

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES  
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro



(43) Internationales Veröffentlichungsdatum  
19. Juni 2003 (19.06.2003)

PCT

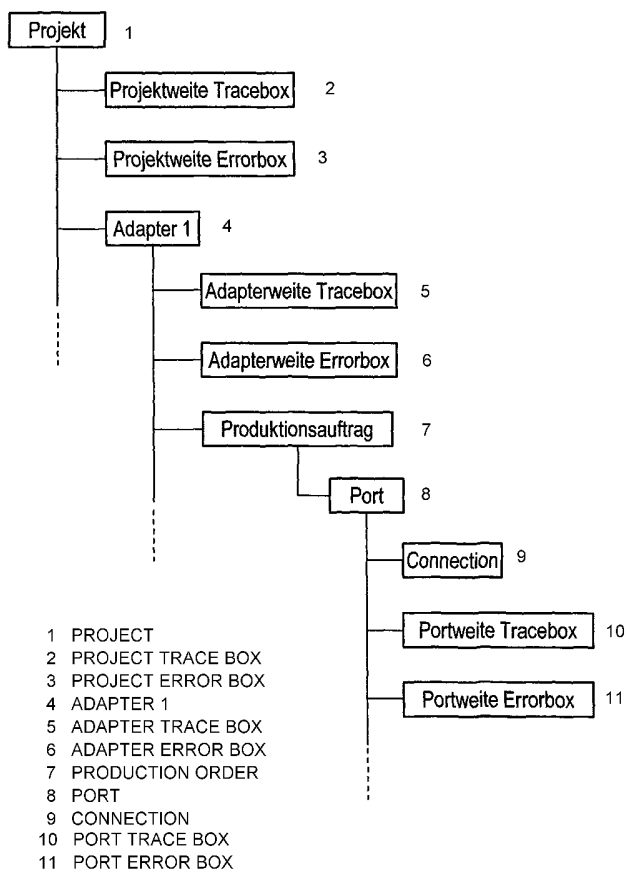
(10) Internationale Veröffentlichungsnummer  
**WO 03/050676 A2**

- (51) Internationale Patentklassifikation<sup>7</sup>: **G06F 9/40**
- (21) Internationales Aktenzeichen: PCT/DE02/04382
- (22) Internationales Anmeldedatum:  
28. November 2002 (28.11.2002)
- (25) Einreichungssprache: Deutsch
- (26) Veröffentlichungssprache: Deutsch
- (30) Angaben zur Priorität:  
101 61 140.4 12. Dezember 2001 (12.12.2001) DE
- (71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): **SIEMENS AKTIENGESELLSCHAFT** [DE/DE];  
Wittelsbacherplatz 2, 80333 München (DE).
- (72) Erfinder; und  
(75) Erfinder/Anmelder (nur für US): **THURNER, Elmar** [AT/DE]; Gneisenaustrasse 12, 90491 Nürnberg (DE).  
**LANGKAFEL, Dirk** [DE/DE]; Bergstr. 15a, 91090 Efeltrich (DE). **MERKL, Thomas** [DE/DE]; Schlesienstr. 9, 95478 Kemnath (DE).
- (74) Gemeinsamer Vertreter: **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).
- (81) Bestimmungsstaat (national): US.
- (84) Bestimmungsstaaten (regional): europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).

[Fortsetzung auf der nächsten Seite]

(54) Title: SYSTEM AND METHOD FOR TRACING AND/OR EVALUATING THE EXCHANGE OF INFORMATION

(54) Bezeichnung: SYSTEM UND VERFAHREN ZUM VERFOLGEN UND/ODER AUSWERTEN DES INFORMATIONSAUSTAUSCHES



(57) Abstract: The invention relates to a system and a method for tracing and evaluating the communication of software applications, especially MES applications (A1 - A6) in an entire system. The tracing and evaluation can be carried out progressively or step-by-step on a project level, an adapter level and a port level, i.e. individual communication connections and individual applications (A1 - A6), but the entire project is also traced by (even heterogeneous) application environments. The tracing (TB, EB) and evaluation mechanisms (AE) enable higher-level services to access online the archived trace and error data at all times. An example of one such service is the generation of a report in the pharmaceutical field, indicating the materials used and sent in production orders.

(57) Zusammenfassung: System und Verfahren zur Verfolgung und Auswertung der Kommunikation von Softwareanwendungen, insbesondere MES-Anwendungen (A1 - A6) in einem Gesamtsystem. Es wird eine skalierte bzw. stufenweise Verfolgung und Auswertung ermöglicht: Projektebene, Adapterebene und auf Portebene, d.h. einzelne Kommunikationsverbindungen, einzelne Anwendungen (A1 - A6), aber auch das Gesamtprojekt wird über (auch heterogene) Anwendungslandschaften erfasst. Die Erfassungs- (TB, EB) und Auswertungsmechanismen (AE) erlauben höherwertigen Diensten online jederzeit Zugriff auf die archivierten Trace- und Fehlerdaten. Ein Beispiel solch eines Dienstes wäre die Generierung eines Reports, der pharmazeutischen Bereich die verwendeten Materialien nachweist, die innerhalb von Produktionsaufträgen verschickt wurden.

WO 03/050676 A2

**Erklärungen gemäß Regel 4.17:**

- hinsichtlich der Berechtigung des Anmelders, ein Patent zu beantragen und zu erhalten (Regel 4.17 Ziffer ii) für die folgenden Bestimmungsstaaten europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR)
- Erfindererklärung (Regel 4.17 Ziffer iv) nur für US

**Veröffentlicht:**

- ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

## Beschreibung

System und Verfahren zum Verfolgen und/oder Auswerten des Informationsaustausches

5

Die Erfindung bezieht sich auf ein System und Verfahren zum Verfolgen und/oder Auswerten des Informationsaustausches zwischen insbesondere heterogenen Softwareapplikationen, insbesondere MES-Applikationen, in einem die Softwareapplikationen koppelnden Softwaresystem, wobei die Softwareapplikationen und das Softwaresystem auf mindestens einer Rechneinheit gespeichert sind.

Ferner betrifft die Erfindung ein entsprechendes Computerprogramm, ein Computerprogrammprodukt sowie eine Datenverarbeitungseinrichtung.

Aus "Software für die Automatisierung - Transparenz über die Abläufe schaffen", Artikel von Dirk Kozian in Elektronik für die Automatisierung 11, 17.11.1999 ist bekannt, für die Automatisierung von Produktions- bzw. Fertigungsabläufen so genannte Manufacturing Execution Systems (MES) einzusetzen. Diese Systeme integrieren die Automatisierungsebene (Controls) mit den ERP-Systemen (ERP: Enterprise Resource Planning) der Unternehmensleitebene. Manufacturing Execution Systems sind Systeme, die z.B. Informationen zur Optimierung von Produktionsabläufen bereitstellen. Zum einen müssen die Manufacturing Execution Systems die groben Planungsdaten der ERP-Systeme um anlagenspezifische und aktuelle Feinplanungsdaten ergänzen und diese entsprechend an die unterlagerte Automatisierungsebene weiterleiten, zum anderen haben sie die Aufgabe, aus der Automatisierungsebene produktionsrelevante Informationen zu übernehmen, diese aufzubereiten und an die Unternehmensleitebene weiterzumelden. MES-Systeme erfüllen somit die Aufgabe einer vertikalen Integration zwischen der Unternehmensleitebene und der Automatisierungsebene. Typische Einzelaufgaben von MES-Systemen sind Enterprise Asset Manage-

ment, Maintenance Management, Information Management, Scheduling, Dispatching und Trace & Track. Diese Aufgaben werden jeweils von MES-Komponenten bzw. MES-Applikationen ausgeführt.

5

Aufgrund der software- und datentechnischen Heterogenität der MES-Applikationen lassen sich diese aber nur sehr schwer in ein gemeinsames System bzw. Projekt integrieren und der Datenaustausch zwischen diesen Applikationen ist nur mit Aufwand möglich.

10

Aus "Massive Wiederverwendung: Konzepte, Techniken und Organisation", Artikel von Ulrich Lindner in OBJEKTSpektrum 1/96, S. 10 - 17, ist es bekannt, Softwarekomponenten durch so genannte Adapter oder durch Wrapping (Verpacken) in ein Softwaresystem zu integrieren. Ziel ist es dabei die Wiederverwendbarkeit von Softwarekomponenten zu erhöhen.

15

In komplexen Systemen, wie z.B. MES-Systemen, ist es sehr schwierig den Informationsaustausch zwischen den beteiligten Komponenten zu überwachen und zu verifizieren.

20

In US 6,083,281 ist ein Verfahren beschrieben, um die Aktivitäten von Entitäten in Softwaresystem zu verfolgen. Dieses Verfahren bezieht sich aber auf die Arbeitsweise der Entitäten und nicht auf den Datenaustausch zwischen den Entitäten. Dieses Verfahren soll Softwareentwickler in der Entwicklungs- und Testphase unterstützen.

25

Des Weiteren sind vom Betriebssystem Windows NT von der Firma Microsoft ein "Event-Log"-Mechanismus und eine "Ereignisanzeige" bekannt.

30

Aufgabe der vorliegenden Erfindung ist es, ein System und ein Verfahren zur Verfolgung, Auswertung und Protokollierung der Kommunikation von Softwarekomponenten, insbesondere in MES-Systemen, zur Verfügung zu stellen.

35

Gemäß der Erfindung wird die oben genannte Aufgabe für ein System durch die Merkmale des Anspruchs 1 gelöst. Ein wesentlicher Vorteil der Erfindung liegt in der einfachen und automatischen Verifikation des Datenaustausches. Es kann sicher-

5 gestellt werden, ob vorgesehene Informationen auch wirklich ausgetauscht wurden. D.h. es kann verifiziert werden, ob Informationen auch wirklich von einem Sender abgeschickt wurden, bzw. beim Empfänger angekommen sind. Weiterhin können aufgetretene Übertragungsfehler oder Flaschenhalse sehr

10 leicht erkannt und analysiert werden. Daraus lassen sich Optimierungen für das zugrundeliegende Softwaresystem bzw. Projekt ableiten. Ein weiterer Vorteil besteht darin, dass sehr einfach Nachweisverpflichtungen gegenüber Behörden, z.B. FDA vorgenommen werden können (z.B. durch generierte Reports).

15 Eine erste vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein System liegt darin, dass die Ergebnisse der Analyse des Informationsaustausches und/oder der Analyse der Fehler vom Softwaresystem direkt weiterverarbeitbar sind. Die

20 Ergebnisse der Analyse und die daraus gezogenen Schlüsse (z.B. Optimierungsmöglichkeiten) können direkt im Softwaresystem berücksichtigt werden (z.B. Änderung der Konfiguration von Komponenten, Dimensionierung oder Ausgestaltung von Kommunikationsverbindungen). Dadurch entsteht ein "quasi geschlossener selbstregulierender" Regelkreis.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein System liegt darin, dass durch eine Filterbildung bestimmte Informationen und/oder Fehler auswählbar

30 sind. Ein Anwender kann dadurch sehr leicht gezielt auf Informationen zugreifen. Möglich Filter sind z.B. Absender, Empfänger, Zeitraum, IDs von Nachrichten. Die Filter können z.B. über Eingabemasken und Eingabehilfsmittel (Tastatur, Maus etc.) definiert werden. Dadurch hat ein Anwender die

35 Möglichkeit einer skalierten und dedizierten Informationsbeschaffung.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein System liegt darin, dass die Verfolgung und/oder Auswertung auf Projektebene oder auf Adapterebene oder auf Ebene eines Informationskanals erfolgt. Dadurch ist  
5 eine skalierte Informationsbeschaffung und Auswertung auf unterschiedlichen Abstraktionsebenen (Projektebene, Adapterebene, Portebene) möglich.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein System liegt darin, dass als Koppelmechanismen Adapter und/oder Wrapper eingesetzt sind. Adapter- und Wrappertechnologien sind in der Informationstechnologie bekannte Mechanismen, um Softwarekomponenten in ein System zu integrieren. Ein Wrapper bildet das API (Application Program-  
15 mable Interface) einer Fremdkomponente (z.B. eine MES-Applikation eines Drittanbieters) in das Objektmodell des Rahmenprogramms ab. So wird z.B. aus einer Methode des API der Fremdkomponente eine Methode des Rahmenprogramms bzw. aus einem Integer-Datentyp des API der Fremdkomponente wird ein  
20 Integer-Datentyp des Rahmenprogramms, usw. Für COM-Objekte (Component Object Model) ist die Erstellung eines Wrappers für eine zu integrierende Komponente automatisierbar. Ein Wrapper entspricht einer Bridge. Wrapper lassen sich sehr schnell realisieren.

