensicht sowie die strukturierte Verwaltung von Anwendungen, die auf der Basis einzelner Komponenten entwickelt werden.

In den Vorgehensmodellen zur Entwicklung komponentenorientierter Anwendungen spielt es somit vor allem in den Phasen „Systementwurf/Komponentenauswahl“, „Konfigurierung“ und „Betrieb/Wartung“ eine wichtige Rolle und ist der Entwicklung im Großen, also dem Bau von Anwendungen und Baugruppen aus fertigen Komponenten, zuzuordnen (vgl. Abbildung 1)<sup>1</sup>.



**Abbildung 1:** Flexibles Multipfad-Vorgehensmodell der komponentenorientierten Anwendungsentwicklung (basierend auf [Ortn1998], [Over2002a]).

Das Komponenten-Konfigurationsmanagement besteht im Wesentlichen aus zwei Anwendungsgebieten, die wie folgt voneinander abzugrenzen sind:

- **Auswahlmanagement:** Bei ausgereiften Komponentenmärkten (vgl. [Hahn2002]) ist allgemein davon auszugehen, dass eine Vielzahl von Bauteilen in einem oder mehreren Teilelagern zur Verfügung stehen. Die Herausforderung der komponentenorientierten Anwendungsentwicklung besteht nun zunächst darin, aus dieser Vielzahl von Teilen durch systematische Auswahl von Komponenten eine Anwendung zu konfigurieren, die die Anforderungen aus der Voruntersuchung bzw. dem Fachentwurf möglichst optimal erfüllt. Dabei ist es ferner wichtig, schon bei der Auswahl eine möglichst gute Abschätzung der einzukaufenden Komponenten und somit der zu erwartenden Einkaufskosten zu liefern, um ggf. andere (günstigere) Komponenten zu suchen und in den Entwicklungsprozess einzubeziehen.
- **Änderungsmanagement/Versionsmanagement:** Wartungsarbeiten an bestehenden komponentenorientierten Anwendungen werden häufig durch die Einführung neuer

<sup>1</sup> Im Gegensatz hierzu befasst sich die Entwicklung im Kleinen mit der Entwicklung einzelner Komponenten. Sie konzentriert sich also auf die Phasen „Komponententwurf“ und „Implementierung“ (vgl. Abbildung 1).

Versionen von Komponenten bzw. den Ersatz bereits verwendeter Komponenten verursacht. Mögliche Herausforderungen bei einem solchen Wartungsprozess bestehen einerseits in der effizienten Ermittlung der von Änderungen betroffenen Anwendungen sowie andererseits dem Einbinden der neuen (gegebenenfalls inkompatiblen) Komponente in die bestehenden Anwendungen. Dabei ist zunächst zu ermitteln, in welchen Anwendungen eine auszutauschende Komponente verwendet wird und ob durch die einzubauende Komponente Kompatibilitätsprobleme auftreten werden.

Beide Gebiete sind auf eine effiziente Methode zur Erfassung der mengenmäßigen Verwendung von Komponenten in Anwendungen bzw. Baugruppen angewiesen. Dazu können als Grundlage einerseits die während des Systementwurfs erstellten Baupläne (Systemdiagramme) dienen<sup>2</sup>. Diese haben jedoch den Nachteil, dass sie die mengenmäßige Verwendung nicht explizit ausweisen und man sie daher zunächst jedes Mal umständlich auswerten muss.

In der betrieblichen Produktionswirtschaft werden aus diesem Grund statt den vorhandenen Konstruktionsplänen der zu produzierenden Erzeugnisse vor allem Stücklisten verwendet, die sich durch ihre Spezialisierung auf die mengenmäßige Struktur auszeichnen. Hierdurch ergeben sich einige Vorteile: So ist es durch die Stücklistenauflösung einerseits effizient möglich, sowohl die verschiedenen Arten der Teile zu identifizieren, die jeweils als Bauteile in ein Erzeugnis eingehen, als auch die jeweilige Menge der benötigten Teile zu berechnen (und auf dieser Basis die Einkaufskosten zu berechnen). Andererseits können Stücklisten auch leicht so erweitert werden, dass sie einen sog. Verwendungsnachweis führen. Dieser erfasst, in welchen Erzeugnissen (genauer gesagt Erzeugnisarten) ein Bauteil verwendet wurde und ermöglicht so das Aufspüren der betroffenen Erzeugnisse im Falle der Änderung eines Bauteils.

Auf Grund ihrer Eigenschaften eignen sich Stücklisten auch für die in diesem Beitrag genannten Aufgaben des Komponenten-Konfigurationsmanagements und werden im Folgenden als dessen methodische Grundlage verwendet (zur prinzipiellen Eignung von Stücklisten in der komponentenorientierten Anwendungsentwicklung vergleiche insbesondere auch [OrLK1999], [Ortn2001]). Da Stücklisten bereits seit längerer Zeit für ähnliche Aufgabenstellungen eingesetzt werden, ist die Wahrscheinlichkeit gegeben, somit eine ausgereifte Methode wieder zu verwenden und ggf. auch einige an diese anknüpfende Methoden für das Komponenten-Konfigurationsmanagement anpassen zu können.

### **3 Stücklisten für das Konfigurationsmanagement**

Allgemein betrachtet beschreiben Stücklisten die Zusammensetzung von Erzeugnissen aus Teilen und Baugruppen. Im Rahmen der komponentenorientierten Anwendungsentwicklung lassen sich dabei als mögliche Bauteile zur Entwicklung einer Konfiguration die einzelnen Komponenten betrachten. Solche Konfigurationen, die sich aus mehreren Komponenten zusammensetzen, können mit Baugruppen gleichgesetzt werden. Da Baugruppen wiederum als (komplexe) Komponenten angesehen werden können und als solche in die Anwendungsentwicklung eingehen [ShGa1996:196ff], bilden sie eine weitere Klasse von Teilen. Eine fertige Anwendung (also ein Endprodukt) wird dabei lediglich als eine besondere Art der Baugruppe angesehen.

