

(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
18. Mai 2006 (18.05.2006)

PCT

(10) Internationale Veröffentlichungsnummer
WO 2006/050550 A1

(51) Internationale Patentklassifikation:
G06F 9/45 (2006.01) **G06F 9/44** (2006.01)

(74) **Anwalt: MATSCHNIG, Franz**; Siebensterngasse 54,
A-1071 Wien (AT).

(21) Internationales Aktenzeichen: PCT/AT2005/000457

(81) **Bestimmungsstaaten** (*soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(22) Internationales Anmeldedatum:
15. November 2005 (15.11.2005)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
A 1903/2004 15. November 2004 (15.11.2004) AT

(84) **Bestimmungsstaaten** (*soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(71) **Anmelder** (*für alle Bestimmungsstaaten mit Ausnahme von US*): **TECHNISCHE UNIVERSITÄT WIEN** [AT/AT]; Karlsplatz 13, A-1040 Wien (AT).

(72) **Erfinder; und**

(75) **Erfinder/Anmelder** (*nur für US*): **KIRNER, Raimund** [AT/AT]; Triester Strasse 9/15, A-1100 Wien (AT). **PUSCHNER, Peter** [AT/AT]; Franz-Mika-Weg 5/1/16, A-1100 Wien (AT).

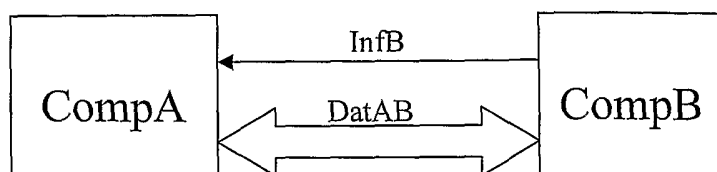
Veröffentlicht:

— mit internationalem Recherchenbericht

[Fortsetzung auf der nächsten Seite]

(54) **Title:** METHOD FOR EXCHANGING DATA

(54) **Bezeichnung:** VERFAHREN ZUM AUSTAUSCH VON DATEN



(57) **Abstract:** The invention relates to a method for exchanging data (DatAB) between a first computer system (CompA) and a second computer system (CompB), said data (DatAB) being represented on the first computer system (CompA) in a first platform-specific data representation and on the second computer system (CompB) in a second platform-specific data

representation. The term "platform-specific data representation" refers to the binary representation of numerical data which is conditional on the used platform (processor and compiler). The conversion of data (DatAB) between the first and second data representation is carried out in both directions on the first computer system (CompA), whereby the information (InfB) on the second computer system (CompB) required by the first computer system (CompA) is completely calculated on the second computer system (CompB), said information (InfB) being required for conversion of the data (DatAB) in both directions of data exchange.

(57) **Zusammenfassung:** Die Erfindung betrifft ein Verfahren zum Austausch von Daten (DatAB) zwischen einem ersten Computersystem (CompA) und einem zweiten Computersystem (CompB), wobei die Daten (DatAB) auf dem ersten Computersystem (CompA) in einer ersten plattformspezifischen Datenrepräsentation und auf dem zweiten Computersystem (CompB) in einer zweiten plattformspezifischen Datenrepräsentation dargestellt werden, wobei plattformspezifische Datenrepräsentation die binäre Repräsentation von numerischen Daten bezeichnet, die durch die verwendete Plattform (Prozessor und Compiler) bedingt ist. Die Konvertierung der Daten (DatAB) zwischen der ersten und zweiten plattformspezifischen Datenrepräsentation wird in beiden Richtungen auf dem ersten Computersystem (CompA) durchgeführt, wobei die auf dem ersten Computersystem (CompA) benötigte Information (InfB) über das zweite Computersystem (CompB), welche Information (InfB) zur Umwandlung der Daten (DatAB) für beide Richtungen des Datenaustausches notwendig ist, vollständig auf dem zweiten Computersystem (CompB) berechnet wird.



WO 2006/050550 A1



— vor Ablauf der für Änderungen der Ansprüche geltenden Frist; Veröffentlichung wird wiederholt, falls Änderungen eintreffen

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.

VERFAHREN ZUM AUSTAUSCH VON DATEN

Die Erfindung betrifft ein Verfahren zum Austausch von Daten zwischen einem ersten Computersystem und einem zweiten Computersystem, wobei die Daten auf dem ersten Computersystem in einer ersten plattformspezifischen Datenrepräsentation und auf dem zweiten Computersystem in einer zweiten plattformspezifischen Datenrepräsentation dargestellt werden, und die Konvertierung der Daten zwischen der ersten und zweiten Datenrepräsentation in beiden Richtungen auf dem ersten Computersystem durchgeführt wird, wobei plattformspezifische Datenrepräsentation die die binäre Repräsentation von numerischen Daten bezeichnet, die durch verwendeten Prozessor und Compiler bedingt ist. Diese konkrete Definition von plattformspezifischer Datenrepräsentation ist wesentlich für die Problemstellung und der Lösung durch die vorliegende Erfindung. Im Folgenden bezeichnen heterogene Computersysteme jene Computersysteme, welche eine unterschiedliche plattformspezifischer Datenrepräsentation aufweisen.