25 Adapter sind eine Abstraktionsstufe höher als Wrapper. Sie bieten eine einheitliche Sicht auf angekoppelte Applikationen. Ein Adapter bietet Funktionalität, um die anzukoppelnde Komponente zu starten, zu bedienen etc. Ein Adapter entspricht in der Sprache der Design Patterns einer "Facade".  
30

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein System liegt darin, dass die erste und zweite Einrichtung durch eine Einrichtung realisiert sind oder die  
35 erste, zweite und vierte Einrichtung durch eine Einrichtung realisiert sind oder die erste und vierte Einrichtung durch eine Einrichtung realisiert sind oder die zweite und vierte

Einrichtung durch eine Einrichtung realisiert sind. Dadurch wird eine kompakte und ressourcenminimierende (z.B. Speicherplatzeinsparung) Bauweise ermöglicht.

- 5 Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein System liegt darin, dass die dritte Einrichtung in eine der anderen Einrichtungen integriert ist. Auch dadurch wird eine kompakte und ressourcenminimierende (z.B. Speicherplatzeinsparung) Bauweise ermöglicht. Weiterhin können  
10 dadurch Zugriffszeiten auf Datenbankinhalte optimiert werden.

- Gemäß der Erfindung wird die oben genannte Aufgabe für ein Verfahren durch die Merkmale des Anspruchs 8 gelöst. Ein wesentlicher Vorteil der Erfindung liegt in der einfachen und  
15 automatischen Verifikation des Datenaustausches. Es kann sichergestellt werden, ob vorgesehene Informationen auch wirklich ausgetauscht wurden. D.h. es kann verifiziert werden, ob Informationen auch wirklich von einem Sender abgeschickt wurden, bzw. beim Empfänger angekommen sind. Weiterhin können  
20 aufgetretene Übertragungsfehler oder Flaschenhälse sehr leicht erkannt und analysiert werden. Daraus lassen sich Optimierungen für das zugrundeliegende Softwaresystem bzw. Projekt ableiten. Ein weiterer Vorteil besteht darin, dass sehr  
25 einfach Nachweisverpflichtungen gegenüber Behörden, z.B. FDA vorgenommen werden können (z.B. durch generierte Reports).

- Eine erste vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein Verfahren liegt darin, dass die Ergebnisse der  
30 Analyse des Informationsaustausches und/oder der Analyse der Fehler vom Softwaresystem direkt weiterverarbeitet werden. Die Ergebnisse der Analyse und die daraus gezogenen Schlüsse (z.B. Optimierungsmöglichkeiten) können direkt im Softwaresystem berücksichtigt werden (z.B. Änderung der Konfiguration  
35 von Komponenten, Dimensionierung oder Ausgestaltung von Kommunikationsverbindungen). Dadurch entsteht ein "quasi geschlossener selbstregulierender" Regelkreis.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein Verfahren liegt darin, dass durch eine Filterbildung bestimmte Informationen und/oder Fehler ausgewählt werden. Ein Anwender kann dadurch sehr leicht gezielt auf Informationen zugreifen. Möglich Filter sind z.B. Absender, Empfänger, Zeitraum, IDs von Nachrichten. Die Filter können z.B. über Eingabemasken und Eingabehilfsmittel (Tastatur, Maus etc.) definiert werden. Dadurch hat ein Anwender die Möglichkeit einer skalierten und dedizierten Informationsbeschaffung.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein Verfahren liegt darin, dass die Verfolgung und/oder Auswertung auf Projektebene oder auf Adapterebene oder auf Ebene eines Informationskanals durchgeführt wird. Dadurch ist eine skalierte Informationsbeschaffung und Auswertung auf unterschiedlichen Abstraktionsebenen (Projektebene, Adapterebene, Portebene) möglich.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein Verfahren liegt darin, dass als Koppelmechanismen Adapter und/oder Wrapper eingesetzt werden. Adapter- und Wrappertechnologien sind in der Informationstechnologie bekannte Mechanismen, um Softwarekomponenten in ein System zu integrieren. Ein Wrapper bildet das API (Application Programmable Interface) einer Fremdkomponente (z.B. eine MES-Applikation eines Drittanbieters) in das Objektmodell des Rahmenprogramms ab. So wird z.B. aus einer Methode des API der Fremdkomponente eine Methode des Rahmenprogramms bzw. aus einem Integer-Datentyp des API der Fremdkomponente wird ein Integer-Datentyp des Rahmenprogramms, usw. Für COM-Objekte (Component Object Model) ist die Erstellung eines Wrappers für eine zu integrierende Komponente automatisierbar. Ein Wrapper entspricht einer Bridge. Wrapper lassen sich sehr schnell realisieren.

Adapter sind eine Abstarktionsstufe höher als Wrapper. Sie bieten eine einheitliche Sicht auf angekoppelte Applikationen. Ein Adapter bietet Funktionalität, um die anzukoppelnde Komponente zu starten, zu bedienen etc. Ein Adapter entspricht in der Sprache der Design Patterns einer "Facade".

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein Verfahren liegt darin, dass die erste und zweite Einrichtung durch eine Einrichtung realisiert werden oder die erste, zweite und vierte Einrichtung durch eine Einrichtung realisiert werden oder die erste und vierte Einrichtung durch eine Einrichtung realisiert werden oder die zweite und vierte Einrichtung durch eine Einrichtung realisiert werden. Dadurch wird eine kompakte und ressourcenminimierende (z.B. Speicherplatzeinsparung) Bauweise ermöglicht.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung für ein Verfahren liegt darin, dass die dritte Einrichtung in eine der anderen Einrichtungen integriert ist. Auch dadurch wird eine kompakte und ressourcenminimierende (z.B. Speicherplatzeinsparung) Bauweise ermöglicht. Weiterhin können dadurch Zugriffszeiten auf Datenbankinhalte optimiert werden.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass das erfindungsgemäße Verfahren durch ein Computerprogramm implementiert ist. Dadurch können eventuelle Modifizierungen bzw. Anpassungen leicht durchgeführt werden.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass das Computerprogramm für das erfindungsgemäße Verfahren auf einem Datenträger gespeichert ist. Dadurch ist das Verfahren bezüglich der Logistik und Verteilung leicht handhabbar.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass das Computerprogramm für das erfindungsgemäße Verfahren auf einer Datenverarbeitungseinrichtung installiert ist. Dadurch wird die Performance erhöht.

5

Weitere Vorteile und Details der Erfindung ergeben sich anhand der nun folgenden Beschreibung vorteilhafter Ausführungsbeispiele und in Verbindung mit den Figuren. Soweit in unterschiedlichen Figuren Elemente mit gleichen Funktionalitäten beschrieben sind, sind diese mit gleichen Bezugszeichen gekennzeichnet.

10

Es zeigen:

15 FIG 1 in einer Übersichtsdarstellung die "Unternehmenspyramide" mit drei Steuerungsebenen,

FIG 2 ein beispielhaftes Übersichtsbild mit Software- und Hardwareeinheiten für MES-Lösungen,

20

FIG 3 die zentrale Stellung des die Softwareapplikationen koppelnden Rahmenprogramms,

FIG 4 den prinzipiellen Aufbau eines Adapters,

25

FIG 5 mehrere verschaltete Adapter,

FIG 6 eine funktionelle Prinzipdarstellung der Erfindung,

30 FIG 7 ein Instanzendiagramm zu einem Projekt,

FIG 8 ein Sequence Chart: Initialisierung,

FIG 9 ein Sequenze Chart: Betrieb,

35

FIG 10 ein Sequence Chart: Auswertemodus,

FIG 11 eine "Component" als Metaobjektmodell in UML-  
Notation,

FIG 12 eine Kommunikationsbeziehung zwischen MES-  
5 Applikationen und

FIG 13 die Objektstruktur einer Connection in UML.

Die Darstellung gemäß FIG 1 zeigt in einer Übersichtsdarstellung die drei Steuerungsebenen, wie sie üblicherweise in einem produzierenden bzw. fertigenden Unternehmen zu finden sind. Durch die Pyramidenform wird ausgedrückt, dass nach oben hin eine Verdichtung der Informationen stattfindet. Die oberste Ebene ist die ERP-Ebene (Enterprise Resource Planning. Auf dieser Unternehmensleitebene werden üblicherweise die betriebswirtschaftlichen und vertrieblichen Aufgaben in einem Unternehmen durchgeführt (z.B. Finanzwesen, Vertriebswesen, Personalwesen, Berichterstattung). Aber auch produktionsanlagenübergreifende logistische Aufgaben (z.B. Auftrags- und Materialverwaltung) werden auf dieser Ebene durchgeführt. Das System SAP R/3 ist ein ERP-System, das auf der Unternehmensleitebene sehr häufig verwendet wird.

Die unterste Ebene der Pyramide ist die Automatisierungsebene (Controls). Auf dieser Ebene kommen üblicherweise speicherprogrammierbare Steuerungen (SPS) in Verbindung mit Visualisierungs- und Prozessleitsystemen (PLS) zum Einsatz. Die Antriebe, Aktoren und Sensoren der Produktions- und/oder Fertigungseinrichtungen stehen direkt mit den Systemen dieser Ebene in Verbindung.

Das Verbindungsglied zwischen der ERP-Ebene und der Automatisierungsebene wird durch die MES-Ebene gebildet. Die Applikationen der MES-Ebene sorgen somit für eine vertikale Integration zwischen der ERP-Ebene und der Automatisierungsebene. Die MES-Applikationen müssen einerseits die Grobplanungen der ERP-Systeme um produktionsanlagen spezifische Feinplanungen

ergänzen und an die Systeme der Automatisierungs-Ebene weiterleiten, andererseits ist es Aufgabe der MES-Applikationen produktionsrelevante Daten der Automatisierungs-Ebene aufzunehmen, aufzubereiten und an die ERP-Ebene (Unternehmensleitungsebene) weiterzuleiten.

Typische MES-Applikationen sind u.a. Quality Management (QM), Maintenance Management (MM), Performance Analysis (PA), Process Management, Labor Management, Asset Management. Durch jeweils drei Punkte wird in FIG 1 ausgedrückt, dass sich auf einer Ebene weitere Elemente (Applikationen, Systeme etc.) befinden können.

MES-Systeme bzw. ERP-Systeme enthalten in der Regel ein so genanntes Laufzeitsystem zur zeitlichen Ablaufsteuerung der beteiligten Komponenten (Teilkomponenten, Module, Tasks, Prozesse des Betriebssystems etc.), sowie ein so genanntes Engineeringssystem zum Erstellen und Editieren von Programmen, welche zur Ausführung im Laufzeitsystem vorgesehen sind.

Die Darstellung gemäß FIG 2 zeigt ein beispielhaftes Übersichtsbild mit Software- und Hardwareeinheiten für MES-Lösungen. Die einzelnen MES-Applikationen A1 bis A3 sind über Adapter AD1 bis AD3 mit einem Rahmenprogramm (Framework) IF verbunden. Über einen bidirektionalen Informationspfad I1 ist ein Benutzerarbeitsplatz PIW1 mit dem Rahmenprogramm IF gekoppelt und kann somit die daran hängenden bzw. integrierten MES-Applikationen verwalten und überwachen. Der Benutzerarbeitsplatz PIW1 besteht üblicherweise aus einer Anzeigevorrichtung (Monitor, Display, etc.), einer Datenverarbeitungsanlage (z.B. PC) mit Prozessor und Speichereinrichtungen sowie Eingabeeinheiten (Keyboard, Maus, etc.). Die MES-Applikationen A1 bis A3 sowie das Rahmenprogramm IF können auf eigenen Datenverarbeitungseinheiten bzw. Prozessoren ablaufen, es ist aber auch möglich, dass sie auf der Datenverarbeitungseinheit des PIW1 ablaufen.

