

(12) **Österreichische Patentanmeldung**

(21) Anmeldenummer: A 1825/2011
(22) Anmeldetag: 13.12.2011
(43) Veröffentlicht am: 15.03.2013

(51) Int. Cl. : **G06F 9/46** (2006.01)
H04L 29/08 (2006.01)

(56) Entgegenhaltungen:
US 2011119328 A1
US 7979247 B2

Kopetz, Hermann. "Real-time systems. Design principles for distributed embedded applications". Second Edition. Springer Science+Business Media, New York, USA, 2011. Seiten 104 bis 108.

(73) Patentanmelder:
FTS COMPUTERTECHNIK GMBH
1040 WIEN (AT)

(54) **VERFAHREN ZUR INTEGRATION VON EDV-SYSTEMEN IN EIN SYSTEM-OF-SYSTEMS**

(57) Die vorliegende Erfindung legt ein Verfahren offen, wie autonome EDV Systeme (120,151, 152,153), die in unterschiedlichen Organisationen betrieben werden, dynamisch in ein System-of- Systems (SoS) integriert werden können. Durch die Einführung einer globalen Zeit und die Angabe von Gültigkeitszeiten in der Schnittstellenspezifikation eines einen Dienst anbietenden EDV Systems kann das Problem der unkoordinierten Evolution der autonomen EDV Systeme, die ein SoS bilden, gelöst werden.

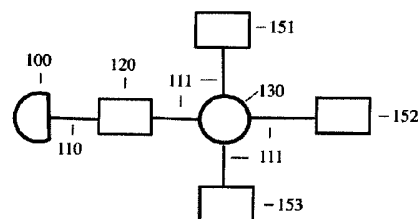


FIG. 1

ZUSAMMENFASSUNG

Die vorliegende Erfindung legt ein Verfahren offen, wie autonome EDV Systeme, die in unterschiedlichen Organisationen betrieben werden, dynamisch in ein System-of-Systems (SoS) integriert werden können. Durch die Einführung einer globalen Zeit und die Angabe einer Gültigkeitszeit in der Schnittstellenspezifikation eines einen Dienst anbietenden EDV Systems kann das Problem der unkoordinierten Evolution der autonomen EDV Systeme, die ein SoS bilden, gelöst werden.

Verfahren zur Integration von EDV Systemen in ein *System-of-Systems* (SoS)

Zitierte Literatur

Patente:

- [1] US 7,979,247 Butterfield, et al.. *System, method and computer program product for developing a system-of-systems architecture model*. Granted July 12, 2011
- [2] US 7,797,141 Kinsella et al.. *Remote component and connection architecture*. Granted June 8, 2010.

Sonstige:

- [3] Sycara, C. et al.. *Automated Discovery, Interaction and Composition of Semantic Web Services*, Web-Semantic Journal., retrieved Dec.7, 2011 from URL: test.websemanticsjournal.org . pp. 1-27.
- [4] Kopetz, H. *Real-Time Systems, Design Principles for Distributed Embedded Applications*. Springer Verlag. 2011.
- [5] Newell, A.. *Human Problem Solving*. Longman Higher Education.1972.

Technisches Umfeld

Die vorliegende Erfindung liegt im Bereich der Computertechnik. Sie beschreibt ein innovatives Verfahren zur dynamischen Integration von autonomen EDV Systemen in ein System-of-Systems (SoS).

Kurze Beschreibung der Erfindung

Durch die Integration bestehender EDV Systeme (*legacy systems*) in ein *System-of-Systems* (SoS) können neue Dienste angeboten und wirtschaftliche Vorteile realisiert werden. Kennzeichnend für ein solches SoS ist, dass die einzelnen EDV Systeme, die dynamisch in ein SoS integriert werden, von unabhängigen Organisationen betrieben werden, die ihre EDV Systeme laufend den sich ändernden lokalen Anforderungen anpassen müssen. Es ist deshalb schwierig in einem großen SoS ein konsistentes Verhalten aller beteiligten EDV Systeme zu garantieren.

