



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 600 14 455 T2 2005.03.03**

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 323 136 B1**

(51) Int Cl.⁷: **G06T 17/10**

(21) Deutsches Aktenzeichen: **600 14 455.0**

(86) PCT-Aktenzeichen: **PCT/DK00/00027**

(96) Europäisches Aktenzeichen: **00 901 050.5**

(87) PCT-Veröffentlichungs-Nr.: **WO 00/43959**

(86) PCT-Anmeldetag: **21.01.2000**

(87) Veröffentlichungstag
der PCT-Anmeldung: **27.07.2000**

(97) Erstveröffentlichung durch das EPA: **02.07.2003**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **29.09.2004**

(47) Veröffentlichungstag im Patentblatt: **03.03.2005**

(30) Unionspriorität:

9599 22.01.1999 DK

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE**

(73) Patentinhaber:

Lego A/S, Billund, DK

(72) Erfinder:

**THOMSEN, Michael, DK-7183 Randbol, DK;
KROGH, Poul, DK-8680 Ry, DK; TIMCENKO, Olga,
DK-7190 Billund, DK**

(74) Vertreter:

Diehl, Glaeser, Hiltl & Partner, 22767 Hamburg

(54) Bezeichnung: **VIRTUELLE-REALITÄT MODELLIERUNG**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

Gebiet der Erfindung

[0001] Diese Erfindung betrifft das Gebiet von computergestützter Modellierung einer virtuellen Realität mittels vordefinierter, geometrischer Elemente.

Hintergrund der Erfindung

[0002] Computergestützte Modellierung einer virtuellen Realität ist die Aufgabe, ein Modell eines geometrischen Gegenstands zu erzeugen, das Modell zu interpretieren, das Modell zu manipulieren, und auf andere Weise ein Modell eines geometrischen Gegenstands in einem Computersystem zu handhaben.

[0003] Aus einem ersten Gesichtspunkt betrachtet, ist Modellierung einer virtuellen Realität dadurch ein interessantes Thema, dass sie ermöglicht, Ideen zu visualisieren, bevor sie tatsächlich in der realen Welt umgesetzt werden. Wenn das Modell virtueller Realität ausreichend einfach zu modifizieren ist, kann viel Zeit in dem Prozess zum Entwickeln und Verfeinern eines geometrischen Gegenstands, verglichen mit einer Situation, gespart werden, in der der gleiche Prozess in der realen Welt ausgeführt werden sollte. Eine einfache Aufgabe zum Anstreichen eines Gegenstands in der realen Welt kann einfach mehrere Stunden dauern, wohingegen der Computer eine neue Farbe zum Visualisieren eines Modells innerhalb von Millisekunden oder Sekunden aufbringen kann.

[0004] Unter einem zweiten Gesichtspunkt betrachtet, ist Modellierung einer virtuellen Realität interessant darin, dass sie das Erzeugen eines Modells eines Gegenstands, der in der realen Welt existiert, und über einen Computer Visualisieren und z.B. Manipulieren des Modells in irgendeinem Sinne ermöglicht. Daher kann das Modell des Gegenstands in der realen Welt zu unterschiedlichen Zwecken, z.B. für fortgeschrittene Dokumentationszwecke, gespeichert werden.

[0005] Obwohl eine große Menge möglicher Anwendungen für computergestützte Modellierung virtueller Realität existieren, besteht eine spezielle Anwendung darin, die Modellierung virtueller Realität für die Unterhaltung oder Erziehung zu verwenden. Aus der realen Welt sind, lange vor der Entwicklung des elektronischen Computers, verschiedene Typen von Modellierungskonzepten bekannt. Insbesondere waren und sind Konzepte sehr beliebt, die modulare oder halbmodulare Konzepte verwenden. Typischerweise liefern diese Konzepte einen Satz vorgefertigter Elemente, die miteinander in gewissen, vorbestimmten Weisen gemäß den Modulen der vorgefertigten Elemente verbunden werden können. Die vorgefertigten Elemente ähneln gut bekannten Gegen-

ständen, die an eine spezifische Modellierungsaufgabe angepasst sind. So können z.B. beim Bauen eines Modells eines Hauses die Elemente Mauersteinen, Dachziegeln, Türen und Fenstern ähneln. Das Ziel des Auswählens der Elemente auf diese Weise besteht darin, dass die mit dem Bau eines Modells eines Hauses verknüpfte Arbeit bedeutend, verglichen mit einer Situation, reduziert wird, in der alle Details des Hauses jedes Mal definiert werden müssen, wenn ein neues Modell hergestellt werden sollte. Die vollständige Freiheit beim Bau eines Hauses oder anderen Gegenstands wird jedoch gegen die Einfachheit des Baus des Modells eingetauscht.