Über Adapter AD1 bis AD3 sind die jeweiligen MES-Applikationen A1 bis A3 mit dem Rahmenprogramm IF verbunden. Die Adapter sind somit die Koppelbausteine zwischen dem Rahmenprogramm IF und den Applikationen. Über die Adapter können  
5 somit auch an sich heterogene Applikationen miteinander verbunden werden, und durch die Integration mit dem Rahmenprogramm IF ist es möglich, zwischen den Applikationen zu kommunizieren und Datenaustausch zu betreiben. Die Adapter sind Softwaremodule, die Verbindungen zu verschiedenen Anwendungen  
10 bzw. Applikationen herstellen. In typischen Integrationsszenarien sind dies Integrationen zu Systemen aus der MES-, ERP-, SCADA- oder Controls-Welt Ein Adapter bietet Funktionalität, um eine anzukoppelnde Komponente zu starten, zu bedienen, etc. Ein Adapter erlaubt den Zugriff auf Daten und Funktionen der anzukoppelnden Anwendung bzw. Applikation, stellt  
15 bestimmte Runtimedaten zur Verfügung und erlaubt das Laden von Engineeringinformationen aus der anzukoppelnden Anwendung bzw. Applikation. Adapter können sich hinsichtlich ihres Aufbaus und Umfangs unterscheiden. So können Adapter z.B. fest  
20 programmiert sein oder sie können konfiguriert bzw. modelliert werden. Auch bezüglich der Möglichkeiten des Zugriffs auf die anzukoppelnde Applikation können sie sich unterscheiden, so können z.B. Adapter nur einen datentechnischen Zugang gestatten, es ist aber auch möglich, dass Adapter einen Zugang  
25 auf höherwertige Geschäftsabläufe zulassen. Beim Hochfahren werden die Adapter mit den hinterlegten Modellen und Zustandsinformationen geladen. Zur Laufzeit wird dann überprüft, ob und wie die unterschiedlichen integrierten Applikationen bzw. Anwendungen zusammenpassen. Über eine Visualisierungs- bzw. Monitoringkomponente ist es möglich, den Status  
30 eines Adapters abzufragen und am Benutzerarbeitsplatz PIW1 darzustellen (auch graphisch). Durch Adapter erhält das System und auch der Anwender eine standardisierte und einheitliche Sicht auf Applikationen (je nachdem, welche Abstraktions-  
35 ebene bei den Adaptern vorhanden ist).

- Eine weitere Möglichkeit Softwarekomponenten zu integrieren, ist es, Wrapper einzusetzen. Ein Wrapper bildet das API (Application Programmable Interface) einer Fremdkomponente (z.B. eine MES-Applikation) in das Objektmodell des Rahmenprogrammes ab. So wird z.B. aus einer Methode des API der Fremdkomponente eine Methode des Rahmenprogramms bzw. aus einem Integer-Datentyp des API der Fremdkomponente wird ein Integer-Datentyp des Rahmenprogramms.
- 10 Neben MES-Applikationen können auch Applikationen aus der Unternehmensleitebene (Enterprise Resource Planning-Ebene) und/aus der Automatisierungsebene (Controls-Ebene) über das Rahmenprogramm IF integriert werden und über den Arbeitsplatz PIW1 (das Acronym PIW steht für Personalized Industrial
- 15 Workplace) überwacht bzw. verwaltet werden. Das Rahmenprogramm IF bildet somit eine Integrationsplattform für den gesamten industriellen Bereich. Unterschiedliche Applikationen aus der Unternehmensleitebene, der MES-Ebene und der Automatisierungsebene lassen sich durch das Rahmenprogramm IF einfach und wirtschaftlich mit Hilfe von Adaptern und/oder Wrap-
- 20 pern integrieren. Das Rahmenprogramm IF ist somit als Middleware-Plattform und als Manufacturing Application Integration-Werkzeug anzusehen. Über den Arbeitsplatz PIW1 kann ein Benutzer (z.B. der Anlagenfahrer) die jeweiligen Zustände der
- 25 zu überwachenden Applikationen sehen, und er kann auch auf Daten und auf Methoden der Applikationen zugreifen, und weiterhin kann er durch diesen Zugriff Applikationen miteinander in Verbindung setzen.
- 30 Das Rahmenprogramm IF ermöglicht es somit, zum einen eine vertikale Integration von Applikationen aus unterschiedlichen Unternehmensebenen zu erreichen und zum anderen ermöglicht das Rahmenprogramm IF eine horizontale Integration von Applikationen der MES-Ebene.
- 35 Der Arbeitsplatz PIW1 stellt für einen Benutzer an der Frontendseite von MES-Applikationen oder anderen Applikationen aus

dem Unternehmen ein "One Window to the World" dar. Das heißt, der Arbeitsplatz ermöglicht, über eine gemeinsame einheitliche Oberfläche einen integrativen Zugang auf unterschiedliche, auch heterogene Anwendungen im Unternehmen. Der Benutzer des Arbeitsplatzes PIW1 kann somit von diesem einen Arbeitsplatz aus alle integrierten MES- oder anderen Anwendungen überwachen und verwalten. Dieser Arbeitsplatz kann über das Internet, das Intranet, LAN (Local Area Network) oder andere denkbare Verbindungen mit den Applikationen verbunden sein.

Es ist auch möglich, diesen Arbeitsplatz als mobile Station, z.B. als mobiles Endgerät (PDA, Handy) auszugestalten. Diese Mobilität würde für einen Benutzer noch weitere Vorteile bringen.

Die Darstellung gemäß FIG 3 zeigt die zentrale Stellung des die Softwareapplikationen koppelnden Rahmenprogramms. Um das erfindungsgemäße System oder Verfahren zu realisieren, bietet es sich an, eine Client-Server-Architektur zu wählen. Das Rahmenprogramm (IF; FIG 2) kann dabei auf einem einzigen Server oder auf mehreren beliebigen Servern, die sich in einer IT-Landschaft verteilen können, realisiert sein. In FIG 3 ist dargestellt, dass sich das Rahmenprogramm (IF; FIG 2) auf einem Server IFS (Industrial Framework Server) befindet. An diesem zentralen Server IFS hängen durch die bidirektionalen Informationspfade I2 - I8 verbunden, die Clients. Zu den Clients zählen zum einen die Applikationen aus der ERP-, der MES- und der Automatisierungsebene. In FIG 3 sind diese Applikationen am unteren Bildrand dargestellt. Über die Adapter AD4 - AD6 sind diese Applikationen mit dem Rahmenprogramm (IF; FIG 2) und somit mit dem Server IFS verbunden. Die Verbindung der Adapter AD4 - AD6 mit den Applikationen erfolgt über API-Schnittstellen API1 - API3 (API steht für Application Programmable Interface). Ein Application Programmable Interface stellt eine Schnittstelle mit einer Menge von Kommandos dar. APIs werden auch verwendet bei der Umsetzung von Parameterlisten von einem Format in ein anderes Format und bei der Interpretation der Argumente in eine oder beide Richtun-

gen. Die APIs sind sozusagen der Klebstoff zwischen den Applikationen und den Adaptern. Die Verbindung zwischen den Adaptern AD4 - AD6 mit dem Rahmenprogramm (IF; FIG 2) (in FIG 3 dargestellt durch die bidirektionalen Informationspfade I3 - I5) geschieht über geeignete Datenformate (z.B. XML), geeignete Protokolle (XOP, OPC, etc.) und geeignete Transportmechanismen (z.B. DCOM oder MSMQ). Auch HTTP (Hyper Text Transfer Protocol) kann hierbei verwendet werden. Auch das auf XML eXtensible Markup Language) beruhende Protokoll SOAP (Simple Object Access Protocol) kann für die Integration der Adapter AD4 - AD6 an das Rahmenprogramm (IF; FIG 2) bzw. den zugehörigen Server IFS verwendet werden. Clients bzw. Applikationen, die ActiveX-Dokumente bzw. -Aufrufe unterstützen, lassen sich besonders vorteilhaft in das Rahmenprogramm (IF; FIG 2), bzw. den Server IFS integrieren. Die Anbindung der Applikationen an das Rahmenprogramm kann neben Adaptern auch durch Wrapper oder anderen Integrationsmechanismen erfolgen.

Als weiterer Client kann mit dem Server IFS das Repository IFR (Industrial Framework Repository) verbunden sein. In FIG 3 ist die Verbindung durch den bidirektionalen Informationspfad I2 dargestellt. Das Repository IFR wird verwendet, um Daten sicher und persistent zu halten. Über Methodenaufrufe kann auf diese Daten zugegriffen werden. Im Repository sind u.a. Objekte, Methoden und Laufzeitdaten gespeichert.

Auf der oberen Bildhälfte sind weitere Clients des Servers IFS dargestellt. Der Personalized Industrial Workplace PIW2 und eine eventuell vorhandene Engineeringumgebung EU sind Clients des Servers IFS. Der Personalized Industrial Workplace PIW2 ist durch den bidirektionalen Informationspfad I6 mit dem Rahmenprogramm (IF; FIG 2) bzw. mit dem Server verbunden, die Engineeringumgebung EU entsprechend durch den bidirektionalen Informationspfad I7. Durch die drei Punkte ist dargestellt, dass weitere Clients am Server IFS hängen können. In FIG 3 ist angedeutet, dass außerdem ein weiterer Client C, verbunden durch den Informationspfad I8, am Server IFS hängt.

Die Verbindung der Clients IFR, PIW2, EU, C geschieht entsprechend über APIs bzw. über gängige Datenformate (z.B. XML), gängige Protokolle (XOP, OPC, etc.) und gängige Transportmechanismen (z.B. DCOM, HTTP oder MSMQ).

5

Die eingesetzten Adapter AD4 - AD6 ermöglichen den Zugang zu Daten und auch zu Methoden der einzelnen Applikationen, die sie mit dem Rahmenprogramm (IF; FIG 2) verbinden. Diese Adapter sind sehr flexibel und nicht auf einzelne spezielle Protokolle oder spezielle Transportmechanismen festgelegt. Wenn  
10 die Adapter in einer Laufzeitumgebung eingesetzt werden, dann sind sie so konfiguriert, dass sichergestellt ist, dass bestimmte benötigte Daten aus einer Applikation zum richtigen Zeitpunkt in der Serverumgebung vorhanden sind. Dies kann, wie schon gesagt, über unterschiedliche Protokolle und Transportmechanismen erfolgen. In einer Laufzeitumgebung können  
15 sich mehrere Adapter, die auch kleine Servereigenschaften (wie beispielsweise das Ausführung von Workflows, die Bereitstellung verschiedener Kommunikationsmöglichkeiten, ...) besitzen können, befinden. Diese Adapter können auf dem jeweiligen Applikationsrechner laufen. Sie müssen aber nicht nur auf einer Maschine laufen, sie können auch verteilt sein.