Die beiden Computersysteme – unter Computersystem ist dabei beispielsweise ein Prozessor auf dem Daten verarbeitet werden zu verstehen – können dabei aus völlig unterschiedlichen Hardware- und Softwarestrukturen bestehen. Insbesondere eignet sich die vorliegende Erfindung, wie weiter unten noch erläutert, wenn auf dem zweiten Computersystem, welches im folgenden auch als „Target-Computersystem“ oder „Target“ bezeichnet wird, der Ressourcenverbrauch, welcher für die Umwandlung der Daten in eine andere plattformspezifische Datenrepräsentation nötig ist, minimiert ist. Das erste Computersystem wird im folgenden auch als „Host-Computersystem“ oder „Host“ bezeichnet.

Im Folgenden soll der oben eingeführte Begriff der „plattformspezifischen Datenrepräsentation“ näher erläutert werden. In diesem Dokument wird unter diesem Begriff die binäre Repräsentation von numerischen Daten, welche durch die verwendete Computerplattform (Prozessor und/oder Compiler) bedingt ist, verstanden. Relevante Parameter für diese binäre Repräsentation von numerischen Daten sind beispielsweise:

- Gesamtgröße des Datenfeldes zur Speicherung von Datenwertes (z.B. Anzahl der Bits)
- Adressausrichtung des Datenfeldes (d.h., von welcher Größe die konkrete Adresse des Datenfeldes immer ein Vielfaches sein muss)

- Binäre Kodierung von Zahlen (z.B., Einer- bzw. Zweierkomplement für ganzzahlige Werte, oder IEEE-Format für Fließkommazahlen, etc.)
- Geometrische Anordnung der einzelnen Datenbits innerhalb des Datenfeldes (z.B., Endianität (little versus big endian), Platzierung von Elementen zusammengesetzter Daten innerhalb des Datenfeldes, etc.)

Das Problem der Konvertierung von Daten zum Austausch von Informationen zwischen heterogenen Computersystemen ist schon relativ lange bekannt, da sich die plattformspezifischen Datenformate von verschiedenen Computersystemen und den darauf laufenden Programmen typischerweise in gewissen Details zu einander unterscheiden. Prinzipiell gibt es zwei Arten, mit dem Problem der unterschiedlichen Datenformate umzugehen:

1. Austausch der Daten über ein einheitliches Zwischenformat: Jedes Computersystem muss dann die Daten zwischen dem internen plattformspezifischen Format und dem einheitlichen Zwischenformat konvertieren. Der Vorteil dieser Methode ist, dass Computersysteme Daten austauschen können ohne gegenseitiges Wissen über die jeweils verwendeten plattformspezifischen Datenformate zu besitzen. Der Nachteil dieser Methode ist mangelnde Effizienz, da gewisse Aspekte der Datenrepräsentation oft unnötig konvertiert werden müssen.
2. Austausch der Daten über ein plattformspezifisches Datenformat: Daten werden dabei nur dann konvertiert, wenn es aufgrund der plattformspezifischen Datenformate der beteiligten Computersysteme notwendig ist. Der Vorteil dieser Methode ist, dass die Datenkonvertierung effizienter als bei der ersten Methode realisierbar ist, da unnötige Konvertierungen vermieden werden. Grundsätzlich sind zwei Varianten denkbar:
 - a. Symmetrisch: Jedes System konvertiert die Daten im Bedarfsfalle nur beim Empfangen (bzw. nur beim Senden). Damit übernehmen beide Computersysteme einen Teil der notwendigen Datenkonvertierung. Diese Methode kann beispielsweise realisiert werden, indem zu Beginn der übertragenen Daten eine Formatbeschreibung der plattformspezifischen Datenrepräsentation hinzugefügt wird. Nachteilig an dieser Variante ist dabei, dass beide Computersysteme mit der Datenkonvertierung beschäftigt sind.
 - b. Asymmetrisch: Ein Computersystem wandelt die Daten vor dem Senden in das plattformspezifische Format des Zielsystems bzw. nach dem Empfangen

in das eigene plattformspezifische Format um. Bei dieser Methode ist das zweite Computersystem völlig vom Aufwand der Datenkonvertierung entlastet. Dies setzt allerdings voraus, dass bei dem die Konvertierung durchführenden Computersystem Wissen über das interne plattformspezifische Datenformat des anderen Computersystems bekannt ist.

Ein Datenaustauschverfahren nach Methode 2.b eignet sich besonders für Target-Computersysteme, bei denen der Ressourcenverbrauch, welcher für die Umwandlung der Daten nötig ist, minimiert ist. Nachteilig an den bisher bekannten Verfahren nach Methode 2.b ist allerdings, dass die Unterstützung der plattformspezifischen Datenrepräsentation einer konkreten Target-Plattform *a priori* in den Host eingebaut sein muss, d.h. dass der Host über die notwendigen plattformspezifischen Informationen für den Datenaustausch mit einem bestimmten Target bereits verfügen muss. Die Unterstützung eines neuen Targets erfordert daher eine explizite *a priori* Anpassung Ausführungslogik (z.B. Software) auf dem Host.