---

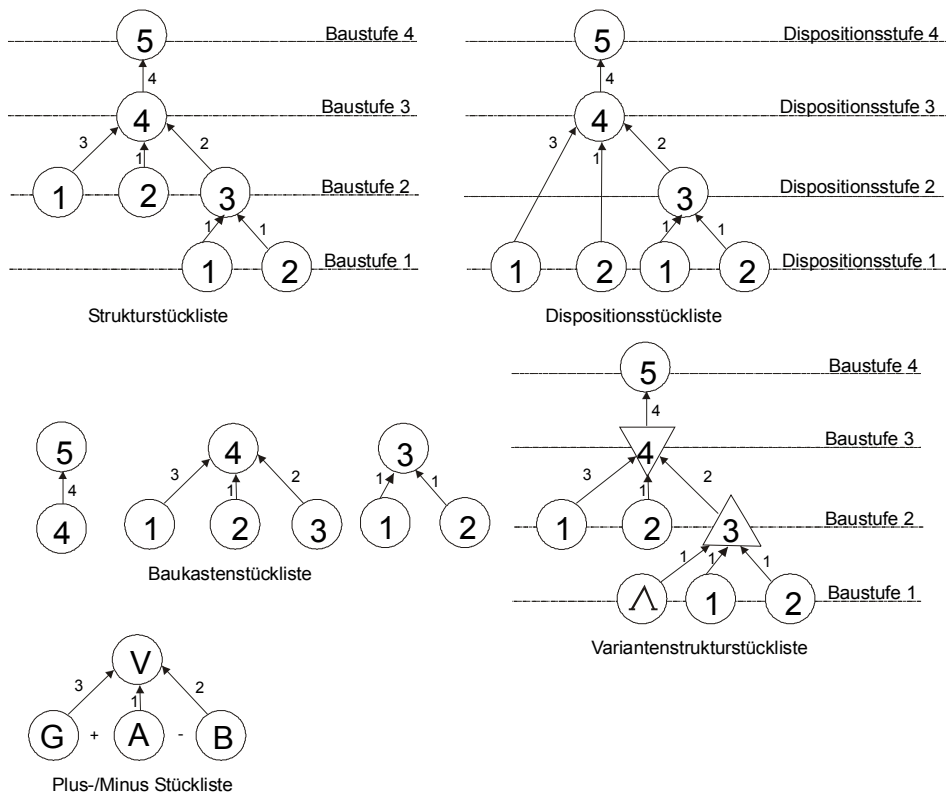
<sup>2</sup> So finden sich beispielsweise in der UML auch speziell auf die Modellierung komponentenorientierter Anwendungen spezialisierte Diagrammsprachen [HoNo2001].

In der allgemeinen Betriebswirtschaftslehre bzw. Produktionswirtschaft werden verschiedene Arten von Stücklisten mit unterschiedlichen Vorzügen unterschieden, insbesondere Mengestücklisten, Strukturstücklisten, Dispositionsstücklisten, Baukastenstücklisten und Variantenstücklisten [Schn1999:201ff]. Im Folgenden werden diese jeweils kurz charakterisiert, um im Anschluss daran aufzuzeigen, welche Form der Stücklistenorganisation sich für einen Einsatz im Konfigurationsmanagement am Besten eignet (vgl. Abbildung 2):

- **Übersichts- bzw. Mengestücklisten** geben ausschließlich die aufsummierten (aggregierten) Mengen der jeweiligen Teile wieder, die in das Endprodukt eingehen.
- **Strukturstücklisten** erfassen weiterhin nach Baustufen geordnet, welche Teile bzw. Baugruppen in welchen Mengen zu neuen Baugruppen bzw. dem Endprodukt zusammengesetzt werden. Sie können durch Gozinto-Graphen veranschaulicht werden.
- **Dispositionsstücklisten** bauen auf Strukturstücklisten auf und untergliedern sie nach der Tiefe der Dispositionsstufe (des Gozinto-Graphen). Es werden dabei jeweils alle Teile einer bestimmten Art auf dieselbe Ebene des Graphen gebracht, so dass alle Teile eines Typs während desselben Produktionsschritts disponiert werden können. Sollten also gleiche Teile auf mehreren Dispositionsstufen vorkommen, werden sie einheitlich in die unterst mögliche Dispositionsstufe eingeordnet.
- **Baukastenstücklisten** blenden die Erzeugnisstruktur der einzelnen Bauteile, die in andere Teile eingehen, aus. Sie enthalten somit jeweils nur die Teile und Baugruppen, die direkt in ein Erzeugnis eingehen. Dies entspricht im Prinzip der bei der komponentenorientierten Anwendungsentwicklung üblichen Vorgehensweise, eine Baugruppe (Konfiguration) als neue Black-Box-Komponente anzusehen, deren Aufbau nicht notwendigerweise bekannt ist. Somit zerfällt der Gozinto-Graph, der den Erzeugniszusammenhang darstellt, in mehrere Graphen mit jeweils zwei Stufen.
- **Variantenstücklisten** können für das Konfigurationsmanagement nützlich sein, wenn verschiedene Produktvarianten (die normalerweise in getrennten Stücklisten erfasst werden müssten) gemeinsam in einer Stückliste vereinigt werden. Dieses Konzept ist auf die drei zuvor genannten Stücklisten anwendbar, die somit jeweils zu einer Variantenstückliste erweitert werden können. Um eine effiziente Darstellung für die Darstellung von Varianten zu erhalten, wurde eine Erweiterung der Knotenarten klassischer Stücklisten vorgeschlagen [WeMü1981]. Die verschiedenen Knotentypen sowie ihre Bedeutungen sind Tabelle 1 zu entnehmen. Abbildung 2 zeigt beispielhaft eine Variantenstrukturstückliste. Mit ihnen können beispielsweise Buchhaltungsanwendungen, die als Bilanzierungskomponente eine US-GAAP-Variante (US Generally Accepted Accounting Principles) bzw. eine IAS-Variante (International Accounting Standard) verwenden, gemeinsam erfasst werden. Über die Unterscheidung verschiedener Bilanzierungsarten hinaus wäre es auch denkbar, zu jeweils einer Bilanzierungsart verschiedene Komponenten zu unterscheiden, die diese implementieren – diese würde man ebenfalls als Variante bezeichnen und in der gemeinsamen Variantenstückliste erfassen. In der Literatur finden sich weitere Möglichkeiten Variantenstücklisten darzustellen. Plus-/Minus-Stücklisten [Zimm1988:113ff] bilden beispielsweise die Differenz zwischen einer Grundvariante und den restlichen Varianten dadurch ab, dass Stücklistenpositionen durch Subtraktion von der Grundliste entfernt werden können und durch Addition neue Positionen hinzukommen. Die darüber hinaus bekannten fiktiven bzw. Gleichteilestücklisten bilden dabei eine Sonderform von

Plus-/Minus Stücklisten, in denen nur additive Positionen auftreten, d.h. die Grundvariante wird hierbei immer um geeignete Teile erweitert.