Softwareapplikationen, insbesondere MES-Applikationen (Manufacturing Execution Systems), liegen oft in einer heterogenen Form vor, z.B. können sie unterschiedliche Datenformate oder Objektmodelle besitzen oder sie sind in unterschiedlichen Programmiersprachen realisiert. Das erfindungsgemäße System bzw. Verfahren ermöglicht es, solche heterogenen Applikationen mit Hilfe eines Rahmenprogramms zu integrieren. Die Kommunikation zwischen diesen Applikationen erfolgt auf der Basis von Kommunikationsmitteln wie z.B. HTTP, DCOM, MSMQ, etc. Die Kommunikation, d.h. der Datenaustausch zwischen den Applikationen ist aber unabhängig von diesen Kommunikationsmitteln, d.h. die Applikationen können von den Applikationsmitteln abstrahieren.  
25  
30  
35

Die Darstellung gemäß FIG 4 zeigt den prinzipiellen Aufbau eines Adapters. Jeder Adapter ist eine spezielle Component (die "Component" als Metaobjektmodell wird in FIG 11 genauer erläutert) mit der besonderen Eigenschaft, dass die Component eines Adapters jeweils in einem eigenen Prozess läuft. Jeder Adapter bringt schon eine bestimmte Default-Struktur mit, die wieder als Baumstruktur von Objekten des Rahmenprogramms, d.h. Components und Variablen, dargestellt ist, und die bestimmte Stellen zur Verfügung stellt, an denen der Adapter bestimmte Informationen nach außen legen kann. Beispiele dafür sind Statusinformationen über den Zustand des Adapters (ist der Adapter mit seiner Anwendung verbunden, läuft die Anwendung, Versionsinformationen, usw.). Weiterhin kann ein Adapter Informationen über das externe System, d.h. die externe Applikation, nach außen geben. Durch den "Object Space" kann ein Adapter Strukturen nach außen legen, auf die andere Adapter bzw. andere Applikationen zugreifen können. Auch kann ein Adapter Informationen bezüglich einer Kommunikationsinfrastruktur nach außen geben. Unter Kommunikationsinfrastruktur versteht man Objekte, um Nachrichten zu senden, zu empfangen, Nachrichten zu transformieren. Aber auch Mechanismen, um ein Routing vorzunehmen und Mechanismen, um den Datenaustausch des Adapters zu protokollieren. Weiterhin gehören zur Kommunikationsinfrastruktur eines Adapters so genannte Inports und Outports, die physikalisch die Nachrichten empfangen oder versenden. Ein Adapter ist als eigener Prozess des Betriebssystems vorhanden, d.h. wenn ein Adapter abstürzt, dann hat das keine Auswirkungen auf andere Adapter. Ein Adapter ist somit eine eigene geschlossene Anwendung, die alleine ausführbar ist

Adapter und Wrapper sind Mechanismen für die Integration von Softwarekomponenten in ein Softwaresystem. Ein Wrapper bildet das API (Application Programmable Interface) einer Fremdkomponente bzw. einer zu integrierenden Applikation in das Objektmodell des Softwaresystems, hier Rahmenprogramms (IF; FIG 2) ab. So wird z.B. aus einer Methode des API der zu in-

tegrierenden Applikation eine Methode des Rahmenprogramms bzw. aus einem Integer-Datentyp des API der zu integrierenden Applikation wird ein Integer-Datentyp des Rahmenprogramms, usw. Ein Wrapper bringt somit das API der zu integrierenden Applikation in die Sprache, in die Nomenklatur und in die Objektwelt des Rahmenprogramms.

Ein Adapter dagegen ist eine Abstraktionsstufe höher als ein Wrapper. Adapter stellen eine standardisierte Sicht auf zu integrierende Applikationen dar, und das Rahmenprogramm, in das die zu integrierende Applikation eingehängt wird, kann von dieser Applikation abstrahieren. Ein Adapter stellt Funktionalität zur Verfügung, um das zu integrierende System, d.h. die zu integrierende Applikation, zu starten, zu bedienen und zu handeln. Mit Hilfe der Adapter kann z.B. ein Anwender eine SAP-Applikation benutzen, ohne ein SAP-Experte zu sein. Durch die IDOC-Adapter werden SAP-Applikationen in das Objektmodell des Rahmenprogramms abgebildet und werden dann als Objekte des Rahmenprogramms (Component IDOC) verwendet.

Durch das erfindungsgemäße System und Verfahren können zwei Applikationen (z.B. MES-Applikationen) peer-to-peer-mäßig zusammengebracht werden, ohne dass eine solche Verbindung jeweils peer-to-peer-mäßig programmiert werden muss. Diese Verbindungen werden erfindungsgemäß projiziert durch das Etablieren einer Connection (siehe FIG 13).

Tracebox (TB; FIG 6) und Errorbox (EB; FIG 6) als Mechanismen zum Sammeln von Informationen, die den Adapter betreffen können bereits in der Defaultstruktur der Adapter vorhanden sein.

Die Darstellung gemäß FIG 5 zeigt mehrere verschaltete Adapter AD7 bis AD10. Die Adapter sind als Rechtecke gekennzeichnet, die Verschaltung wird durch Verbindungslinien dargestellt. Durch Adapter werden Softwareapplikationen, z.B. MES-Applikationen, miteinander integriert. Wenn nun kein koppeln-des Softwaresystem (z.B. ein Framework) vorhanden ist, in das

jeder Adapter integriert wird, dann steigt die Komplexizität und der Aufwand beim Integrieren der Adapter untereinander sehr stark an.

5 Die Darstellung gemäß FIG 6 zeigt eine funktionelle Prinzipdarstellung der Erfindung. Die Softwareapplikationen A4 bis A6 sind durch die Adapter AD11 bis AD13 jeweils mit dem kopelnden Softwaresystem SS verbunden bzw. in das Softwaresystem SS integriert. Der erfindungsgemäße Gegenstand ermöglicht  
10 die Verfolgung und Auswertung der Kommunikation im Softwaresystem SS und der im Softwaresystem integrierten Applikationen A4 bis A6 auf unterschiedlichen Ebenen.

Zum einen erfolgt die Verfolgung und Auswertung der Kommunikation auf Port-Ebene. Die logische Verbindungsstelle zwischen einem Adapter und einer Connection (siehe FIG 12 und FIG 13) ist ein so genannter Port. Ein Port ist die Stelle,  
15 an der Informationen bzw. Nachrichten in einen Adapter rein-gehen bzw. einen Adapter verlassen.

20 Die zweite Stufe der Verfolgung und Auswertung der Kommunikation geschieht auf Adapterebene. An einem Adapter können mehrere Ports und Connections hängen.

25 Die dritte Stufe der Verfolgung und Auswertung der Kommunikation geschieht projektweit. Ein Projekt, z.B. ein MES-Projekt, beinhaltet mehrere Applikationen bzw. MES-Applikationen. Bei der Verfolgung und Auswertung der Kommunikation auf Projektebene wird der gesamte Nachrichtenaustausch aller  
30 beim Projekt beteiligten Applikationen erfasst.

Auf jeder dieser Ebenen gibt es Traceboxen TB und Errorboxen EB. Eine Tracebox TB dient zum Sammeln von durchlaufenden Nachrichten, eine Errorbox EB dient zum Sammeln von Nachrichten,  
35 die nicht zugestellt werden konnten oder bei deren Weiterleitung Fehler auftraten. Über die Informationspfade I9 bzw. I10 gelangen diese Nachrichten in die Tracebox TP bzw.

die Errorbox EB. Über die Informationspfade I11 bzw. I12 werden die von der Tracebox TP bzw. der Errorbox EB gesammelten Nachrichten im Repository DB abgelegt. Ein Anwender kann auf die im Repository DB hinterlegte Information dediziert über die Definition von Filterbedingungen zugreifen. Mögliche Filterbedingungen sind z.B. ein Anwender will nur die Nachrichten von bestimmten Absendern oder bestimmten Empfängern sehen. Es ist aber auch möglich, sich nur die Nachrichten eines bestimmten Zeitraums darstellen zu lassen. Dies gilt für Nachrichten aus der Tracebox TP und aus der Errorbox EB.

Die Daten aus dem Repository DB können aber auch zu Auswert- und Diagnosezwecken verwendet werden. In FIG 6 ist dargestellt, dass eine Auswerteeinrichtung AE über den Informationspfad I13 mit Daten versorgt wird. Über diese Auswerteeinrichtung AE können nun Auswertungen, Statistiken, Verdichtungen der Daten, aber auch Reports erstellt werden. Solche Reports sind insbesondere bei Nachweispflichten gegenüber Behörden oder Dienststellen sehr hilfreich, damit dann nachgewiesen werden kann, dass nur bestimmte Informationen geflossen sind und bei einem Fehlverhalten des Systems oder Anlage ein Eigenverschulden nachweislich auszuschließen ist. Dies ist insbesondere im pharmazeutischen Bereich von Interesse. („FDA-Nachweislisten“). Über die Auswertungen können aber auch z.B. Flaschenhälse erkannt werden und bestimmte Optimierungen vorgenommen werden. Diese Erkenntnisse können von der Auswerteeinrichtung AE direkt wieder ins Softwaresystem SS einfließen. In FIG 6 ist dies durch den Informationspfad I14 dargestellt. Damit kann das erfindungsgemäße System bzw. das Verfahren im weitesten Sinne auch als ein feldregulierender Regelkreis angesehen werden.

Die Darstellung gemäß FIG 7 zeigt ein Instanzendiagramm zu einem Projekt, z.B. einem MES-Projekt. Die Verfolgung und Auswertung der Kommunikation von Applikationen, insbesondere MES-Applikationen, kann auf Projektebene, auf Adapterebene oder auf Portebene erfolgen. Deshalb gibt es jeweils eine

projektweite Trace- und Errorbox, für jeden Adapter gibt es jeweils eine adapterweite Trace- und Errorbox und für jeden Port gibt es auch jeweils eine portweite Trace- und Errorbox. Die Traceboxen dienen zum Sammeln von durchlaufenden Nachrichten, die Errorboxen dienen zum Sammeln von Nachrichten, die nicht zugestellt werden konnten oder bei deren Weiterleitung Fehler auftraten. Für die Erstellung eines Projektes können mehrere Applikationen verwendet werden. Jede Applikation ist üblicherweise über einen Adapter in das Projekt integriert. Jeder Adapter besitzt eine eigene adapterweite Tracebox bzw. eine eigene adapterweite Errorbox. Auch jeder Port besitzt eine eigene portweite Tracebox bzw. eine eigene portweite Errorbox. Aufgabe eines Ports ist es, als Interceptor zu agieren. Ein Port ist Teil eines zugrunde liegenden Kommunikationssystems und nicht Teil eines Adapters bzw. der adaptierten Anwendung.

Ein Port kann mehrere Connections (siehe FIG 12 und FIG 13) haben und ein Adapter kann mehrere Ports besitzen. Ein Port ist die Stelle, an der logisch die Daten über die Connection in einen Adapter rein- bzw. rausgehen.

Ports hängen z.B. an Variablen, die sie mit Informationen bzw. Werten versorgen, in FIG 7 hängt der eingezeichnete Port an der Variable Produktionsauftrag. Diese Variable Produktionsauftrag gehört zum Adapter 1. Die Traceboxen und Errorboxen für Adapter können bereits in der Default-Struktur eines Adapters vorhanden sein.

In FIG 7 ist durch die Strichelung der vertikalen Linien an ihrem Ende dargestellt, dass ein Projekt ein Adapter bzw. ein Port außer den eingezeichneten Objekten weitere Objekte oder Elemente enthalten kann.

Die Darstellung gemäß FIG 8 zeigt ein Sequence Chart für die Initialisierung des Verfolge- und Auswertemodus.

Die Darstellung gemäß FIG 9 zeigt ein Sequence Chart für den Betrieb des Verfolgemodus.

Die Darstellung gemäß FIG 10 zeigt ein Sequence Chart für den Betrieb des Auswertemodus.

Die Darstellung gemäß FIG 11 zeigt die Objektstruktur einer "Component" als Metaobjektmodell des Rahmenprogramms (IF; FIG 2) in UML (Unified Modelling Language). Beim erfindungsgemäßen System und Verfahren werden die zu integrierenden Softwareapplikationen und die zugrunde liegenden Kommunikationsmittel (z.B. HTTP, MSMQ, etc.) auf ein Objektmodell des Rahmenprogramms (IF; FIG 2) abgebildet. Dadurch besitzen an sich heterogene Applikationen ein gemeinsames Objektmodell. Dadurch werden alle Methoden, Datenstrukturen, Objekte usw. der zu integrierenden Fremdapplikationen in einem gemeinsamen Objektmodell dargestellt. Dieses Objektmodell ist sehr generisch und stellt ein Metaobjektmodell dar. Der Kern dieses Objektmodells besteht aus einem Objekttyp namens "Component". Eine Component kann wiederum andere Components aggregieren und/oder spezielle Typen von Components, so genannte Variablen, denen im Betrieb bestimmte Werte zugewiesen sind, enthalten. Components und Variablen können jeweils auch zusätzliche Attribute besitzen.