Es ist eine Aufgabe der Erfindung, ein Datenaustauschverfahren nach Methode 2.b dahingehend zu verbessern, dass auch für neue Targets eine solche Anpassung der Ausführungslogik (z.B. Software) auf dem Host-Computersystem nicht notwendig ist.

Diese Aufgabe wird mit einem eingangs erwähnten Verfahren dadurch gelöst, dass erfindungsgemäß die auf dem ersten Computersystem benötigte plattformspezifische Information über das zweite Computersystem, welche Information zur Umwandlung der Daten für beide Richtungen des Datenaustausches notwendig ist, vollständig auf dem zweiten Computersystem berechnet wird.

Die oben genannte Aufgabe wird weiters noch mit einem entsprechenden plattformunabhängigen Verfahren mit einem entsprechenden Target-Computersystem sowie mit einer Testumgebung wie in den Ansprüchen dargelegt gelöst.

Die Erfindung betrifft ein ressourcensparendes Verfahren zum Datenaustausch zwischen einem mit relativ vielen verfügbaren Ressourcen ausgestatteten Hostcomputer und einem mit relativ knappen Ressourcen ausgestatteten Eingebetteten System (Target). Das eingebettete System verwendet zur Datenübertragung der ausgetauschten Daten wenig Rechenzeit und Speicherplatz.

Die Konvertierung der Daten wird auf dem Hostcomputer durchgeführt, wobei die hierzu benötigten plattformspezifischen Informationen über das Target automatisch auf dem Target berechnet und an den Hostcomputer übertragen werden.

Zugleich wird beim Host a priori keine Information über das plattformspezifische Datenformat des Targets benötigt, da das Target die benötigte Information selbst berechnet und an den Host überträgt.

Der Host kümmert sich somit um die Konvertierung der Daten in die jeweiligen plattformspezifischen Datenformate. Das Target muss dabei lediglich einmal vor dem ersten Austausch von Nutzdaten gewisse Kenngrößen der eigenen Plattform bestimmen und an den Host schicken. Diese Kenngrößen enthalten beispielsweise die Endianität (little endian oder big endian, d.h. die Anordnung der Bytes innerhalb eines Datenwortes). Das Wissen über die Endianität ist erforderlich, um weitere Informationen interpretieren zu können.

Ein wichtiger Aspekt bei einem solchen Datenaustauschverfahren ist die Portierbarkeit auf neue Plattformen sowohl auf Seite des Hosts wie auch auf Seite des Targets. Das Verfahren dieser Erfindung trägt dem Rechnung, indem das Verfahren zur Datenkonvertierung auf Host und Target in plattform-unabhängiger Form realisierbar ist. Weder die Ausführungslogik des Hosts noch die Ausführungslogik des Targets brauchen geändert werden, wenn das jeweils andere System auf eine neue Plattform migriert wurde.

Um ein Datenaustauschverfahren einzusetzen ist es auch wichtig, dass es mit den Entwicklungswerkzeugen, welche für eine konkrete Computerplattform zur Verfügung stehen, kompatibel ist. Das Verfahren dieser Erfindung trägt dem Rechnung, indem es für die Realisierung keinerlei spezielle Entwicklungswerkzeuge benötigt. Das Verfahren ist in plattform-unabhängiger Form realisierbar.

Das Verfahren lässt sich mit Standard-Werkzeugen zur Systementwicklung realisieren. Vorzugsweise ist entsprechend Anspruch 2 das Verfahren zur Berechnung der Information auf dem zweiten Computersystem in plattform-unabhängiger Form realisiert. Der benötigte Ausführungslogik auf Seite des Targets lässt sich somit plattform-unabhängig realisieren, wobei der Hostcomputer ohne Änderung seiner Ausführungslogik mit unterschiedlichen Targets kommunizieren kann. Es sind somit keine speziell angepassten Entwicklungswerkzeuge notwendig, um eine Kommunikation basierend auf dem Verfahren zu realisieren.

Entsprechend Anspruch 3 ist vorgesehen, dass die plattformspezifische Information, welche von dem zweiten Computersystem berechnet wird, ein Endian-Flag zur Beschreibung der Byte-Reihenfolge, in welcher Daten auf dem zweiten Computersystem abgelegt werden, enthält. Damit werden die Speicherstrategien Little Endian bzw. Big Endian dargestellt. Der Wert dieses Endian-Flags ist der erste Datenwert innerhalb der Information, da dieser bestimmt, wie die weiteren Werte der Information bezüglich Byte-Reihenfolge zu interpretieren sind. Weiters können noch Informationen über die Länge oder Wertebereiche der Grunddatentypen von dem zweiten Computersystem enthalten sein.