Die Methode ein SoS zu entwickeln, die in [1] offengelegt wird, geht auf diese kontinuierliche unkoordinierte Evolution der einzelnen EDV Systemen eines SoS nicht ein. Vielmehr wird in diesem Ansatz ein einziges hierarchisches Architekturmodell vorgeschlagen in dem alle Änderungen konsistent gehalten werden

können. Dies setzt voraus, dass alle EDV Systeme, die in das SoS integriert werden, von einer einzigen Instanz verwaltet werden, was in vielen SoSs nicht der Fall ist.

Das Patent *Remote Component and Connection Architecture* [2] legt eine Methode offen, wie eine dynamische Verkettung von Software Komponenten realisiert werden kann. Das Problem einer möglichen Inkonsistenz der Softwarekomponenten, die sich durch die notwendigen lokalen Änderungen ergeben, wird in [2] nicht behandelt.

Die Integration der einzelnen EDV Systeme in ein SoS erfolgt über die Bereitstellung und den Aufruf von Diensten an einer gemeinsamen Schnittstelle zwischen einem einen Dienst anbietenden EDV System (*service providing System--SPS*) und einem einen Dienst anfordernden EDV System (*service requesting System--SRS*). Das SPS spezifiziert in dieser gemeinsamen Schnittstelle den Dienst den es zu leisten in der Lage ist, wie z.B. in [3] ausgeführt. Das SRS überprüft auf der Basis dieser Schnittstellenspezifikation ob der angebotene Dienst die Anforderungen des SRS erfüllt und, falls dies der Fall ist, aktiviert mittels eines entsprechenden Aufrufs den Dienst. Da Änderungen in dem Dienst des SPS jederzeit vorgenommen werden können, kann zwischen dem Zeitpunkt der Überprüfung des Dienstes und dem Zeitpunkt des Aufrufs des Dienstes eine solche Änderung stattfinden. In diesem Fall wird das SRS eine Antwort erhalten, die falsch sein kann.

In der vorliegenden Erfindung wird dieses Problem dadurch gelöst, dass alle beteiligten EDV System Zugriff auf eine gemeinsame Zeit (wie in [4] angeführt) haben und jedes SPS in der gemeinsamen Schnittstelle mindestens ein *Gültigkeitszeitfeld* verwaltet das angibt, bis zu welchem zukünftigen Zeitpunkt die angebotene Schnittstellenspezifikation und die darin enthaltenen Informationen gültig sein werden. Das SPS geht die Verpflichtung ein, vor diesem im *Gültigkeitszeitfeld* enthaltenen Zeitpunkt weder die Syntax noch die Semantik des angebotenen Dienstes, noch den Inhalt der dargebotenen Informationen zu ändern. Gegebenenfalls können zwei unterschiedliche *Gültigkeitszeitfelder* eingeführt werden, wobei ein *Gültigkeitszeitfeld* die Gültigkeit der Syntax und Semantik der Schnittstellenspezifikation und weiteres *Gültigkeitszeitfeld* die Gültigkeit der dargebotenen Information beschränkt. In einem statisches System, in dem weder die Syntax noch die Semantik des angebotenen Dienstes verändert werden, genügt ein einziges *Gültigkeitszeitfeld*, das die Gültigkeit der dargebotenen Information beschränkt. Ein weiteres Zeitfeld (*Timeout_I*) in der gemeinsamen Schnittstelle gibt an wie lange die Abarbeitung eines Dienstes durch das SPS dauert. Ein SRS kann nun vor Inanspruchnahme des Dienstes überprüfen, ob der Dienst vor Ende des Gültigkeitszeit der Schnittstellenspezifikation abgeschlossen sein wird und kann bei Aufruf eines Dienstes ein lokales *Timeout_I* im SRS setzen, um einen Ausfall eines Dienstes sofort erkennen zu können.

Zusammenfassung

Die vorliegende Erfindung legt ein Verfahren offen, wie autonome EDV Systeme, die in unterschiedlichen Organisationen betrieben werden, dynamisch in ein System-of-Systems (SoS) integriert werden können. Durch die Einführung einer globalen Zeit und die Angabe von Gültigkeitszeiten in der Schnittstellenspezifikation eines einen Dienst anbietenden EDV Systems kann das Problem der unkoordinierten Evolution der autonomen EDV Systeme, die ein SoS bilden, gelöst werden.