25

Dieses Objektmodell bildet die Basis der Adaptierung von MES-Applikationen oder anderen Applikationen. Das bedeutet, dass die Strukturen der Applikationen in Strukturen dieses Objektmodells übersetzt bzw. abgebildet werden. Alle zu integrierenden Applikationen werden innerhalb des Rahmenprogramms (IF, FIG 2) in der Darstellung dieses Objektmodells repräsentiert. Auch die zugrunde liegenden Kommunikationsmittel werden auf dieses generische Objektmodell abgebildet. Im Rahmenprogramm (IF; FIG 2) sind nun alle Applikationen und alle zugrunde liegenden Kommunikationsmittel in einem einheitlichen und homogenen Objektmodell repräsentiert. Dadurch können

sehr einfach und leicht Kommunikationsbeziehungen zwischen den Applikationen eingerichtet werden.

5 In der Darstellung gemäß FIG 11 ist die generische Komponente "Component", die den Kern des Objektmodells darstellt, in UML-Notation aufgezeigt.

Die rechteckigen Kästchen stellen Teile des Objektmodells dar. Durch eine Rautenbeziehung wird eine Aggregation (ist Teil von-Beziehung) dargestellt, durch eine Dreiecksbeziehung wird die Vererbung (ist ein-Beziehung) dargestellt. In der Darstellung gemäß FIG 11 wird somit gezeigt, dass eine Component aus mehreren Components bestehen kann, weiterhin kann eine Component Teil einer anderen Component sein. Durch die Rautenbeziehung ist eine Component mit Attributen und Variablen verbunden. Das heißt, eine Component kann Attribute und Variable beinhalten. Attribute sind beschreibende Daten. Alle Objekte einer Klasse besitzen dieselben Attribute, jedoch im Allgemeinen unterschiedliche Attributwerte. Ein Attributwert ist ein einem Attribut zugeordneter Wert aus seinem Wertebereich. Eine Variable kann Werte von bestimmten Datentypen (z.B. Integer, Real etc.) annehmen. Wie durch die Rautenbeziehung dargestellt, kann eine Component mehrere Variablen enthalten. Eine Component kann aber auch eine Oberklasse einer Variable sein, wie durch die Dreiecksbeziehung dargestellt. Das heißt, eine Variable kann eine abgeleitete Component sein. Die Rauten- und die Dreiecksbeziehungen können auch Kardinalitäten beinhalten.

30 Durch Abbildungsmechanismen wie z.B. Adapter oder Wrapper werden die zu integrierenden Softwareapplikationen und die zugrunde liegenden Kommunikationsmittel auf diese generische Objektstruktur, d.h. "Component"-Struktur, des Rahmenprogramms (IF; FIG 2) abgebildet.

35

In der Darstellung gemäß FIG 12 wird dargestellt, wie eine Kommunikationsbeziehung zwischen MES-Applikation eingerichtet

wird. Wie schon oben erklärt, werden die Applikationen und die zugrunde liegenden Kommunikationsmechanismen auf eine einheitliche "Component"-Struktur (Metaobjektmodell des Rahmenprogramms) abgebildet.

5

In FIG 12 ist dargestellt, wie eine Kommunikationsverbindung zwischen den MES-Applikationen MES' und MES'' eingerichtet wird. Über Adapter bzw. über Wrapper werden diese MES-Applikationen MES' und MES'' auf die "Component"-Struktur des Rahmenprogramms (IF; FIG 2) abgebildet. Auch die zugrunde liegenden Kommunikationsmittel (HTTP, MSMQ, DCOM, etc.) werden über Adapter oder Wrapper auf das generische Objektmodell einer "Component"-Struktur abgebildet. Im Rahmenprogramm (IF; FIG 2) werden dann diese Kommunikationsmittel durch eine Component Transportsystem (TS) repräsentiert. Diese Component Transportsystem (TS) abstrahiert von dem zugrunde liegenden Kommunikationsmitteln. Bei der Einrichtung einer Kommunikationsverbindung ist es somit egal, welches Kommunikationsmittel letztendlich physikalisch zugrunde liegt. Die Kommunikationsmittel sind somit grundsätzlich austauschbar.

Die Kommunikationsbeziehung zwischen den MES-Applikationen MES' und MES'' wird durch eine so genannte "Connection" (s. FIG 13) hergestellt. Eine Connection verbindet zwei beliebige Objekte des Objektmodells. Da sich die Connection auf Objekte des generischen Objektmodells bezieht, kennt sie deren Semantik und weiß, welche Besonderheiten beim Verbinden herzustellen sind, d.h. Methodenobjekte sind anders zu handhaben als z.B. reine Datenobjekte. Weiterhin stützt sich die Connection auf ein Transportsystem ab. Auch das Transportsystem liegt als Teil des generischen Objektmodells des Rahmenprogramms (IF; FIG 2) vor. Eine Connection kann unidirektional oder bidirektional sein.

Die Darstellung gemäß FIG 13 zeigt die Objektstruktur einer "Connection" in UML-Notation. Eine Connection ist ein generisches Objekt, das im Wesentlichen aus zwei Teilen besteht.

Zum einen aus den Wrappern des spezifischen Kommunikationsmittels und andererseits aus einer Sammlung von Properties, d.h. Eigenschaften, die über dieses Kommunikationsmittel eingestellt werden können. Diese Properties und die Connection selbst sind wiederum "Components" des Objektmodells, wobei die Properties einfach durch eine Liste von Variablen gegeben sind. Ein Transport-Wrapper enthält auch wiederum Properties, d.h. Eigenschaften. Im Falle der Verwendung von MSMQ als Kommunikationsmittel ist z.B. der Name der verwendeten Message Queue eine solche Eigenschaft. Durch die Kardinalität (1:2) wird angedeutet, dass die Connection aus FIG 13 mindestens einen, aber höchstens zwei Transport-Wrappers beinhaltet. Auch die anderen Aggregationsbeziehungen (Rautensymbol), wie sie in FIG 13 dargestellt sind, können Kardinalitäten beinhalten. Ein Transport-Wrapper kann auch Methoden beinhalten. Zum Beispiel eine Methode zum Öffnen eines spezifischen Transportkanals, eine Methode zum Schließen eines spezifischen Transportkanals, eine Methode, um einen String zu senden, eine Methode, um einen String zu empfangen, usw.

Kommunikationsmittel werden üblicherweise durch Wrappers auf das Objektmodell des Rahmenprogramms abgebildet, prinzipiell ist es aber auch möglich, Kommunikationsmittel durch Adapter im Rahmenprogramm zu integrieren.

Zusammenfassend betrifft die Erfindung ein System und ein Verfahren zur Verfolgung und Auswertung der Kommunikation von Softwareanwendungen, insbesondere MES-Anwendungen in einem Gesamtsystem. Es wird eine skalierte bzw. stufenweise Verfolgung und Auswertung ermöglicht: Projektebene, Adapterebene und auf Portebene, d.h. einzelne Kommunikationsverbindungen, einzelne Anwendungen, aber auch das Gesamtprojekt wird über (auch heterogene) Anwendungslandschaften erfasst. Die Erfassungs- und Auswertungsmechanismen erlauben höherwertigen Diensten online jederzeit Zugriff auf die archivierten Trace- und Fehlerdaten. Ein Beispiel solch eines Dienstes wäre die Generierung eines Reports, der pharmazeutischen Bereich die

verwendeten Materialien nachweist, die innerhalb von Produktionsaufträgen verschickt wurden. Der online-Zugriff kann Datenverbindungen (z.B. über ein Local Area Network) oder über internetartige Mittel erfolgen.

5

Das oben beschriebene erfindungsgemäße System bzw. Verfahren lässt sich als Computerprogramm in dafür bekannten Sprachen implementieren. Ein derartig implementiertes Computerprogramm kann in ebenfalls bekannter Weise über elektronische Datenwege, aber auch auf Datenträgern (Computerprogrammprodukte) abgespeichert und transportiert werden.

10

## Patentansprüche

1. System zum Verfolgen und/oder Auswerten des Informationsaustausches zwischen insbesondere heterogenen Softwareapplikationen (A1 - A6), insbesondere MES-Applikationen, in einem die Softwareapplikationen koppelnden Softwaresystem (SS), wobei die Softwareapplikationen und das Softwaresystem auf mindestens einer Rechneinheit gespeichert sind,  
d a d u r c h g e k e n n z e i c h n e t, dass
- 5 a) die Softwareapplikationen (A1 - A6) durch Koppelmechanismen (AD1 - AD9) mit dem Softwaresystem (SS) verbunden sind,
- b) eine erste Einrichtung (TB) den Informationsaustausch zwischen den Softwareapplikationen erfasst,
- 15 c) eine zweite Einrichtung (EB) auftretende Fehler des Informationsaustausches erfasst,
- d) eine dritte Einrichtung (DB) den Informationsaustausch und die Fehler abspeichert und
- e) eine vierte Einrichtung (AE) den Informationsaustausch
- 20 und/oder die Fehler analysiert.
2. System nach Anspruch 1,  
d a d u r c h g e k e n n z e i c h n e t, dass die Ergebnisse der Analyse des Informationsaustausches und/oder der
- 25 Analyse der Fehler vom Softwaresystem (SS) direkt weiterverarbeitbar sind.
3. System nach Anspruch 1 oder Anspruch 2,  
d a d u r c h g e k e n n z e i c h n e t, dass durch eine
- 30 Filterbildung bestimmte Informationen und/oder Fehler auswählbar sind.
4. System nach Anspruch 1, 2 oder 3,  
d a d u r c h g e k e n n z e i c h n e t, dass die Verfolgung und/oder Auswertung auf Projektebene oder auf Adapter-
- 35 ebene oder auf Ebene eines Informationskanals erfolgt.

5. System nach einem der vorstehenden Ansprüche,  
dadurch gekennzeichnet, dass als Koppelmechanismen Adapter (AD1 - AD9) und/oder Wrapper eingesetzt sind.
- 5 6. System nach einem der vorstehenden Ansprüche,  
dadurch gekennzeichnet, dass die erste und zweite Einrichtung durch eine Einrichtung realisiert sind oder die erste, zweite und vierte Einrichtung durch eine Einrichtung realisiert sind oder die erste und vierte Einrichtung durch eine Einrichtung realisiert sind oder die zweite und vierte Einrichtung durch eine Einrichtung realisiert sind.
- 10 7. System nach einem der vorstehenden Ansprüche,  
dadurch gekennzeichnet, dass die dritte Einrichtung (DB) in eine der anderen Einrichtungen integriert ist.
- 15 8. Verfahren zum Verfolgen und/oder Auswerten des Informationsaustausches zwischen insbesondere heterogenen Softwareapplikationen (A1 - A6), insbesondere MES-Applikationen, in einem die Softwareapplikationen koppelnden Softwaresystem (SS), wobei die Softwareapplikationen und das Softwaresystem auf
- 20 mindestens einer Rechneinheit gespeichert werden,  
dadurch gekennzeichnet, dass
- 25 a) die Softwareapplikationen (A1 - A6) durch Koppelmechanismen (AD1 - AD9) mit dem Softwaresystem (SS) verbunden werden,
- 30 b) eine erste Einrichtung (TB) den Informationsaustausch zwischen den Softwareapplikationen (A1 - A6) erfasst,
- c) eine zweite Einrichtung (EB) auftretende Fehler des Informationsaustausches erfasst,
- d) eine dritte Einrichtung (DB) den Informationsaustausch und
- 35 die Fehler abspeichert und
- e) durch eine vierte Einrichtung (AE) der Informationsaustausch und/oder die Fehler analysiert werden.