Für den Einsatz im hier betrachteten Komponenten-Konfigurationsmanagement kommen die vergleichsweise einfach aufgebauten Mengen- bzw. Baukastenstücklisten nicht in Betracht, da die Strukturen der zusammengesetzten Teile durch die jeweilige Anordnung in diesen speziellen Stücklisten verloren gehen und somit nicht mehr direkt zu entnehmen sind. Falls nötig, können die komplexeren Formen von Stücklisten jedoch problemlos (d.h. effizient) in diese beiden einfacheren Arten transformiert werden, so dass durch die Nichtbetrachtung kein Nachteil entsteht.



**Abbildung 2:** Gegenüberstellung verschiedener Stücklistenarten

Strukturstücklisten und Dispositionsstücklisten sind hingegen für eine Verwendung prinzipiell gleichermaßen geeignet. Dispositionsstücklisten können von Vorteil sein, wenn die Teile einer Dispositionsstufe zuerst entwickelt und getestet werden bzw. genauer evaluiert werden müssen. Sie liefern bedingt durch ihre spezielle Struktur die zeitliche Reihenfolge für die Ausführung solcher Tätigkeiten, die jeweils der Reihenfolge der Dispositionsstufen entspricht. Vereinfachend werden im Folgenden jedoch die einfacher strukturierten Strukturstücklisten verwendet, die zur Verdeutlichung des Einsatzgebiets bereits ausreichend sind.

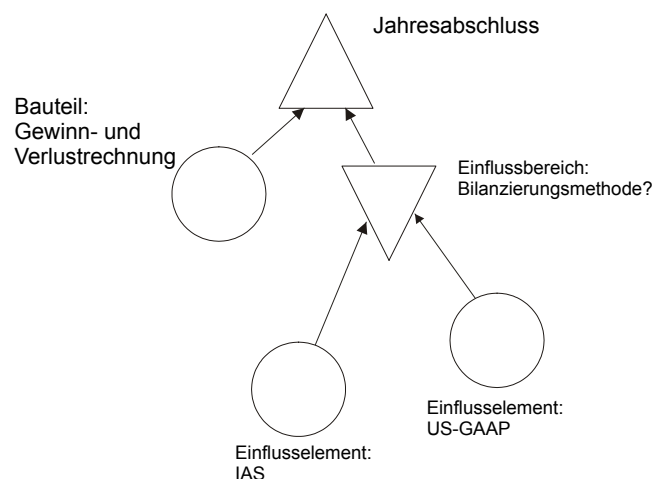
Diese werden für das Komponenten-Konfigurationsmanagement als Variantenstrukturstücklisten verwendet. Dies ermöglicht die Dokumentation von Anwendungen, die sich nur an einigen Stellen durch den Einsatz anderer Komponenten (Varianten) unterscheiden. Dabei können parametrisierte Komponenten als diskrete bzw. stetige Varianten aufgefasst werden [Zimm1988:6ff]. In dieser Arbeit wird dabei die Darstellung gem. Tabelle 1

verwendet. Für eine effiziente Speicherung der Stückliste bietet sich bei der Anwendungsentwicklung Plus-/Minus Stücklisten oder Gleichteilestücklisten eher an, da die zu erwartenden Variationen gegenüber einer Grundvariante gering sein dürften und damit die Differenzinformationen ebenfalls gering ausfallen. Dabei ist eine Umwandlung dieser Stücklisten in die hier verwendete Darstellungsform problemlos möglich.

Knotentyp	Bedeutung	Symbol
Teileknoten	Repräsentiert ein Teil. Mögliche Teile sind Komponenten oder zu einem neuen Bauteil zusammengefasste Komponenten. Die Mengenangabe an der Kante gibt an, wie oft das Teil in die übergeordnete Struktur eingeht.	○
Konjunktivknoten	Fasst alle Teile an eingehenden Kanten zu einem neuen Bauteil zusammen (entspricht einem logischen Und).	△
Alternativknoten	Genau eine der eingehenden Kanten wird ausgewählt (entspricht einem logischen exklusiven Oder). Das gewählte Teil rückt de facto an die Stelle des Alternativknotens.	▽
Leerer Knoten	Wird benötigt, um im Zusammenhang mit Alternativknoten eine optionale Auswahl zu modellieren.	⊕

**Tabelle 1:** Knotentypen zur Darstellung einer Variantenstückliste

Abbildung 3 zeigt eine Baugruppe, die Dienste zur Erstellung eines Jahresabschlusses im Rahmen der Finanzbuchführung anbietet. Hierbei gibt es mehrere verschiedene Baugruppen, die einen Jahresabschluss nach unterschiedlichen Bilanzrichtlinien ermöglichen (beispielsweise IAS und US-GAAP). Die unterschiedliche Funktionalität wird bedingt durch die Verwendung einer jeweils angepassten Komponente „Bilanzierung“, also den Einsatz von Varianten – eine Variantenstückliste vermag diese Anwendungen zusammengefasst darzustellen.



**Abbildung 3:** Variantenstückliste für einen Jahresabschluss

Kann die Menge der Erzeugnisvarianten bei einer Variantenstückliste explizit, z.B. in Form eines Kataloges, angegeben werden, ergibt sich eine geschlossene Variantenstückliste. Ist diese Menge hingegen nur implizit durch die möglichen Entscheidungen innerhalb der Stückliste definiert, so handelt es sich um eine offene Stückliste. Aufgrund der zu erwartenden ho-

hen Variantenanzahl im betrachteten Anwendungsgebiet werden in diesem Beitrag offene Variantenstücklisten betrachtet.

Zusammengefasst lässt sich festhalten, dass im Rahmen des Komponenten-Konfigurationsmanagements der Einsatz (offener) Strukturvariantenstücklisten als methodische Grundlage zu empfehlen ist. Ein erstes mögliches Einsatzgebiet für solche Stücklisten findet sich im Rahmen der Kalkulation der Einkaufskosten bei der Komponentenauswahl. Alle genannten Stücklisten lassen sich durch die Möglichkeit der Stücklistenauflösung für die Kalkulation dieser Einkaufskosten heranziehen. Bei Variantenstücklisten ist dabei zunächst zu unterscheiden, ob die Entscheidungen für eine Variante jeweils bereits getroffen wurden oder nicht. Falls die Entscheidung nicht getroffen wurde, kann die Kostenkalkulation entweder für jede erlaubte Variantenkombination separat vorgenommen werden (wobei sich dann eine Vielzahl von möglichen Kosten ergibt) oder für jeden Variantenknoten ein Kostenminimum und ein Kostenmaximum bestimmt werden. Durch Aggregation entlang der Stückliste erhält man auf diese Weise eine untere und obere Schranke der anfallenden Kosten. Ist die Entscheidung für eine Variante bereits bei allen Variantenknoten gefallen, so wird aus der Variantenstückliste eine „klassische“ Stückliste, die mit den entsprechenden Verfahren aufgelöst werden kann.