Kurze Beschreibung der Zeichnungen

Die vorliegende Erfindung wird an Hand der folgenden Zeichnungen erklärt:

Fig. 1 zeigt den Aufbau eines SoS bestehend aus vier unabhängigen EDV Systemen und ein sie verbindendes Kommunikationssystem.

Fig. 2 zeigt die zeitliche Lage von zwei verschiedenen Versionen einer Schnittstelle eines SPS und den Aufruf eines Dienstes.

Beschreibung einer Realisierung

Die Erfindung wird nun an Hand der Abbildungen Fig. 1 und Fig. 2 im Detail beschrieben.

Fig. 1 zeigt den Aufbau eines SoS bestehend aus vier unabhängigen EDV Systemen und ein sie verbindendes Kommunikationssystem 130 das mit den Datenleitungen 111 mit den vier EDV Systemen 120, 151, 152 und 153 verbunden ist. Es wird davon ausgegangen, dass die vier EDV Systemen 120, 151, 152 und 153 eine gemeinsame globale Zeit mit der Präzision P aufbauen. Diese globale Zeit kann mit der externen Zeit TAI z.B. über ein GPS Signal mit der Genauigkeit A synchronisiert werden. Der Aufbau einer fehlertoleranten globalen Zeit mit der Präzision P und die externe Uhrensynchronisation mit der Genauigkeit A sind in [4] ausführlich beschrieben. Es ist vorteilhaft, wenn die vier EDV Systemen 120, 151, 152 und 153 nur in erlaubten Zeitintervallen (wie in [4] unter dem Namen *sparse time model* beschrieben) Nachrichten senden, damit die Nachrichten im gesamten SoS konsistent geordnet werden können.

Der Problemsteller 100 ist über eine Datenverbindung 110 mit dem EDV System 120 verbunden. Das EDV System 120 unterstützt den Problemsteller 100 bei der Formulierung seines Problems und dem Aufsuchen eines konkreten *Lösungszustands*. In der Folge unterteilt das EDV System 120 den Pfad vom gegebenen *Ausgangszustand* zum *Lösungszustand* durch die Festlegung einer Sequenz von *Zwischenzuständen*, wie z.B. in [5] angeführt. Dieses Problemlösungsverfahren basiert auf der Annahme dass es wesentlich einfacher ist einen Zielzustand anzugeben als einen Weg zu finden, der zu diesem Zielzustand führt [5, p.76]. Das System 120 sucht zuerst einen Dienstanbieter der die Transformation vom Ausgangszustand zum ersten Zwischenzustand durchführen kann. Ein EDV System das einen Dienst anfordert bezeichnen wird als ein *SRS* (*service requesting system*) während ein EDV System, das einen Dienst anbietet ein *SPS* (*service providing system*) genannt wird. Dieser Prozess der Transformation von einem Zwischenzustand zum nächsten Zwischenzustand wird wiederholt, bis der Lösungszustand erreicht ist.

Im Beispiel der Fig. 1 verwaltet das EDV System 153 einen *Katalog* in dem die Dienste von vorhandenen Diensteanbietern syntaktisch und semantisch beschrieben sind. Das SRS 120 konsultiert diesen Katalog um herauszufinden, welches EDV System den gewünschten Dienst anbietet [3]. Es wird nun von der Annahme ausgegangen, dass das SPS 151 ein Diensteanbieter ist, der den gesuchten Dienst anbietet. Als nächstes untersucht das SRS 120 die Schnittstellenspezifikation vom SPS 151. Erfindungsgemäß ist in dieser Schnittstellenspezifikation mindestens ein Gültigkeitszeitfeld enthalten das angibt, bis zu welchem zukünftigen Zeitpunkt diese Schnittstellenspezifikation gültig ist. Das EDV System 151 verpflichtet sich selbst, vor dem im Gültigkeitszeitfeld enthaltenen Zeitpunkt keine Änderung der Schnittstellenspezifikation vorzunehmen. In der Schnittstellenspezifikation vom SPS 151 ist auch ein weiteres Feld, genannt *timeout_1*, enthalten. In diesem Feld befindet sich das maximale Zeitintervall das benötigt wird um eine Dienstanforderung im SPS 151 abzuarbeiten. Das SRS 120 benötigt diese Information um einerseits zu überprüfen ob die Schnittstellenspezifikation von SPS 151 auch noch am Ende der Abarbeitungszeit gültig sein wird und andererseits um einen *fail-silent Fehler* (siehe [4]) vom SPS 151 sofort erkennen zu können.