9. Verfahren nach Anspruch 8,  
d a d u r c h g e k e n n z e i c h n e t, dass die Ergebnisse der Analyse des Informationsaustausches und/oder der Analyse der Fehler vom Softwaresystem (SS) direkt weiterver-  
5 arbeitet werden.
10. Verfahren nach Anspruch 8 oder 9,  
d a d u r c h g e k e n n z e i c h n e t, dass durch eine Filterbildung bestimmte Informationen und/oder Fehler ausge-  
10 wählt werden.
11. Verfahren nach einem der Ansprüche 8 bis 10,  
d a d u r c h g e k e n n z e i c h n e t, dass die Verfolgung und/oder Auswertung auf Projektebene oder auf Adapter-  
15 ebene oder auf Ebene eines Informationskanals durchgeführt wird.
12. Verfahren nach einem der Ansprüche 8 bis 11,  
d a d u r c h g e k e n n z e i c h n e t, dass als Koppel-  
20 mechanismen Adapter (AD1 - AD9) und/oder Wrapper eingesetzt werden.
13. Verfahren nach einem der Ansprüche 8 bis 12,  
d a d u r c h g e k e n n z e i c h n e t, dass die erste  
25 und zweite Einrichtung durch eine Einrichtung realisiert werden oder die erste, zweite und vierte Einrichtung durch eine Einrichtung realisiert werden oder die erste und vierte Einrichtung durch eine Einrichtung realisiert werden oder die  
30 zweite und vierte Einrichtung durch eine Einrichtung realisiert werden.
14. Verfahren nach einem der Ansprüche 8 bis 13,  
d a d u r c h g e k e n n z e i c h n e t, dass die dritte  
Einrichtung (DB) in eine der anderen Einrichtungen integriert  
35 ist.