Bei einer „klassischen“ Stücklistenauflösung werden zunächst die verschiedenen Teilearten, die in das Erzeugnis eingehen, und deren jeweilige Mengen ermittelt. Anhand der Summe der mit ihren jeweiligen Einkaufskosten multiplizierten Teile kann daran anschließend eine Abschätzung der gesamten Einkaufskosten vorgenommen werden. Besonderheiten ergeben sich hierbei eventuell durch nicht konstante Einkaufskosten pro Teil. Übertragen auf die komponentenorientierte Anwendungsentwicklung bedeutet dies beispielsweise, dass unterschiedliche Lizenzmodelle mit Auswirkung auf den Einkaufspreis bestehen können. So ist denkbar, dass jede in der Anwendungsentwicklung verwendete Komponente mit der gleichen Lizenzgebühr erworben werden muss oder durch eine einmalig vergütete Lizenzgebühr beliebig viele Komponenten dieser Art verwendet werden dürfen. In letzter Zeit etablieren sich, bedingt durch neue Vertriebsarten wie ASP (Application Service Providing) oder XML Web-Services, auch Lizenzmodelle, die eine Gebühr erst bei der Benutzung der Komponente durch einen Anwender zur Laufzeit vereinbaren. In diesem Fall ist die Verwendung im Rahmen der Entwicklung sogar (bezogen auf die Einkaufskosten) kostenfrei. Daher geht in die Preiskalkulation stets noch eine weitere Funktion ein, die die verwendeten Teile im Hinblick auf die jeweils vereinbarten Lizenzbedingungen gewichtet.

#### **4 Stücklistenbasiertes Auswahlmanagement**

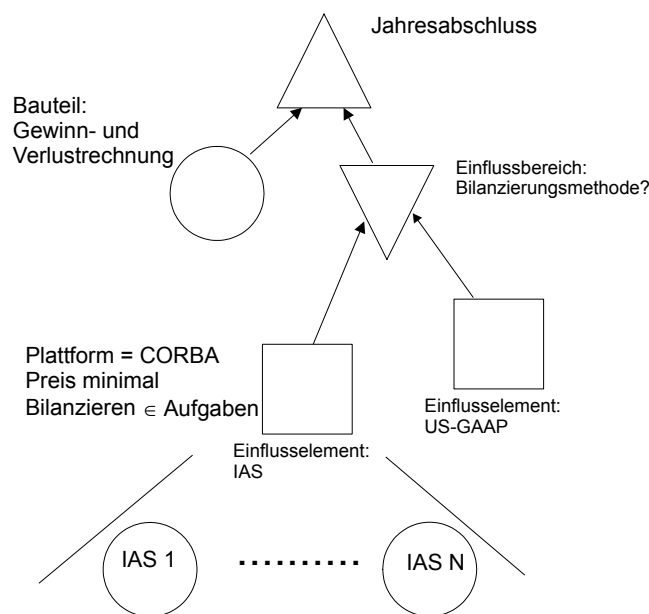
Der Einsatz von Variantenstücklisten unterstützt über die Kostenkalkulation hinaus insbesondere auch die Phase der Komponentenauswahl während der komponentenorientierten Anwendungsentwicklung. Angenommen, es existiert für die zu erstellende Anwendung bereits eine Variantenstückliste, die während des Anwendungsentwurfs (z.B. auf Basis des Fachentwurfs) bereits hergeleitet wurde. Im Rahmen des Anwendungsentwurfs fällt dann zunächst für jeden Variantenknoten der Stückliste die Entscheidung für eine bestimmte Variante. Hierdurch wird der exakte Typ der zu erstellenden Anwendung festgelegt und die Variantenstückliste somit zunächst zu einer „klassischen“ Stückliste ohne Alternativknoten. Bezogen auf das in Kapitel 3 eingeführte Beispiel der Baugruppe „Jahresabschluss“ bedeutet dies, dass sich die Entwickler nunmehr auf die Erstellung einer Baugruppe festlegen, die nach den Vorschriften des IAS bilanziert.



Im Anschluss an diese Entscheidung stellt sich im Rahmen der Komponentenauswahl nun das Problem der Beschaffung der einzelnen in der Stückliste auftretenden Teile. Zu Beginn der Komponentenauswahl ist die Menge an existierenden Komponenten, die anstelle eines in der Stückliste verzeichneten Teils verwendet werden können, noch unbekannt. Um diesen Status klar hervorzuheben, wird die Einführung einer sog. Entscheidungsstückliste vorgeschlagen.

Dabei wird ein neuer Knotentyp eingeführt, der mit einem Quadrat dargestellt werden soll und angibt, dass für den entsprechenden Knoten noch eine Entscheidung für ein konkretes Teil zu treffen ist. In einem Entscheidungsknoten werden Informationen über das benötigte Teil vermerkt, die direkt aus den Anforderungen des Anwendungsentwurfs abgeleitet wurden. Dies geschieht durch die Spezifikation einer fiktiven Referenzkomponente, die auf der Basis eines einheitlichen Spezifikationsrahmens erstellt wird. Es handelt sich hierbei um ein Muster, dem die Spezifikationen der Komponenten möglichst entsprechen müssen.

Bezug nehmend auf das in Kapitel 3 gegebene Beispiel bedeutet dies zunächst, dass die in Abbildung 3 gezeigte Variantenstückliste zu einer Entscheidungsstückliste wird, die in Abbildung 4 dargestellt ist. Der vorhandene Variantenknoten „Bilanzierungsmethode“ ist durch das zusätzliche Erfassen von „IAS“ eliminiert worden und existiert nur noch scheinbar. Aus dem Bauteil „IAS“ ist nunmehr ein Entscheidungsknoten geworden, der für eine (derzeit unbekannte) Menge von (unterschiedlich geeigneten) Komponenten steht, die als Kandidaten für die Anwendungsentwicklung in Frage kommen.