Der zeitliche Ablauf der Ereignisse ist in Fig. 2 gezeigt. Auf der Abszisse 200 von Fig. 2 ist das Fortschreiten der Echtzeit dargestellt. Zum Zeitpunkt 211 legt das SPS 151 das Intervall fest, während dessen die Schnittstellenspezifikation von SPS 151 gültig ist. Dieses Intervall hat die Dauer 210 und den Endzeitpunkt 212. Im Intervall zwischen den Ereignissen 212 und 221 bietet SPS 151 keine gültige Schnittstellenspezifikation an. Zum Zeitpunkt 221 legt das SPS 151 das nächste Intervall mit der Dauer 220 und dem Endzeitpunkt 222 fest während dessen die Schnittstellenspezifikation von SPS 151 gültig ist. Zum Zeitpunkt 241 fordert das SRS 120 einen Dienst von SPS 151 an. Die Abarbeitung dieses Dienstes benötigt die durch das Intervall 250 dargestellte Zeit (*timeout_1*). Da das Ende von *timeout_1* 242 vor 222 liegt, kann der Auftrag vor dem Ablauf der Gültigkeit ausgeführt werden. Wenn bis zum Zeitpunkt 242 keine Antwort vom SPS 151 an das SRS 110 eintrifft, so nimmt das SRS 110 an dass ein *fail-silent* Ausfall von SPS 151 vorliegt.

Um sicherzustellen, dass ein SRS 110 keine zeitlich ungültige Schnittstelle anspricht kann die Computer Hardware des angesprochenen SPS 151 autonom überprüfen ob die globale Zeit gleich einer der in der Schnittstelle enthaltenen Gültigkeitszeiten ist und, falls dies der Fall ist, bei Zugriff auf die Schnittstelle durch ein SRS 151 eine Fehlermeldung auslösen.

Patentansprüche

1. Verfahren zur statischen oder dynamischen Integration einer Menge von EDV Systemen in ein *System-of-Systems* (SoS) **dadurch gekennzeichnet dass** alle am SoS beteiligten EDV Systeme über eine globale Zeit mit der Präzision P verfügen und wo ein einen Dienst anbietendes SPS 151 in der den Dienst spezifizierenden Schnittstelle mindestens ein Gültigkeitszeitfeld verwaltet das angibt, bis zu welchem zukünftigen Gültigkeitszeitpunkt die Syntax und Semantik der Schnittstellenspezifikation und die darin enthaltenen Informationen gültig sein werden und wo das SPS 151 garantiert, dass vor diesem Gültigkeitszeitpunkt keine Änderungen in der Schnittstellenspezifikation und den darin enthaltenen Informationen vorgenommen werden.
2. Verfahren nach Anspruch 1 **dadurch gekennzeichnet dass** die in Anspruch 1 angeführte globale Zeit mit der TAI mit der Genauigkeit A synchronisiert ist.
3. Verfahren nach Anspruch 1 und/oder 2 **dadurch gekennzeichnet dass** die globale Zeit *sparse* ist.
4. Verfahren nach einem oder mehreren der Ansprüche 1 bis 3 **dadurch gekennzeichnet dass** die den Dienst spezifizierenden Schnittstelle zwei Gültigkeitszeitfelder enthält, wobei ein Gültigkeitszeitfeld den Zeitpunkt angibt bis zu dem die Syntax und Semantik der Schnittstelle unverändert bleiben und das zweite Gültigkeitszeitfeld den Zeitpunkt angibt, bis zu dem die Informationen in der Schnittstelle zeitlich gültig sind.
5. Verfahren nach einem oder mehreren der Ansprüche 1 bis 4 **dadurch gekennzeichnet dass** die den Dienst spezifizierenden Schnittstelle ein Gültigkeitszeitfeld enthält das den Zeitpunkt angibt, bis zu dem die Informationen in der Schnittstelle zeitlich gültig sind.
6. Verfahren nach einem oder mehreren der Ansprüche 1 bis 5 **dadurch gekennzeichnet dass** ein SRS vor der Inanspruchnahme eines Dienstes eines SPS überprüft, ob das erwartete Ende des in Anspruch genommen Dienstes vor dem Gültigkeitszeitpunkt liegt, der in der Schnittstellenspezifikation des aufgerufenen SPS enthalten ist.
7. Verfahren nach einem oder mehreren der Ansprüche 1 bis 6 **dadurch gekennzeichnet, dass** die Schnittstellenspezifikation eines SPS zusätzlich ein Timeoutfeld_1 enthält in dem angegeben ist, wie viel Zeit das SPS benötigen wird, um eine Anforderung eines SRS zu bearbeiten.
8. Verfahren nach einem oder mehreren der Ansprüche 1 bis 7 **dadurch gekennzeichnet dass** ein SRS bei Aufruf eines Dienstes des SPS den im *Timeoutfeld_1* des SPS enthaltene Wert als Timeout setzt, um einen Antwortfehler eines SRS sofort erkennen zu können.
9. Verteilte Computer Architektur zur dynamischen Integration einer Menge von Subsystemen (SS) in ein *System-of-Systems* (SoS) **dadurch gekennzeichnet dass** einer oder mehrere der Verfahrensansprüche 1 bis 8 in Hardware realisiert wird.