29

15. Computerprogramm, das ein Verfahren nach einem der Ansprüche 8 bis 14 implementiert.

16. Datenträger, auf dem ein Computerprogramm nach Anspruch  
5 15 gespeichert ist.

17. Datenverarbeitungseinrichtung (PIW1, PIW2, IFS), auf der  
ein Computerprogramm nach Anspruch 15 installiert ist.

10

1/10

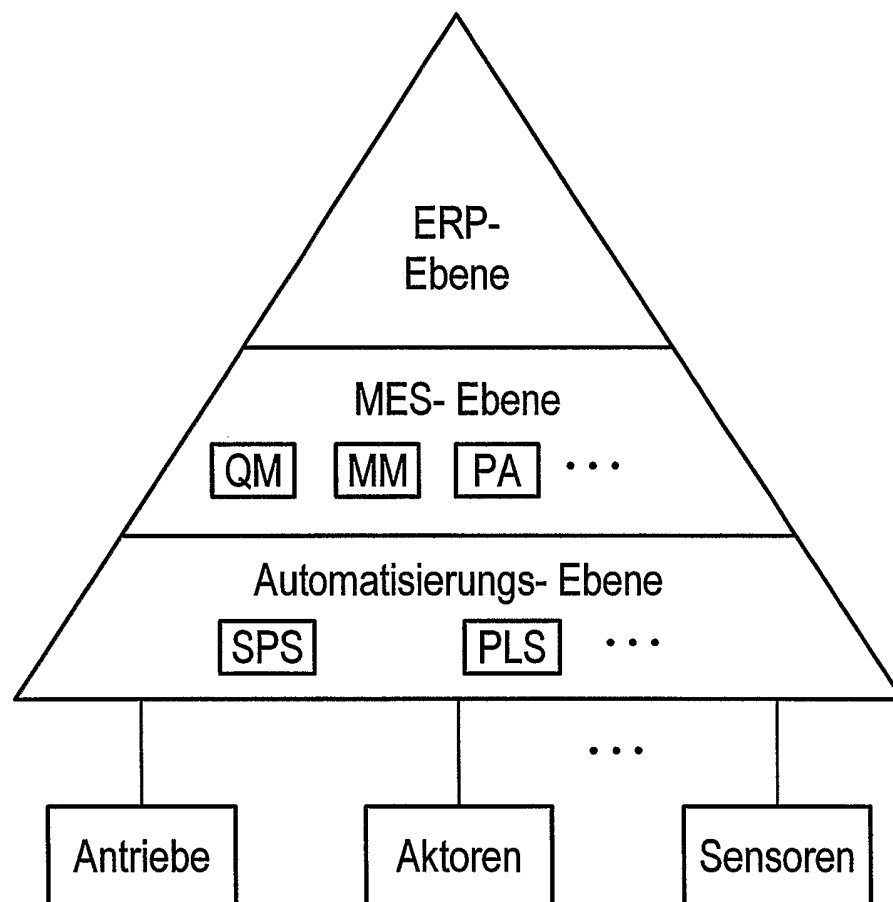


FIG 1

2/10

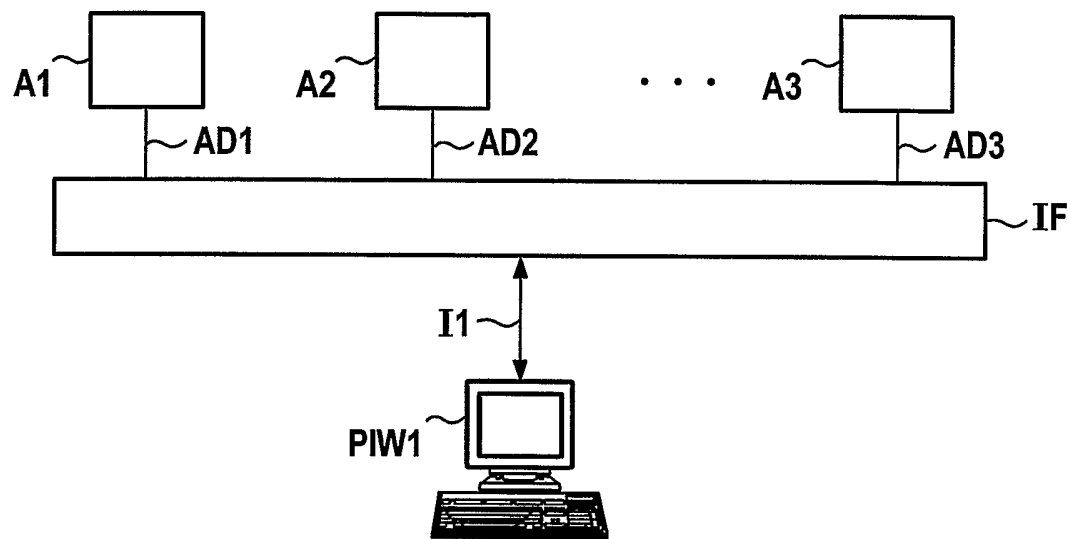


FIG 2

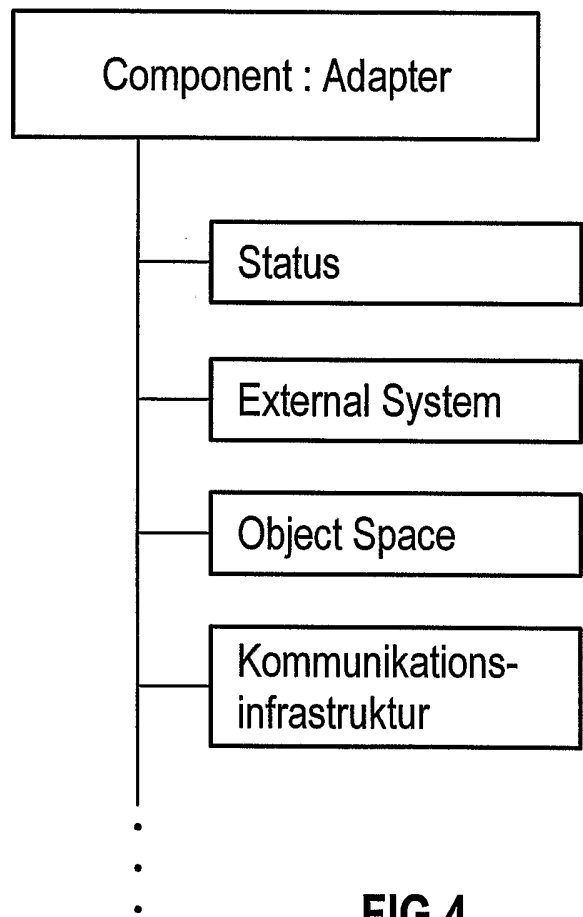


FIG 4

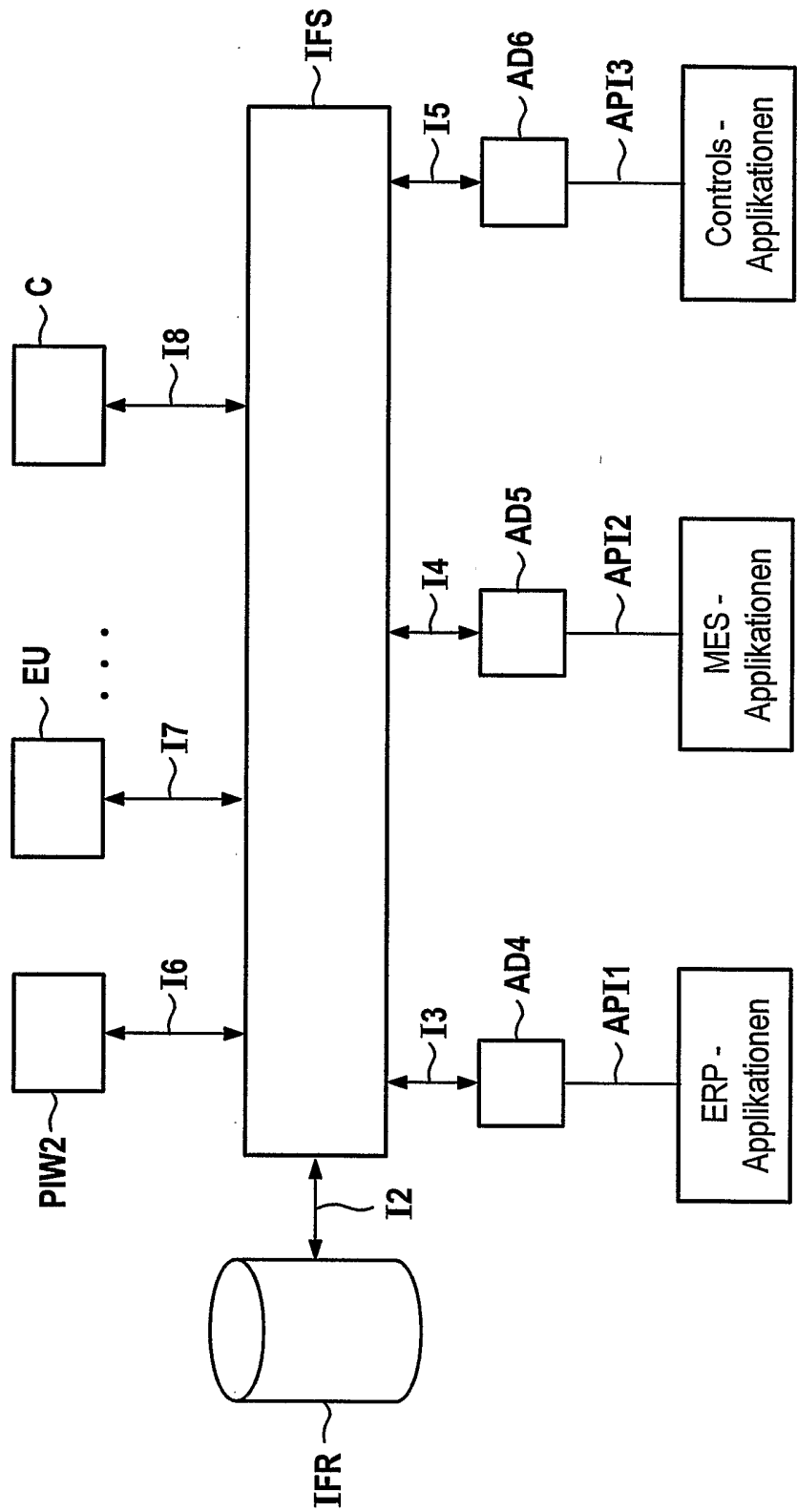
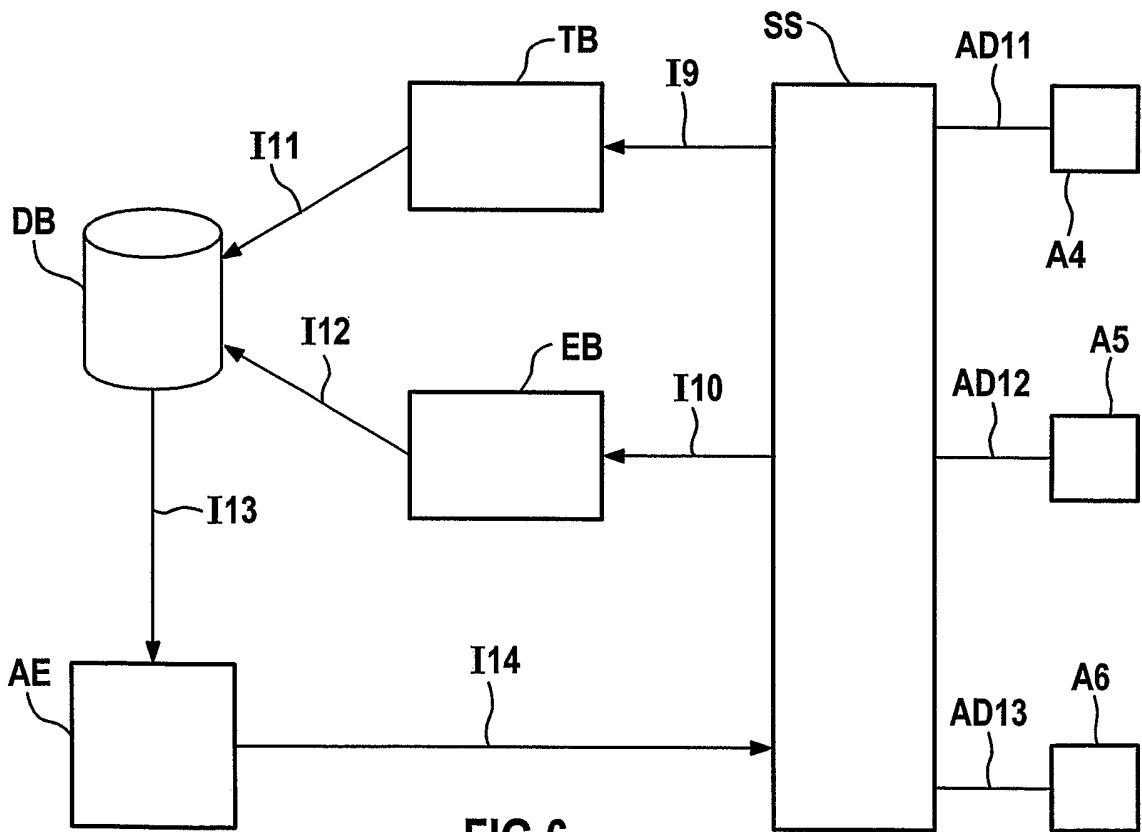
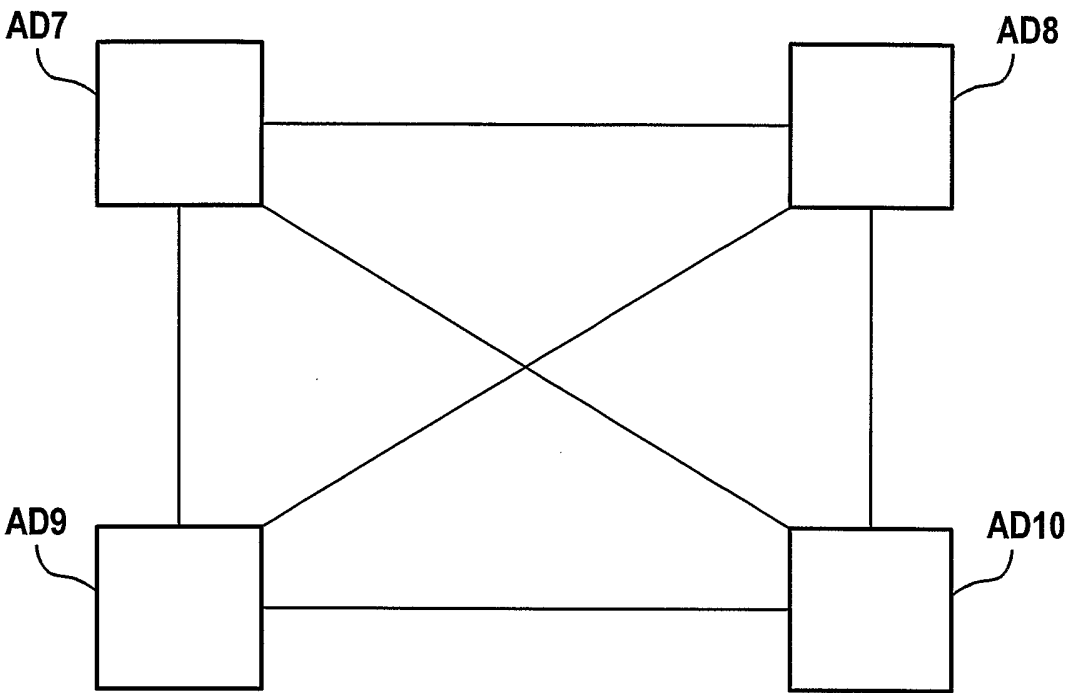