**Abbildung 4:** Entscheidungsstückliste für die Komponentenauswahl

Da es von zentraler Bedeutung für den Auswahlprozess ist, dass sowohl die Komponenten als auch die Referenzspezifikation einheitlich sind, wird als Spezifikationsrahmen der GI-Spezifikationsrahmen [ABC+2002] empfohlen. Dabei ist es jedoch wichtig, den Spezifikationsrahmen als Ganzes in ein maschinenlesbares Format zu bringen, wobei sich hier die Entwicklung einer XML basierten Notation anbietet [OvTh2002], damit die Daten mit möglichst geringem Zusatzaufwand von den entsprechenden CASE-Tools ausgetauscht, eingelesen und

ausgewertet werden können. Auf der Basis einer solchen Notation können entscheidungsunterstützende Systeme arbeiten, die die Auswahl von Komponenten vereinfachen sollen. Die Referenzspezifikation, die in den Entscheidungsknoten beinhaltet ist, muss darüber hinaus nicht vollständig im Sinne des Spezifikationsrahmens sein. In ihr sind lediglich diejenigen Anforderungen zu speichern, die die zur Auswahl stehenden Alternativen (möglichst gut) erfüllen sollen. Wird ein Datensatz oder gar eine ganze Ebene der Spezifikation ausgelassen, so bedeutet dies, dass diese Informationen nicht relevant für die zu treffende Entscheidung sind. Formal ausgedrückt heißt dies, dass die möglichen Alternativen in den ausgelassenen Eigenschaften ihrer Spezifikation nicht gebunden sind.

Die Referenzspezifikation enthält unterschiedliche Angaben. Darunter sind Satisfizierungsziele und Extremalziele zu unterscheiden. Satisfizierungsziele sind Referenzspezifikationen, bei denen es nicht wichtig ist, wie gut sie erfüllt werden, solange sie in irgendeiner Form erfüllt werden. Sie stellen also etwas salopp formuliert K.O.-Kriterien für den Auswahlprozess dar. Bei Extremalzielen wird hingegen gefordert, dass die Referenzspezifikation möglichst gut erreicht wird. In der Entscheidungstheorie werden Referenzspezifikationen der letztgenannten Art über zu optimierende Zielfunktionen abgebildet.

Im Beispiel aus Abbildung 4 fordert eine Referenzspezifikation zum Beispiel als Satisfizierungsziel ein bestimmtes Komponentenframework, die entsprechende Formulierung lautet entsprechend Plattform='CORBA'. Hierbei wird ein Attribut der Vermarktungsebene des Spezifikationsrahmens verwendet, um eine Anforderung zu formulieren. Die zur Auswahl stehenden Komponenten sollen darüber hinaus die geforderte fachliche Aufgabe durchführen können, d.h. es ist zu verlangen, dass „Bilanzieren“ in der Menge der von den Komponenten angebotenen Dienste enthalten sein muss. Auch dies ist ein Satisfizierungsziel, das sich diesmal jedoch auf die Aufgabenebene der Spezifikation bezieht. Da das betrachtete Unternehmen keinen großen finanziellen Spielraum besitzt, fordert das Management, dass die einzukaufende Komponente möglichst billig sein soll, d.h. es wird verlangt, dass der Preis, ein weiteres Attribut der Vermarktungsebene, minimal wird. Hierbei handelt es sich offensichtlich um ein Extremalziel. Dieses Beispiel ließe sich beliebig auf die verbleibenden Ebenen des Spezifikationsrahmens ausdehnen.

Im Folgenden soll der Prozess der Komponentenauswahl auf Basis einer Entscheidungsstückliste genauer beschrieben werden. An dessen Ende sind schließlich alle Teile ausgewählt (und alle Variantenentscheidungen getroffen). Das bedeutet, dass als Resultat wieder eine „klassische“ Stückliste ohne Entscheidungs- bzw. Variantenknoten entsteht. Um dieses Ziel zu erreichen, wird (basierend auf [Kont1995], [KoHe1996]) eine methodische Vorgehensweise in insgesamt sieben Schritten empfohlen. Diese kann zunächst teilautomatisiert werden und durch entsprechende Systeme unterstützt werden. Langfristig ist die vollständig automatisierte Entscheidung eine anzustrebende Perspektive, die zum Beispiel in CASE-Tools für die Anwendungsentwicklung oder in Anwendungen selbst integriert werden könnte (sog. self assembling applications [OvTh2002]).

#### **4.1 Schritt 1: Eliminierung der Variantenknoten**

Zunächst werden die Variantenknoten wie bereits zu Beginn des Kapitels geschildert durch die Entscheidung für einen konkreten Anwendungstyp aus der Stückliste entfernt. Dies geschieht dadurch, dass von der höchsten Ebene ausgehend für jeden Variantenknoten genau eine der möglichen Alternativen ausgewählt wird. Um diese Entscheidung zu unterstützen werden detaillierte Informationen benötigt. Daher wurde in [WeMü1981] vorgeschlagen, den

Variantenknoten sowie den möglichen Alternativen weitere Attribute zuzuordnen, die die Entscheidungsfindung unterstützen sollen. Ein Alternativknoten erhält zunächst eine Information darüber, welche Entscheidung zu treffen ist. Diese Information wird Einflussbereich (der Entscheidung) genannt. Zum Beispiel ist der Einflussbereich bei der Auswahl der richtigen Bilanzierungskomponente aus Kapitel 3 die Art der Bilanzierung. Die möglichen Alternativen bekommen passend zu diesem Einflussbereich ein Attribut, das beschreibt, in welcher Weise der Einflussbereich von der jeweiligen Alternative bedient wird. Diese Information wird Einflusselement genannt.

Ein Auswahlprozess kann mittels dieser Attribute beispielsweise in Form eines dialoggesteuerten Prozesses mit den Entwicklern stattfinden. Ein CASE-Tool kann so, beginnend mit der obersten Ebene der Variantenstückliste, systematisch den Entwickler befragen, welche Auswahl er zu einem gegebenen Einflussbereich treffen möchte. Im Beispiel könnte das System also fragen: „Welche Komponente soll zum Einflussbereich „Bilanzierung“ gewählt werden? Alternative a) IAS oder Alternative b) US-GAAP“. Die zu treffende Entscheidung wird dabei maßgeblich durch die Art der herzustellenden Variante bestimmt, die durch die Wünsche des Kunden explizit gegeben ist.

#### **4.2 Schritt 2: Suche nach potenziell geeigneten Komponenten**

Nachdem die Variantenknoten aus der Stückliste durch die entsprechenden Entscheidungen ersetzt worden sind, folgt nun sukzessive die Auflösung der Entscheidungsknoten. In diesem Schritt wird für jeden Entscheidungsknoten die dort hinterlegte Referenzspezifikation verwendet, um mit dieser auf den Komponentenmärkten nach geeigneten Komponenten zu suchen. Dabei kann sowohl eine Suche auf einem global zugänglichen Komponentenmarktplatz erwogen werden als auch eine Suche in einem unternehmensinternen Komponenten-Repository. Dabei ist es für die Vergleichbarkeit wichtig, dass Komponentenspezifikationen und Referenzspezifikationen dem gleichen Spezifikationsrahmen folgen.