Um das Problem der unterschiedlichen internen Adressausrichtung komplexer Datentypen auf Host und Target zu umgehen, ist entsprechend Anspruch 4 vorgesehen, dass in den Daten die Werte von Objekten komplexer Datentypen, wie beispielsweise Strukturen, in flacher Form, d.h. als Sequenz von in den Objekten komplexer Datentypen enthaltenen Elementen von Grunddatentypen, gespeichert werden.

Gemäß Anspruch 5 ist weiters vorgesehen, dass das erste Computersystem und das zweite Computersystem über einen elektronischen Datenkanal verbunden sind und automatisch vor der ersten Übermittlung der Daten die Information von dem zweiten Computersystem auf das erste Computersystem übertragen wird.

Mit den Merkmalen des Anspruchs 6, nämlich dass von dem zweiten Computersystem die Verbindung zwischen der flachen Datenstruktur der Daten und den komplexen Datenstrukturen über eine eigene Übersetzungstabelle durchgeführt wird, wird erreicht, dass die Größe der für diese Verbindung notwendigen Ausführungslogik unabhängig ist von Typ und Anzahl der Datenelemente in den Daten.

Weiters ist noch vorgesehen, dass die für die Verbindung zwischen den übertragenen Objekten von Grunddatentypen und den am Target verwendeten Objekten von komplexen Datentypen sowie für die Berechnung der Information notwendigen Transformationsschritte vollständig als plattform-unabhängiges Verfahren spezifiziert sind.

Durch diese Merkmale ergibt es sich, dass das erfindungsgemäße Verfahren ohne Änderung weiterhin funktioniert, sollte ein bestimmtes Target-Computersystem durch ein anderes Target, welches unterschiedliche Hardware- und/oder Softwarestrukturen aufweist, ersetzt werden. Das Verfahren kann also sehr leicht auf unterschiedliche Plattformen portiert werden.

Insgesamt lässt sich feststellen, dass das Verfahren keine spezielle Unterstützung seitens der Entwicklungswerkzeuge benötigt, womit die Portabilität des Verfahrens auch diesbezüglich erreicht wird. Die Umwandlung der Daten auf Seite des Host-Computersystems kann unter Zuhilfenahme der plattformspezifischen Information in plattform-unabhängiger Form realisiert werden und muss nur die Umwandlung aller Grunddatentypen zwischen den beiden Plattformen unterstützen.

Neben dem oben beschriebenen Verfahren, einem Computersystem entsprechend den Ansprüchen 8 -12 wird die eingangs genannte Aufgabe auch noch mit einer Testumgebung gelöst entsprechend den Ansprüchen 13, 14.

Bei dieser Testumgebung stellt die Speicherstruktur die benötigten indirekten Datenzugriffe bereit, falls Datentypen nicht direkt als Wert-Parameter an das zu testende (Unter)Programm übergeben werden. Damit kann das Datenaustauschverfahren der Testumgebung ohne Änderung auf unterschiedlichen Plattformen verwendet werden.

Im Folgenden ist die Erfindung an Hand der Zeichnung näher erläutert. In dieser zeigt

Fig. 1 Informationsfluss zwischen Host und Target,

Fig. 2 schematisch das Speicherkonzept am Target zur Umwandlung zwischen kommunizierten Daten und internen Daten benutzt wird, und

Fig. 3 schematisch die Erweiterung des Speicherkonzeptes am Target, um dieses Datenaustauschverfahren für eine dezentrale Testumgebung verwenden zu können.

Fig. 1 veranschaulicht den Datenfluss und Informationsfluss zwischen Host (CompA) und Target (CompB). Hierzu ist festzuhalten, dass bevor Daten (DatAB) zwischen Host (CompA) und Target (CompB) ausgetauscht werden können, das Target (CompB) einmal an den Host (CompA) Informationen (InfB) über Plattform-Eigenschaften des Targets (CompB) übertragen muss. Diese Plattformeigenschaften werden auf dem Target (CompB) automatisch mittels kurzer Berechnungsschritte ermittelt. Sobald die Information (InfB) vom Host (CompA) empfangen wurde, hat der Host (CompA) ausreichend Daten zur Verfügung, um zu wissen, in welcher Form das Target (CompB) die Daten (DatAB) verarbeitet. Die Daten (DatAB) können in beide Richtungen zwischen Host und Target übertragen werden. Die notwendige Konvertierung des Datenformates der Daten (DatAB) wird zur Gänze auf Seite des Hosts (CompA) durchgeführt. Anspruch 3 beschreibt den Informationsgehalt der Informationen (InfB).

Fig. 2 zeigt das Speicherkonzept auf dem Target (CompB), mittels dessen Objekte zusammengesetzter Datentypen (DatB) als Sequenz von Objekten von Grunddatentypen (DatAB) übertragen werden.