10. Verteilte Computer Architektur zur dynamischen Integration einer Menge von Subsystemen (SS) in ein *System-of-Systems* (SoS) **dadurch gekennzeichnet** dass die Computer Hardware eines SPS autonom überprüft ob die globale Zeit gleich der in den Ansprüchen 1-6 enthaltenen Gültigkeitszeit ist und, falls dies der Fall ist, bei Zugriff auf die Schnittstelle durch ein SRS eine Fehlermeldung auslöst.

010672

7

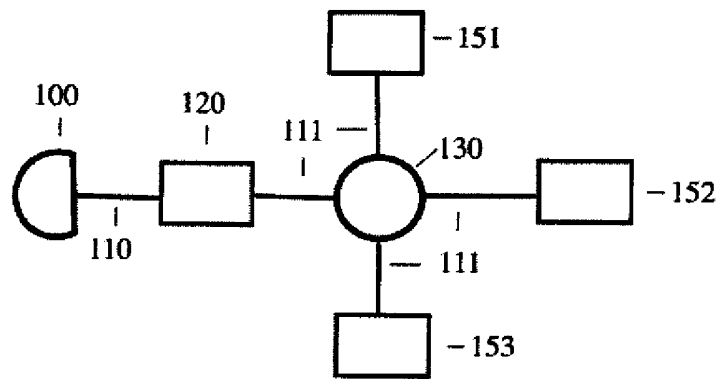


FIG. 1

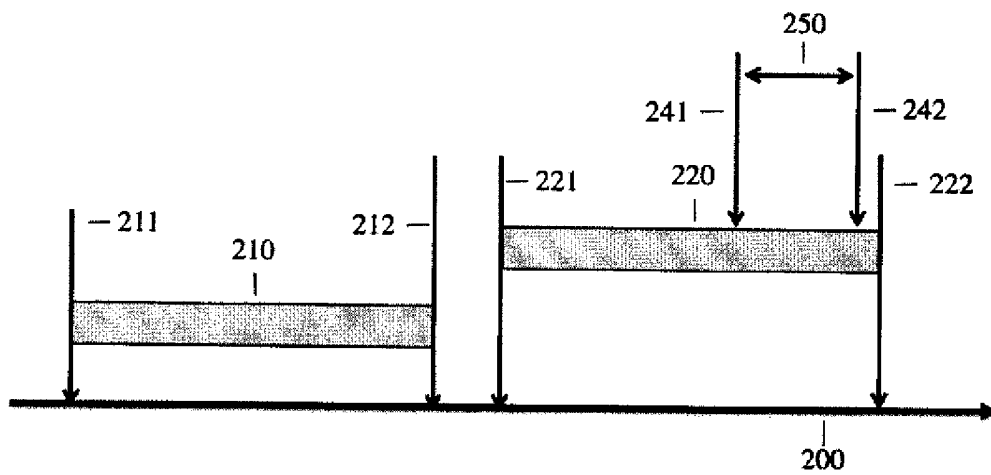
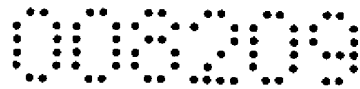