Am Ende dieses Schrittes findet sich unter jedem Entscheidungsknoten eine Menge potenziell einsetzbarer Komponenten. Für das sich später anschließende Verfahren der Entscheidungsfindung ist es wichtig zu postulieren, dass mit dem Suchschritt jeweils alle möglichen Alternativen gefunden und somit berücksichtigt wurden. Daher ist dieser Schritt mit großer Sorgfalt durchzuführen.

#### **4.3 Schritt 3: Sichtung der gefundenen Komponenten**

Im dritten Schritt erfolgt für jeden Entscheidungsknoten eine erste Sichtung der gefundenen Alternativen. Hier werden zum Beispiel solche Alternativen entfernt, die die geforderten Satisfizierungsziele (also die K.O.-Kriterien) schon nicht erfüllen können oder solche, die von anderen Komponenten dominiert werden. Eine Komponente wird von einer anderen dominiert, falls diese in allen Attributen, die von der Referenzspezifikation vorgegeben werden, besser geeignet ist (also in jedem relevanten Attribut übertroffen) wird. Dabei kann die dominierende Komponente in den übrigen Attributen ihrer Spezifikation durchaus schlechter abschneiden – da diese laut Referenzspezifikation jedoch für die Entscheidung irrelevant sind, spielt dies im Sichtungs- und Entscheidungsprozess keine Rolle.

#### **4.4 Schritt 4: Auswahl der am besten geeigneten Komponenten**

Im vierten Schritt erfolgt dann die eigentliche Auswahl der am Besten geeigneten Kompo-

te aus der Menge der verbliebenen Alternativen. Dabei handelt es sich, wie im Beispiel gezeigt, um ein multiattributives Entscheidungsproblem [EiWe2003:115ff], d.h. es existieren mehrere zu berücksichtigende Ziele, die teilweise konkurrierend sein können. Darüber hinaus handelt es sich um eine Entscheidung unter Sicherheit, da die Alternativen nach der Suche und der sich anschließenden Vorauswahl unabhängig von weiteren Einflüssen sind.

Die Entscheidungsfindung kann auf zweierlei Arten ablaufen: Ist anzunehmen, dass die jeweiligen Entscheidungsknoten unabhängig voneinander sind, d.h. dass die jeweils dort gewählten Komponenten keinen Einfluss auf die zu wählenden Komponenten in anderen Entscheidungsknoten haben, so kann jeder Knoten autark entschieden werden. Ist dies (was in der Praxis anzunehmen ist) nicht der Fall, so müssen sämtliche durch die Stückliste ermöglichten Komponentenkombinationen aufgeführt werden und einander zur Entscheidungsfindung gegenübergestellt werden. Dies entspricht im Prinzip auch der Vorgehensweise, die ein Entwickler ohne ein entscheidungsunterstützendes System (etwa durch die Verwendung eines morphologischen Kastens [Over2002a]) wählen könnte.

Bei multiattributiven Entscheidungsproblemen spielt insbesondere die Gewichtung der einzelnen Ziele, beispielsweise in Form von prozentualen Anteilen der einzelnen Ziele, eine entscheidende Rolle. Die Gewichte können dazu benutzt werden, aus den verschiedenen Zielen eine einzelne (gemeinsame) Zielfunktion zu bestimmen, indem eine gewichtete Summe aus den einzelnen Zielkriterien berechnet wird. Vergibt man die Zielgewichte nach Gefühl, wird dies Direct-Ratio-Verfahren genannt. Zum Beispiel könnte ein Entwickler angeben, dass ihm ein niedriger Preis ebenso wichtig ist wie eine schnelle Antwortzeit. In diesem Fall würde er jeweils zu 50% den tatsächlichen Preis bzw. die tatsächliche Antwortzeit in die Zielfunktion einfließen lassen. Da das Direct-Ratio Verfahren einige logische Mängel aufweist, wurden in der Literatur systematischere Verfahren wie das Trade-Off- [EiWe2003:125ff] oder das Swing-Verfahren entwickelt [EiWe2003:129f]. Eine genauere Analyse der verschiedenen Entscheidungstechniken zur Auswahl von Softwarekomponenten findet sich bei [NcDe2002].

Ein weiteres Problem entsteht durch die Bewertung der jeweiligen Alternativen anhand der einzelnen Ziele. Bei Extremalzielen stellt sich die Frage der Bewertung der Alternativen mittels normierter Zahlenwerte, also beispielsweise, ob ein Einkaufspreis von 100€ gegenüber einem Preis von 1000€ oder eine maximale Antwortzeit bei einem Methodenaufruf von 1ms gegenüber einer von 10ms besser ist. Hinzu kommt, dass häufig Angaben mit unterschiedlichen Einheiten sowie unterschiedlich großen Intervallen der möglichen Ausprägungen miteinander verglichen werden müssen.

Kontio [Kont1995] schlägt hierfür einen Bewertungsprozess vor, den er in zwei Fallstudien [KoCB1996], [Kont1996] auf seine Praxistauglichkeit untersucht hat. Zur Veranschaulichung des Prozesses wird hier die aus der Betriebswirtschaftslehre bekannte Nutzwertanalyse verwendet. Hierbei geht es darum, für jede Alternative hinsichtlich eines jeden Ziels einen Nutzwert zu bestimmen. Zum Beispiel würde eine Alternative, die 100€ kostet, gegenüber einer gleichwertigen Lösung zu einem Preis von 1000€ einen höheren Nutzwert erhalten. Zu jeder Alternative wird dann, wie oben erläutert, die gewichtete Summe der Nutzwerte bestimmt und davon die größte ausgewählt. Da die erwähnten Alternativen ansonsten gleichwertig sein sollten, wird ihre Antwortzeit auf denselben Wert von beispielsweise jeweils 7ms gesetzt. Der Nutzwert von 100€ sei beispielsweise 10, der Nutzwert von 1000€ sei 1 und der Nutzwert von einer Antwortzeit von 7ms sei 5. Bei den oben erwähnten 50%-Gewichten der Zielfunktionen ergibt sich also ein Nutzwert von 10 bei der ersten und von 6 bei der zweiten Alternative, d.h. die billigere Alternative würde erwartungsgemäß ausgewählt.

Da die Nutzwertanalyse wie das Direct-Ratio-Verfahren einige Nachteile besitzt, hat Kontio den Analytic Hierachy Process (AHP) [Saat1990] eingesetzt, der einige der Nachteile der Nutzwertanalyse wettzumachen versucht.