Als Grunddatentypen werden dabei Datentypen bezeichnet, welche nur aus einem Zahlenwert bestehen. Anspruch 4 beschreibt diese Aufsplittung der Objekte komplexer Datentypen (DatB) in deren Elemente von Grunddatentypen (DatAB). Abhängig von der verwendeten Software-Entwicklungsumgebung stehen hierfür unterschiedliche Wertebereiche sowie unterschiedliche Zahldarstellungen wie etwa ganzzahlige Werte oder Dezimalwerte zur Verfügung. Zusammengesetzte Datentypen können aus Grunddatentypen, sowie aus hierarchisch zusammengesetzten Datentypen, bestehen. Typische Formen dieser Zusammensetzung, wie sie etwa in Programmiersprachen unterstützt werden, sind Arrays (indizierbare Sequenz gleichartiger Datentypen), Strukturen (Verband beliebiger Datentypen). Bei den Objekten von Grunddatentypen (DatAB) spricht man daher von einer flachen Datenstruktur, während man bei den zusammengesetzten Datentypen von Datentypen (DatB) von einer komplexen Datenstruktur spricht.

Da es sehr viele unterschiedliche Möglichkeiten gibt, wie diese Zusammensetzung von Datentypen konkret auf Speicherzellen abgebildet werden kann, werden Objekte zusammengesetzter Datentypen als Sequenz von Objekten der Grunddatentypen übertragen. Die Zuordnung von Elementen der zusammengesetzten Datentypen auf die übertragenen Grunddatentypen wird über eine Tabelle (TabB) ermöglicht. Diese Tabelle (TabB) enthält einen Eintrag für je ein Objekt eines Grunddatentypes von (DatB), wobei dieser Eintrag eine Referenz auf den Speicherplatz enthält, wo dieses Objekt für die Programmausführung im Target (CompB) erwartet wird. Dieser Speicherplatz kann sich somit natürlich auch innerhalb eines Objektes eines zusammengesetzten Datentypes aus (DatB) befinden.

Nun sei auf Fig. 3 Bezug genommen, in welcher eine Erweiterung des Speicherinterfaces des Datenaustauschverfahrens zur Realisierung einer dezentralen Testumgebung beschrieben ist. Ziel dieser dezentralen Testumgebung ist es, auf dem Target-Computersystem (CompB) ein (Unter)Programm laufen zu lassen, wobei die entsprechenden Testdaten vom Host (CompA) bereitgestellt werden und das Testergebnis zurück an den Host (CompA) übertragen wird. Es kann dabei vorkommen, dass das zu testende (Unter)Programm nicht direkt mit den Werten der Testdaten aufgerufen werden kann. Dies ist beispielsweise dann der Fall, wenn das zu testende (Unter)Programm auf die Testdaten indirekt über andere Datenstrukturen zugreift. In diesem Fall müssen vor dem Test diese zusätzlichen Datenstrukturen aufgebaut und initialisiert werden. Der dafür notwendige Speicherbereich ist mit (MGlue)

benannt. Diese Erweiterung für die Realisierung einer dezentralen Testumgebung ist in Anspruch 13 und 14 beschrieben.

Beispielhafte Realisierung der Datenstrukturen

Im folgenden sei anhand eines Beispiels verdeutlicht, wie die beschriebenen Speicherbereiche für das Datenaustauschverfahren und eine darauf aufbauende Testumgebung realisiert und verwendet werden können. Das Datenaustauschverfahren lässt sich sowohl in Hardware als auch in Software realisieren. Da sich jedoch mit Programmiersprachen eine Ausführungslogik sehr kompakt und in einem allgemein bekannten Format darstellen lässt, wurde zur Beschreibung gewisser Mechanismen die Programmiersprache ANSI C gewählt. Die Konzepte der Testumgebung werden ebenfalls für die Programmiersprache ANSI C skizziert.

Angenommen, es sei auf dem Target CompB das Unterprogramm

```
typedef struct { int a[3]; char b; } Sn_t;
```

```
void test (Sn_t *sn1[20], Sn_t sn2);
```

zu testen. Um die flexiblen Möglichkeiten durch den Speicherbereich (MGlue) zu demonstrieren, sei angenommen, dass die relevanten Testdaten aus den zwei Datensätzen sv1 und sv2 vom Typ Sn_t bestehen. Weiters sei festgelegt, dass bezüglich der Eingabeparameter des zu testenden Unterprogramms das erste Element des Arrays sn1 auf den Testwert sv1 zeigen soll und das zweite Element als Null-Pointer initialisiert sein soll. Der zweite Eingabeparameter soll direkt mit dem Testwert sv2 initialisiert werden.