FIG 3

4/10



5/10

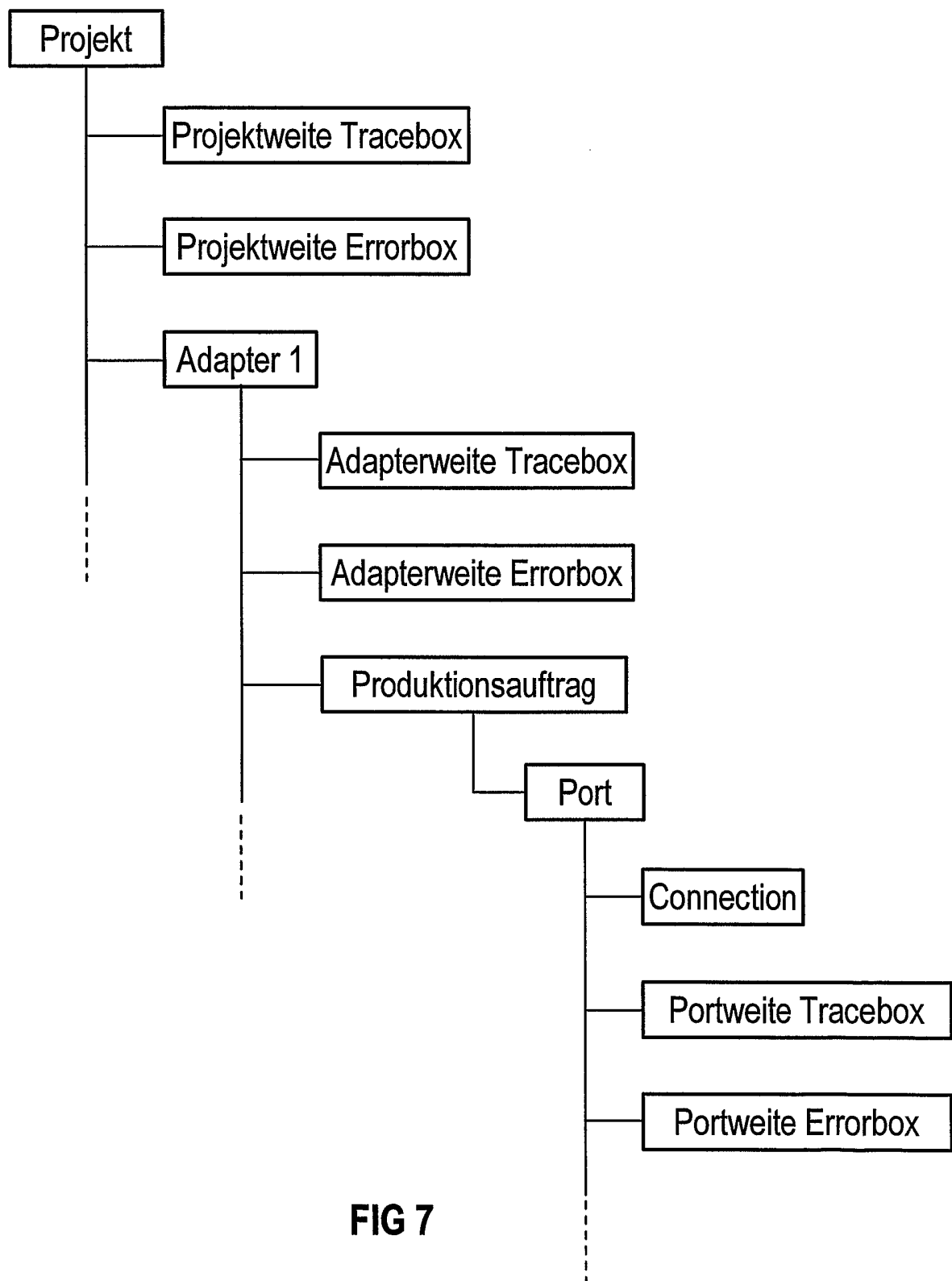


FIG 7

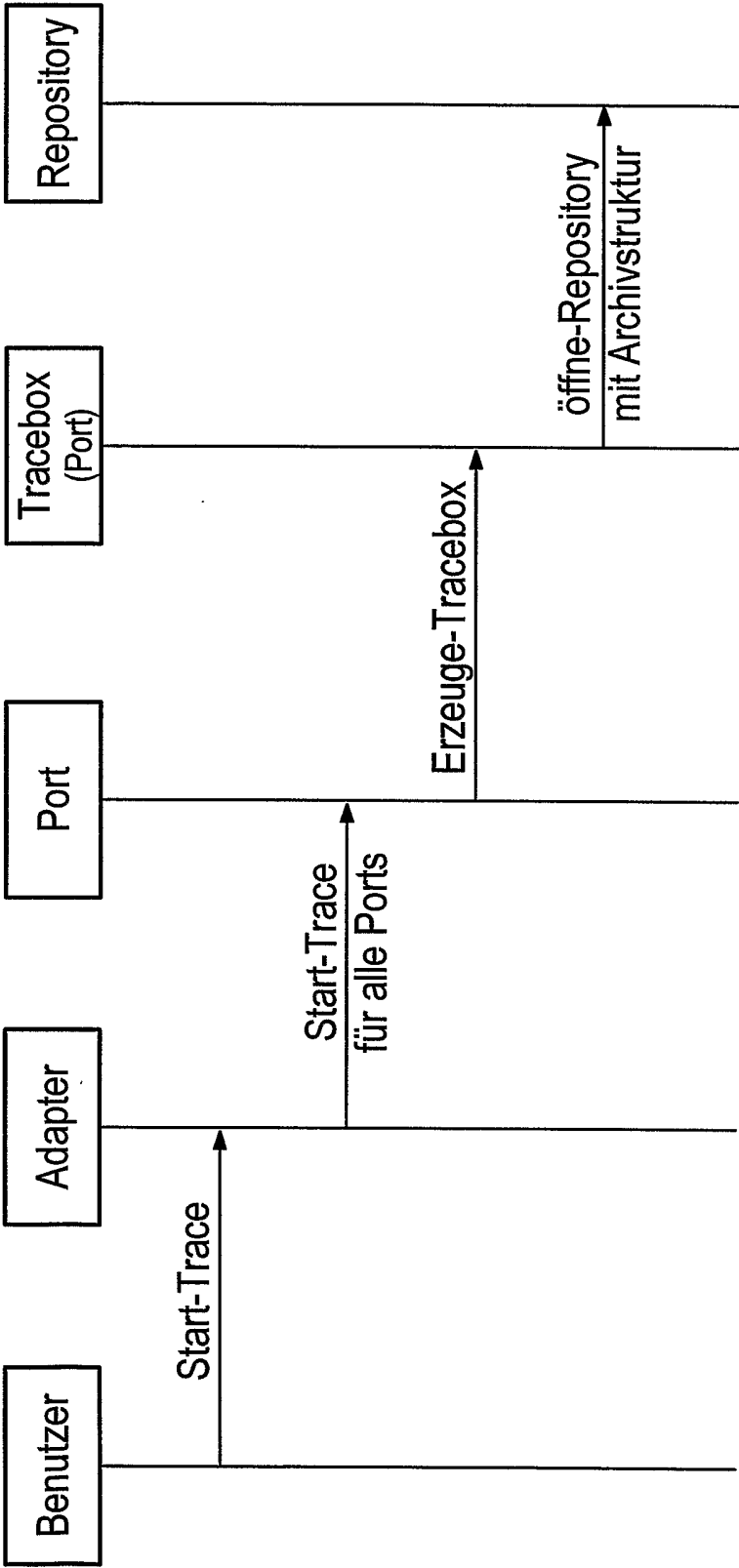


FIG 8

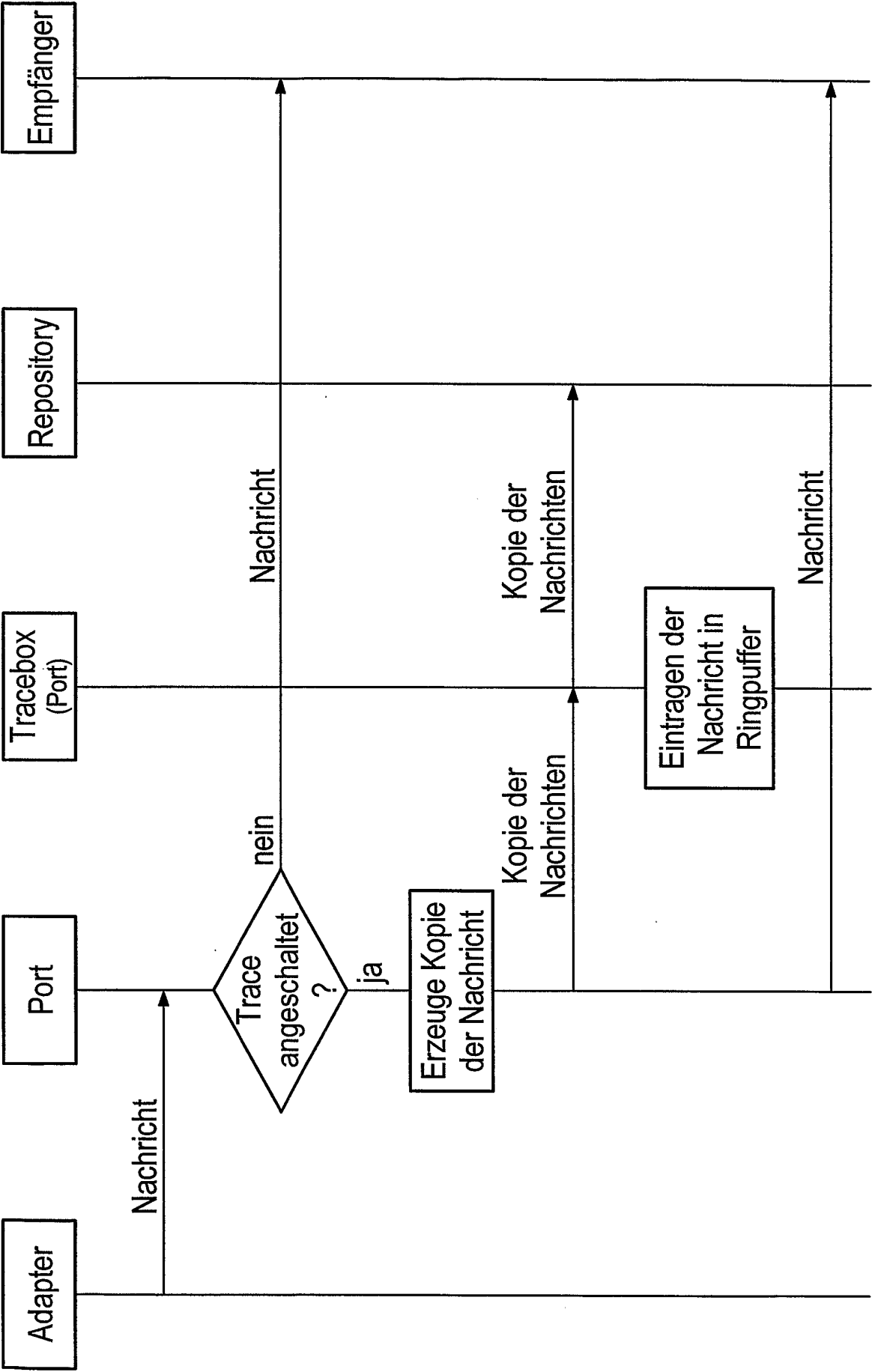


FIG 9

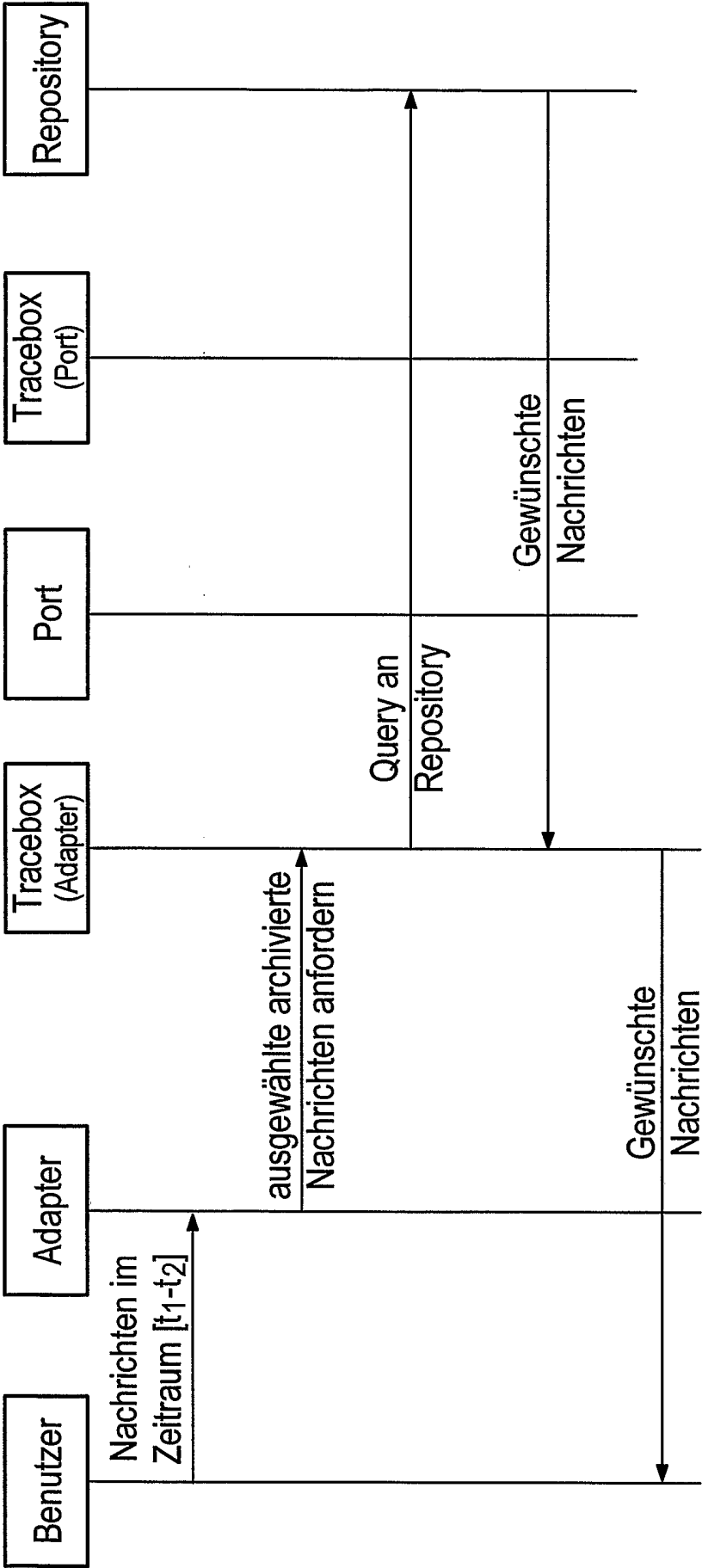


FIG 10

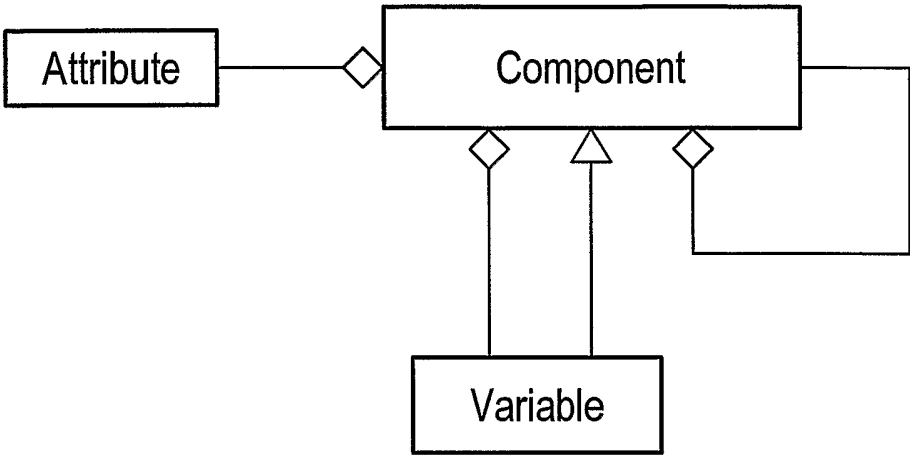


FIG 11

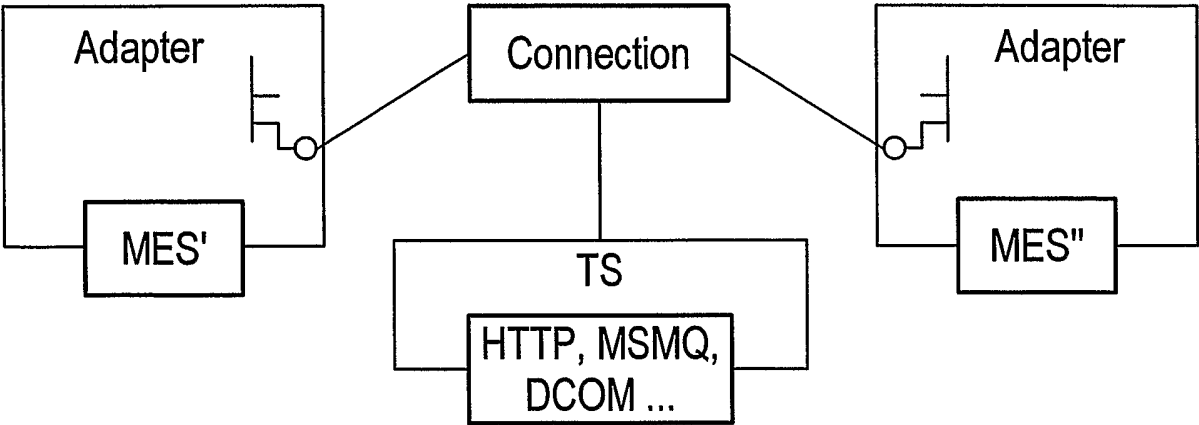


FIG 12

10/10

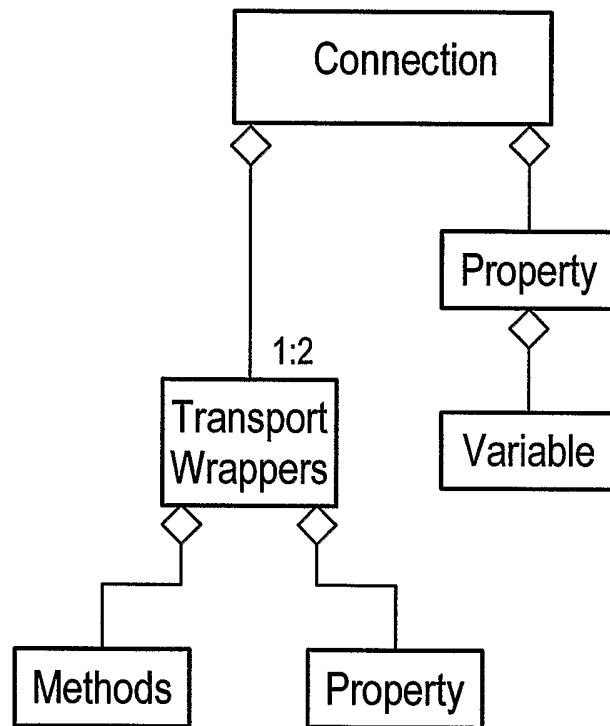


FIG 13