#### 4.5 Schritt 5: Evaluation der getroffenen Komponentenauswahl

Nach der Ermittlung des Entscheidungsvorschlags im letzten Schritt wird dieser nun noch einmal auf seine tatsächliche Verwendbarkeit in der Praxis der Anwendungsentwicklung evaluiert. Dies entspricht sozusagen einer praktischen „Plausibilitätsprüfung“ bzw. Interpretation der Erkenntnisse, die durch ein systematisches Entscheidungsverfahren gewonnen wurden. Bei einer vollautomatischen Entscheidungsfindung mit einem ausgereiften Entscheidungsalgorithmus wird dieser Schritt entfallen.

#### 4.6 Schritt 6: Freigabe der Komponentenauswahl

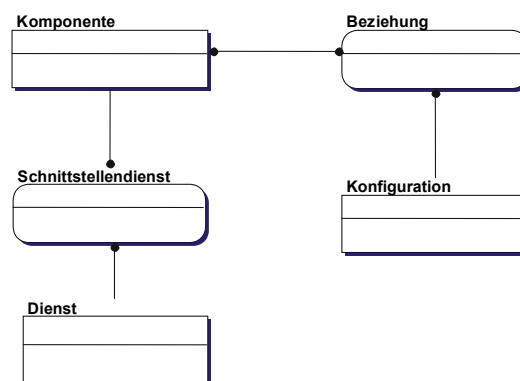
Nach der Bestätigung der Entscheidung für eine bestimmte Kombination von Komponenten wird diese zur Beschaffung freigegeben und zur Anwendungsentwicklung (Konfiguration) weitergegeben.

#### 4.7 Schritt 7: Ex-Post-Analyse des Entscheidungsprozesses

Der Auswahlprozess endet in einem letzten Schritt mit einer Ex-Post-Analyse der getroffenen Entscheidung, wenn sich absehen lässt, wie gut die gefundene Auswahl war. Die hierbei gewonnenen Erkenntnisse werden verwendet, um die beschriebene Vorgehensweise zu kontrollieren und permanent weiter zu verbessern.

### 5 Stücklistenbasiertes Änderungsmanagement

Neben einer Unterstützung beim Auswahlmanagement können Stücklisten auch beim Änderungsmanagement sinnvoll eingesetzt werden. Wird beispielsweise eine häufig in der Entwicklung verwendete Komponente durch eine andere ersetzt (sei es eine neue Version, eine fremde Komponente etc.), so lassen sich die nunmehr zu wartenden Anwendungen und Baugruppen durch Nachverfolgen derjenigen Stücklisten ermitteln, die diese Komponente jeweils als Teil beinhalten.



**Abbildung 5:** Metaschema des Komponentenmodells (basierend auf [ABC+2002]) und angeschlossene Stücklistenorganisation, dargestellt als ER-Diagramm

Ein solcher Verwendungsnachweis einer Komponente ist in der Regel in einem Dokumentationssystem (z.B. einem Repository [Ortn1999]), also einer speziellen Entwicklungsdatenbank, gespeichert. Abbildung 5 zeigt, wie eine Stücklistenorganisation beispielsweise auf das im GI-Spezifikationsrahmen [ABC+2002] enthaltene Dokumentationsschema einer Komponente aufgesetzt werden kann. Hierdurch ist es – ausgehend von der betroffenen Komponente – leicht möglich, alle Stücklisten, die diese Komponente enthalten, zu ermitteln und anschließend nach zu wartenden Konfigurationen (Anwendungen oder Baugruppen) auszuwerten.

Über die von Wartungen betroffenen Konfigurationen hinaus vermag eine Stückliste auch anzugeben, wie oft eine Komponente als Teil während der jeweiligen Anwendungsentwicklung verwendet wurde. Hierdurch lässt sich also sicherstellen, dass jede Anwendung vollständig gewartet und an Änderungen angepasst wird. Gleichzeitig dient dieser mengenmäßige Nachweis auch der Schätzung des bei einer Änderung zu erwartenden Wartungsaufwands.

Etwas komplizierter stellt sich die Verwendung von Variantenstücklisten für das Änderungsmanagement dar. In diesem Fall werden zusätzlich zu der eigentlichen Stückliste auch die bei der Komponentenauswahl getroffenen Entscheidungen (als Eingabewerte für die Auswertung der Stückliste) benötigt.

## **6 Zusammenfassung und Ausblick**

Es wurde gezeigt, wie der Einsatz von Stücklisten als Methode im Konfigurationsmanagement den Prozess der komponentenorientierten Anwendungsentwicklung weiter verbessern kann. Insbesondere der Nutzen bei der Komponentenauswahl und im Änderungsmanagement wurde dabei angesprochen.

Durch die Verbindung eines Standards zur Spezifikation von Fachkomponenten, grundlegender Verfahren aus der Entscheidungstheorie sowie Variantenstücklisten wurde ein systematischer (rationaler) Prozess zur Entscheidungsfindung bei der Auswahl von Komponenten vorgestellt, der die bislang üblichen Ad-hoc-Entscheidungen ablösen könnte. Insbesondere die Methoden der Entscheidungstheorie können weiter auf diesen Einsatzzweck angepasst werden. Zum Beispiel könnte die Einführung von Präferenzfunktionen, die auf einem Komponentenmarktplatz gewonnen werden könnten, dazu dienen, den Kunden beim Kauf einer Komponente gleich mitzuteilen, mit welchen anderen Komponenten diese bevorzugt verwendet wird, um somit mögliches Cross-Selling Potential auszuschöpfen.

Auch eine weitergehende Untersuchung der Ebenen des Spezifikationsrahmens sollte dazu führen, den Auswahlprozess weiter zu verbessern. So könnte eine erste Untersuchung mögliche Satisfizierungsziele und Extremalziele unterscheiden. Daneben wäre ein Algorithmus für den Kompatibilitätstest auf der Basis von Komponentenspezifikationen wünschenswert. Dieser ist derzeit in Planung. Eine Herausforderung besteht darin, die verschiedenen Ebenen des Spezifikationsrahmens auf Kompatibilität zu prüfen, beispielsweise also auf der Terminologieebene festzustellen, ob eine Komponente „Bilanzierung nach US-GAAP“ kompatibel zu einer Komponente „Bilanzierung nach IAS“ ist. Schließlich wäre es denkbar, auf dem Spezifikationsrahmen eine Zielhierarchie aufzubauen, die in vielen Entscheidungsmethoden (z.B. dem AHP) verwendet werden kann. Diese theoretischen Untersuchungen am Spezifikationsrahmen sollten dabei durch den Versuch ergänzt werden, das Konzept auch als Ganzes in einem CASE-Tool zu implementieren. Insbesondere der Einsatz von Variantenstücklisten im Anwendungsentwurf muss dabei durch entsprechende Erfahrungen in der Praxis noch weiter untersucht werden.