Um die Testdaten lokal auf dem Target (CompB) speichern zu können, ist folgender Speicherbereich zu allokalieren:

```
Sn_t sv1, sv2;
```

Da die zu testende Prozedur im ersten Argument eine komplexere Datenstruktur benutzt, ist zusätzlicher Speicherbereich (MGlue) zum Aufbau der komplexen Datenstruktur (DatB) zu allokalieren:

```
Sn_t *av[20];
```

welcher mit folgenden Werten initialisiert wird:

```
av[0] = av[0] = &sv1;
```

```
av[1] = (Sn_t*) 0;
```

Die zu testende Prozedur könnte nun folgendermaßen aufgerufen werden:

```
test (av, sv2);
```

Für den Empfang der vom Host (CompA) kommenden Testdaten kann nun auf dem Target (CompB) folgende Übersetzungstabelle (TabB) verwendet werden:

```
typedef struct { void *ref; int size; } data_list_t;
```

```
data_list_t data[] = {  
  
    { &(sv1.a), sizeof(sv1.a) }, \  
  
    { &(sv1.b), sizeof(sv1.b) }, \  
  
    { &(sv2.a), sizeof(sv2.a) }, \  
  
    { &(sv2.a), sizeof(sv2.b) } };
```

Im Target (CompB) wird eine Empfangseinheit realisiert, welche die Objekte von Grunddatentypen kommend vom Host (CompA) als Bytestrom interpretiert und auf die entsprechenden Speicherzellen, welche in der Tabelle (TabB) spezifiziert sind, verteilt. Dass das Speicherlayout von Objekten zusammengesetzter Datentypen auf Host (CompA) und Target CompB anders aufgebaut ist, wird auf ressourcensparende Weise abstrahiert, indem auf dem Target (CompB) die Daten (DatB) mit der Übersetzungstabelle (TabB) auf die Objekte von Grunddatentypen in (DatAB) aufgeteilt werden.

Mit diesem Beispiel wurde demonstriert, wie man eine Datenkommunikation von Host (CompA) nach Target (CompB) für eine dezentrale Testumgebung realisieren kann. Die beschriebenen Datenstrukturen (Tabelle (TabB) und Speicherbereich (MGlue)) sind spezifisch für das zu testende (Unter)Programm angegeben worden.

Die Realisierung von Datenstrukturen für einen Datenaustausch von Target (CompB) nach Host (CompA) folgt demselben Schema.

Beispielhafte Berechnung der Plattform-Eigenschaften

Im Folgenden wird die Realisierungsmöglichkeit der automatischen Berechnung der Plattform-Eigenschaften (Information (InfB)) auf Seite des Targetsystems (CompB) beschrieben. Die verwendeten Beispiele sollen bloß das Prinzip verdeutlichen und stellen keine Einschränkung des Schutzzumfangs wie in den Ansprüchen definiert dar.

Für den Zeitpunkt, wann die plattformspezifische Information (InfB) berechnet wird, gibt es mehrere Möglichkeiten. Um möglichst viele Ressourcen auf dem Target (CompB) einzusparen, wäre es etwa denkbar, die Berechnung der Informationen (InfB) als eigenständiges Programm auf dem Target (CompB) laufen zu lassen und die plattformspezifischen Informationen (InfB) auf dem Hostsystem (CompA) abzuspeichern für die spätere Kommunikation mit dem Target (CompB). Eine alternative Möglichkeit wäre, die Berechnung der plattformspezifischen Informationen (InfB) direkt in den Applikationscode des Targets (CompB) aufzunehmen, was allerdings zusätzliche Bytes an Programmcode belegt.

Die Plattform-Eigenschaften, welche in den plattformspezifischen Informationen (InfB) beschrieben sind, enthalten an erster Stelle die Endianität (little endian oder big endian, d.h. die Anordnung der Bytes innerhalb eines Datenwortes). Das Wissen über die Endianität ist erforderlich, um weitere Informationen interpretieren zu können. Weitere Zusatzinformationen, die in den Informationen (InfB) enthalten sein können, sind beispielsweise die Wertebereiche der Grunddatentypen.

Die Endianität könnte etwa mit folgendem plattformunabhängigen Programmcode in ANSI C berechnet werden:

```
int e = 1;

if (*((char *)&e) == 1) endian=LITTLE; else endian=BIG;
```

Die Wertebereiche von integralen Grunddatentypen könnten mit folgendem Programmcode in ANSI C berechnet werden:

```
sizeof(char), sizeof(unsigned char),
```

```
sizeof(short), sizeof(unsigned short),  
  
sizeof(int),   sizeof(unsigned int),  
  
sizeof(long),  sizeof(unsigned long)
```

Bei den Grunddatentypen für Fließkommazahlen verhält sich die Berechnung ähnlich. Die Länge dieser Grunddatentypen kann analog berechnet werden:

```
sizeof(float), sizeof(double)
```

Zusätzlich jedoch haben diese Fließkomma-Datentypen eine interne Partitionierung in Vorzeichenbit, Exponentenfeld und Mantissenfeld. Ein Compiler für ANSI C ISO/IEC 9899 würde die Daten für diese Partitionierung bereits als Konstanten in der Datei `float.h` vordefiniert haben. Im Falle von älteren Compilern lassen sich diese Partitionierungsdaten allerdings auch automatisch berechnen.