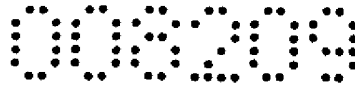
FIG. 2



Patentansprüche

1. Verfahren zur statischen oder dynamischen Integration einer Menge von EDV-Systemen in ein System-of-Systems (SoS), dadurch gekennzeichnet dass, alle am SoS beteiligten EDV-Systeme über eine globale Zeit mit der Präzision P verfügen und wo ein einen Dienst anbietendes service providing system (SPS) 151 in der den Dienst spezifizierenden Schnittstelle mindestens ein Gültigkeitszeitfeld verwaltet das angibt, bis zu welchem zukünftigen Gültigkeitszeitpunkt die Syntax und Semantik der Schnittstellenspezifikation und die darin enthaltenen Informationen gültig sein werden und wo das SPS 151 garantiert, dass vor diesem Gültigkeitszeitpunkt keine Änderungen in der Schnittstellenspezifikation und den darin enthaltenen Informationen vorgenommen werden.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die globale Zeit mit der temps atomique international (TAI) mit der Genauigkeit A synchronisiert ist.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die globale Zeit *sparse* ist, sodass die EDV-Systeme nur in erlaubten Zeitintervallen kommunizieren.
4. Verfahren nach einem oder mehreren der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass die den Dienst spezifizierenden Schnittstelle zwei Gültigkeitszeitfelder enthält, wobei ein Gültigkeitszeitfeld den Zeitpunkt angibt bis zu dem die Syntax und Semantik der Schnittstelle unverändert bleiben und das zweite Gültigkeitszeitfeld den Zeitpunkt angibt, bis zu dem die Informationen in der Schnittstelle zeitlich gültig sind.
5. Verfahren nach einem oder mehreren der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass ein service requesting system (SRS) vor der Inanspruchnahme eines Dienstes eines SPS überprüft, ob das erwartete Ende

NACHGEREICHT



des in Anspruch genommen Dienstes vor dem Gültigkeitszeitpunkt liegt, der in der Schnittstellenspezifikation des aufgerufenen SPS enthalten ist.

6. Verfahren nach einem oder mehreren der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass die Schnittstellenspezifikation eines SPS zusätzlich ein Timeoutfeld_1 enthält, in dem angegeben ist, wie viel Zeit das SPS benötigen wird, um eine Anforderung eines service requesting system (SRS) zu bearbeiten.
7. Verfahren nach einem oder mehreren der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass ein service requesting system SRS bei Aufruf eines Dienstes des SPS den in einem Timeoutfeld_1 des SPS enthaltene Wert als Timeout setzt, um einen Antwortfehler eines SPS sofort erkennen zu können.
8. Verteilte Computer-Architektur zur statischen oder dynamischen Integration einer Menge von EDV-Systemen in ein System-of-Systems (SoS), dadurch gekennzeichnet, dass einer oder mehrere der Verfahrensansprüche 1 bis 7 in Hardware realisiert ist.
9. Verteilte Computer-Architektur zur dynamischen Integration einer Menge von EDV-Systemen in ein System-of-Systems (SoS) nach Anspruch 8, dadurch gekennzeichnet, dass die Computer-Hardware eines SPS autonom überprüft, ob die globale Zeit gleich der Gültigkeitszeit ist, und, falls dies der Fall ist, bei Zugriff auf die Schnittstelle durch ein service requesting system (SRS) eine Fehlermeldung auslöst.

NACHGEREICHT