## Literatur

- [ABC+2002] *Ackermann, J.; Brinkop, F.; Conrad, S.; Fettke, P.; Frick, A.; Glistau, E.; Jaekel, H.; Kotlar, O.; Loos, P.; Mrech, H.; Ortner, E.; Overhage, S.; Raape, U.; Sahm, S.; Schmietendorf, A.; Teschke, T.; Turowski, K.*: Vereinheitlichte Spezifikation von Fachkomponenten. Gesellschaft für Informatik (GI), Arbeitskreis 5.10.3, Augsburg, 2002. <http://www.fachkomponenten.de>, Abruf am 20.11.2002.
- [EiWe2003] *Eisenführ, F.; Weber, M.*: Rationales Entscheiden. 4. Auflage, Springer Verlag, Berlin 2003.
- [Hahn2000] *Hahn, H.*: Ein Modell zur Ermittlung der Reife des Softwaremarktes. In: *Turowski, K. (Hrsg.)*: Tagungsband 4. Workshop Komponentenorientierte betriebliche Anwendungssysteme. Universität Augsburg, Juni 2002, S. 75 – 85.
- [HoNo2001] *Houston, K.; Norris, D.*: Software Components and the UML. In: *Heineman, G. T.; Councill, W. T. (Hrsg.)*: Component-Based Software Engineering – Putting the Pieces Together. Addison-Wesley, Upper Saddle River, 2001, S. 243 – 262.
- [KoCB1996] *Kontio, J.; Caldiera, G.; Basili, V.R.*: Defining Factors, Goals and Criteria for Reusable Component Evaluation, Presented at the CASCON '96 conference, Toronto, Canada, 1996.
- [KoHe1996] *Kolisch, R.; Hempel, K.*: Auswahl von Standardsoftware, dargestellt am Beispiel von Programmen für das Projektmanagement. In: *Wirtschaftsinformatik 38 (1996) 4*, S. 399 – 410.
- [Kont1995] *Kontio, J.*: OTSO: A Systematic Process for Reusable Software Component Selection, Institute for Advanced Computer Studies and Department of Computer Science, University of Maryland, Collage Park, USA. [http://www.soberit.hut.fi/~jkontio/jko\\_pub.htm](http://www.soberit.hut.fi/~jkontio/jko_pub.htm), Abruf am 22.11.2002.
- [Kont1996] *Kontio, J.*: A Case Study in Applying a Systematic Method for COTS Selection, Proceedings: 18th International Conference on Software Engineering in Berlin, IEEE, 1996.
- [LaCr2000] *Larsson, M.; Crnkovic, I.*: Component Configuration Management. In: *Szyperski, C. (Hrsg.)*: Proceedings of WCOP 2000. <http://www.mrtc.mdh.se/publication/0236.pdf>, Abruf am 22.11.2002.
- [Mcil1968] *McIlroy, M. D.*: Mass Produced Software Components. In: *Naur, P., Randell, B. (Hrsg.)*: Software Engineering: Report on a Conference by the NATO Science Committee. NATO Scientific Affairs Division, Brussels, 1968, S. 138–150.
- [NcDe2002] *Ncube, C.; Dean, J. C.*: The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components. In: *Dean, J. C.; Gravel, A. (Hrsg.)*: COTS-Based Software Systems, ICCBSS 2002, Lecture Notes in Computer Science 2255, Springer 2002, S. 176 – 187
- [Ortn1998] *Ortner, E.*: Ein Multipfad-Vorgehensmodell für die Entwicklung von Informationssystemen – dargestellt am Beispiel von Workflow-Management-Anwendungen. In: *Wirtschaftsinformatik 40 (1998) 4*, S. 329 – 337.
- [OrLK1999] *Ortner, E.; Lang, K.-P.; Kalkmann, J.*: Anwendungsentwicklung mit Komponenten. In: *Information Management & Consulting 14 (1999) 2*, S. 35 – 45.
- [Ortn1999] *Ortner, E.*: Repository Systems. Teil 1: Mehrstufigkeit und Entwicklungsumgebung. In: *Informatik Spektrum 22 (1999) 4*, S. 235 – 251. Teil 2: Aufbau und Betrieb eines Entwicklungsrepositoriums. In: *Informatik Spektrum 22 (1999) 5*, S. 351 – 363.
- [Ortn2001] *Ortner, E.*: Komponentenorientierte Anwendungsentwicklung. In: *Management & Consulting 15 (2001) 4*, S. 62 – 72.
- [Over2002a] *Overhage, S.*: Die Spezifikation – kritischer Erfolgsfaktor der Komponentenorientierung. In: *Turowski, K. (Hrsg.)*: Tagungsband 4. Workshop Komponentenorientierte betriebliche Anwendungssysteme. Universität Augsburg, Juni 2002, S. 1 – 17.
- [Over2002b] *Overhage, S.*: Komponentenkataloge auf Basis eines einheitlichen Spezifikationsrahmens – ein Implementierungsbericht. In: *Turowski, K. (Hrsg.)*: Tagungsband 3. Workshop Modellierung

und Spezifikation von Fachkomponenten. Universität Augsburg, September 2002, S. 1 – 16.

- [OvTh2002] *Overhage, S.; Thomas, P.:* On Specifying Web Services Using UDDI Improvements. In: *Dittmar, T. et. al (Hrsg.):* Proceedings Netobjectdays Workshops 2002. IGT, Leipzig, 2002, S. 535 – 550.
- [Saat1990] *Saaty, T. L.:* The Analytic Hierarchy Process. McGraw-Hill, New York, 1990.
- [Schn1999] *Schneeweiß, C.:* Einführung in die Produktionswirtschaft. 7. Auflage, Springer Verlag, Berlin 1999.
- [ShGa1996] *Shaw, M., Garlan, D.:* Software Architecture – Perspectives on an Emerging Discipline. Prentice Hall, Upper Saddle River, New Jersey, 1996.
- [Szyp1998] *Szyperski, C.:* Component Software: Beyond Object-Oriented Programming. Addison-Wesley, Harlow 1998.
- [WeMü1981] *Wedekind, H.; Müller, T.:* Stücklistenorganisation bei einer großen Variantenzahl. In: *Angeordnete Informatik* 23 (1981) 9, S. 377 – 383.
- [Zimm1988] *Zimmermann, G.:* Produktionsplanung variantenreicher Erzeugnisse mit EDV. Springer Verlag, Berlin 1988.