Wien, den

PATENTANSPRÜCHE

1. Verfahren zum Austausch von Daten (DatAB) zwischen einem ersten Computersystem (CompA) und einem zweiten Computersystem (CompB), wobei die Daten (DatAB) auf dem ersten Computersystem (CompA) in einer ersten plattformspezifischen Datenrepräsentation und auf dem zweiten Computersystem (CompB) in einer zweiten plattformspezifischen Datenrepräsentation dargestellt werden, und die Konvertierung der Daten (DatAB) zwischen der ersten und zweiten plattformspezifischen Datenrepräsentation in beiden Richtungen auf dem ersten Computersystem (CompA) durchgeführt wird, wobei plattformspezifische Datenrepräsentation die binäre Repräsentation von Daten bezeichnet, die durch verwendeten Prozessor und/oder Compiler des jeweiligen Computersystems (CompA, CompB) bedingt ist

dadurch gekennzeichnet, dass

die auf dem ersten Computersystem (CompA) benötigte Information (InfB) über das zweite Computersystem (CompB), welche Information (InfB) zur Umwandlung der Daten (DatAB) für beide Richtungen des Datenaustausches notwendig ist, vollständig auf dem zweiten Computersystem (CompB) berechnet wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das Verfahren zur Berechnung der Information (InfB) auf dem zweiten Computersystem (CompB) in plattformunabhängiger Form realisiert ist.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die Information (InfB), welche von dem zweiten Computersystem (CompB) berechnet wird, ein Endian-Flag zur Beschreibung der Byte-Reihenfolge, in welcher Daten auf dem zweiten Computersystem (CompB) abgelegt werden, enthält.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass in den Daten (DatAB) die Werte von Objekten komplexer Datentypen (DatB) der Daten am Target, wie beispielsweise Strukturen, in flacher Form, d.h. als Sequenz der in den komplexen Datentypen enthaltenen Grunddatentypen, gespeichert werden.

5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass das erste Computersystem (CompA) und das zweite Computersystem (CompB) über einen elektronischen Datenkanal verbunden sind und automatisch vor der ersten Übermittlung der Daten (DatAB) die Information (InfB) von dem zweiten Computersystem (CompB) auf das erste Computersystem (CompA) übertragen wird.
6. Verfahren nach Anspruch 4 oder 5, dadurch gekennzeichnet, dass von dem zweiten Computersystem (CompB) die Verbindung zwischen den übertragenen Daten mit flachen Datenstruktur (DatAB) und den Objekten komplexer Datenstrukturen (DatB) über eine eigene Übersetzungstabelle (TabB) durchgeführt wird.
7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass die für die Transformation zwischen den übertragenen Daten (DatAB) und den Objekten komplexer Datentypen (DatB) sowie die Berechnung der Information (InfB) notwendigen Transformationsschritte vollständig als plattformspezifisches Verfahren spezifiziert sind.
8. Computersystem (CompB) für ein Verfahren nach einem der Ansprüche 1 bis 7, welches dazu eingerichtet ist, die auf einem ersten Computersystem (CompA) benötigte Information (InfB) über das zweite Computersystem (CompB), welche Information (InfB) zur Umwandlung der Daten (DatAB) für beide Richtungen des Datenaustausches notwendig ist (InfB), vollständig zu berechnen.
9. Computersystem (CompB) nach Anspruch 8, dadurch gekennzeichnet, dass das Verfahren zur Berechnung der Information (InfB) auf dem zweiten Computersystem (CompB) in plattformunabhängiger Form realisiert ist.
10. Computersystem (CompB) nach Anspruch 8 oder 9, dadurch gekennzeichnet, dass die für die Berechnung der Information (InfB) notwendigen Ressourcen nach erfolgter Berechnung der Information (InfB) anschließend für andere Aufgaben wiederverwendet werden können.
11. Computersystem nach einem der Ansprüche 8 bis 10, dadurch gekennzeichnet, dass die Information (InfB), welche von diesem Computersystem (CompB) berechnet wird, ein Endian-Flag zur Beschreibung der Byte-Reihenfolge, in welcher Daten auf diesem Computersystem (CompB) abgelegt werden, enthält.

12. Computersystem nach einem der Ansprüche 8 bis 11, dadurch gekennzeichnet, dass in den Daten (DatAB) die Werte von Objekten komplexer Datentypen, wie beispielsweise Strukturen, in flacher Form, d.h. als Sequenz der in den Objekten komplexer Datentypen enthaltenen Elementen von Grunddatentypen, gespeichert werden.
13. Testumgebung basierend auf dem Verfahren nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, dass auf einem zweiten Computersystem (CompB) das zu testende (Unter)Programm ausgeführt wird, wobei Textvektoren von dem ersten Computersystem (CompA) empfangen werden sowie Testergebnisse zurück an das erste Computersystem (CompA) übertragen werden, und wobei eine Speicherstruktur (MGlue) verwendet wird, welche die Anbindung der Test-Daten (DatB) an ein Programminterface (DatBInt) des zu testenden (Unter)Programmes realisiert.
14. Testumgebung nach Anspruch 13, dadurch gekennzeichnet, dass die Beschreibung und Initialisierung der Speicherstruktur (MGlue) mit den beiden Interfaces (DatB) und (DatBInt) auf Quellcodeebene als plattformunabhängiges Verfahren realisiert ist.

1/1

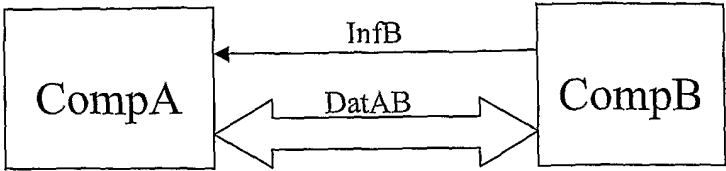


Fig. 1

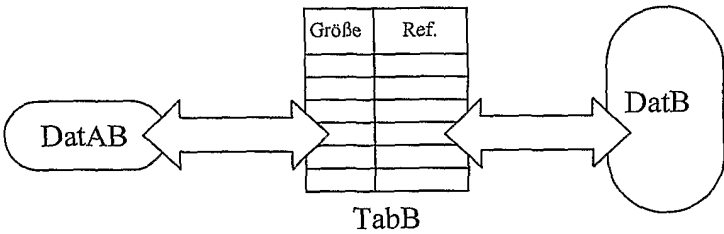


Fig. 2

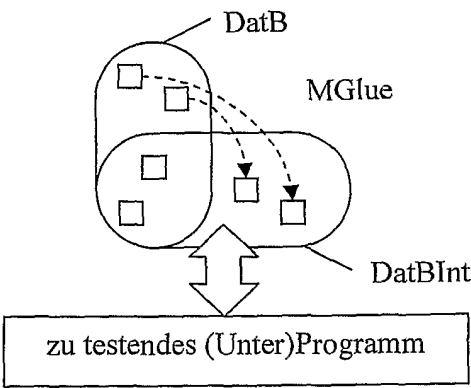


Fig. 3

INTERNATIONAL SEARCH REPORT

International application No

PCT/AT2005/000457

A. CLASSIFICATION OF SUBJECT MATTER
G06F9/45 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 772 413 B2 (KUZNETSOV EUGENE) 3 August 2004 (2004-08-03) column 5, line 5 - line 25 column 5, line 43 - line 50 column 12, line 54 - line 55 column 15, line 3 - line 13 -----	1-14
X	US 2003/179112 A1 (PARRY TRAVIS J ET AL) 25 September 2003 (2003-09-25) paragraph [0007] - paragraph [0010] paragraph [0021] - paragraph [0022] paragraph [0030] paragraph [0035] -----	1-14
X	US 2002/124118 A1 (COLLEY ADRIAN E ET AL) 5 September 2002 (2002-09-05) paragraph [0003] paragraph [0006] - paragraph [0009] paragraph [0034] - paragraph [0035] -----	1-14

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *Z* document member of the same patent family

Date of the actual completion of the international search

6 April 2006

Date of mailing of the international search report

20/04/2006

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Milasinovic, G

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/AT2005/000457

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6772413	B2	03-08-2004	NONE
US 2003179112	A1	25-09-2003	NONE
US 2002124118	A1	05-09-2002	EP 1405185 A2 07-04-2004 WO 02067118 A2 29-08-2002

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/AT2005/000457

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
G06F9/45 G06F9/44

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
G06F

Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

EPO-Internal, WPI Data, PAJ, IBM-TDB

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	US 6 772 413 B2 (KUZNETSOV EUGENE) 3. August 2004 (2004-08-03) Spalte 5, Zeile 5 - Zeile 25 Spalte 5, Zeile 43 - Zeile 50 Spalte 12, Zeile 54 - Zeile 55 Spalte 15, Zeile 3 - Zeile 13	1-14
X	US 2003/179112 A1 (PARRY TRAVIS J ET AL) 25. September 2003 (2003-09-25) Absatz [0007] - Absatz [0010] Absatz [0021] - Absatz [0022] Absatz [0030] Absatz [0035]	1-14
X	US 2002/124118 A1 (COLLEY ADRIAN E ET AL) 5. September 2002 (2002-09-05) Absatz [0003] Absatz [0006] - Absatz [0009] Absatz [0034] - Absatz [0035]	1-14



Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen



Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :

A Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist

E älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist

L Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)

O Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht

P Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist

T Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist

X Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden

Y Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist

Z Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche

6. April 2006

Absenddatum des internationalen Recherchenberichts

20/04/2006

Name und Postanschrift der Internationalen Recherchenbehörde
Europäisches Patentamt, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Bevollmächtigter Bediensteter

Milasinovic, G

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/AT2005/000457

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 6772413	B2	03-08-2004	KEINE
US 2003179112	A1	25-09-2003	KEINE
US 2002124118	A1	05-09-2002	EP 1405185 A2 07-04-2004 WO 02067118 A2 29-08-2002