[0006] Dieser Ansatz, vordefinierte Elemente zu haben, ist im technischen Gebiet von computergestützter Modellierung virtueller Realität gut bekannt. Ferner ist das Konzept gut bekannt, modulare Elemente zu haben, die miteinander verbunden werden können. Wenn es jedoch zur Darstellung solcher Modelle in einem Computer kommt, wird die Zweckdienlichkeit des Konzepts, vordefinierte Elemente und modulare Systeme zu haben, nicht vollständig im Stand der Technik erreicht.

[0007] Solange Computer für computergestütztes Design und Modellierung verwendet worden sind, ist die Aufgabe, die den größten Berechnungseinsatz erfordert, Visualisierung des Modells einschließlich der Berechnung gewesen, wie das Modell aussehen sollte. Einer der Gründe dafür ist es, dass die Komplexität und Verfeinerungen der Modelle virtueller Realität erweitert worden sind, um der neuesten, verfügbaren Computertechnik und Rechenleistung zu folgen.

[0008] Wenn es um die Übertragung und den Austausch von ein Modell darstellenden Daten geht, erscheinen die Mängel der Komplexität und Verfeinerungen der Modelle virtueller Realität sehr deutlich. Nun wird irgendein Typ einer externen Einheit in Form einer Speichereinheit oder eines Computernetzwerks an den Computer angeschlossen und erweitert so den Signalweg für das Speichern, Laden und/oder Übertragen, Empfangen eines Modells. Typischerweise haben solche Signalwege eine relativ niedrige Bandbreite, verglichen mit Signalwegen innerhalb eines einzelnen Computers. Daher wird ein effizientes Schema zum Darstellen eines Modells benötigt.

Stand der Technik

[0009] Ein Schema zum Darstellen eines Modells eines geometrischen Gegenstands ist die gut bekannte "Virtual Reality Modelling Language" [Modellierungssprache virtueller Realität], auch bekannt als "VRML". Dieses Schema wird oft im Internet verwendet. "VRML" basiert auf irgendwelchen Typen von Polygonen, die durch einen Satz von Parametern de-

finiert werden. Alle Modelle basieren auf diesen Polygonen durch Spezifizieren der Parameter, z.B. Definieren der Position von Ecken der Polygone in einem dreidimensionalen Raum. Es ist möglich, die Position, Ausrichtung und den Maßstab einer Gruppe von Polygonen mittels einer gemeinsamen Umwandlung zu definieren.

[0010] Mittels dieses Modellierungsschemas können alle möglichen Modellierungsaufgaben, wie gewünscht, modelliert werden, da das Schema den Benutzer in keiner Weise einschränkt. Dies ist tatsächlich auch einer der Nachteile des Schemas, da jedes der Polygone durch eine extensive Darstellung unterstützt wird. Auf einem Computer weist diese Darstellung Ziffern des Typs Real für jeden Parameter in dem Modell auf. Ferner wird aufgrund der Beschaffenheit der meisten Modellierungssparadigmen eine überwältigende Anzahl von Polygonen benötigt – folglich wird auch eine überwältigende Anzahl von Parametern des Typs Real benötigt. Bei Erwägung z.B. einer gekrümmten Oberfläche kann eine solche Oberfläche in einem sehr groben Modell mittels eines einzelnen Polygons modelliert werden; wenn jedoch ein feines Modell benötigt wird, muss diese gekrümmte Oberfläche aus einer Anzahl von Polygonen bestehen, die mindestens so viele Polygone aufweist, dass die Oberfläche mit ausreichender Glätte erscheinen wird.

[0011] Wenn es um die Manipulierung eines VRML-Modells geht, muss die geometrische Beschreibung mit einer Meta-Beschreibung erweitert werden, die jede der möglichen Manipulierungen definiert. Wenn ein VRML-Modell nicht zuerst mit einer solchen Meta-Beschreibung geliefert wird, sind daher keine Manipulierungen dieses Modells möglich. Dies ist ein Ergebnis des völlig freien Modellierungsschemas.

[0012] Es existiert daher ein Problem im Stand der Technik, dass durch computergestützte Modellierung erhaltene Modelle eine riesige Menge von Speicherkapazität erfordern.

[0013] Im Stand der Technik existieren verschiedene Typen von Komprimierungsschemata; solche Komprimierungsschemata erfordern jedoch einen großen Block von Daten, an dem gearbeitet wird, um die Daten effizient zu komprimieren. Darüber hinaus erfordern solche Komprimierungsschemata einen relativ großen Zeitschlitz beim Durchführen von Komprimierung und Dekomprimierung. Typischerweise ist dies kritisch, wenn interaktive Modellierungsaufgaben auszuführen sind, die Übertragung von Modellen, z.B. über ein Computernetz, aufweisen. Es besteht daher ein Bedarf an einer Darstellung geometrischer Modelle, die schnell zu erzeugen und zu interpretieren ist, und die keine zusätzlichen Komprimierungsschemata erfordert.

Zusammenfassung der Erfindung

[0014] Eine erste Aufgabe der Erfindung besteht in der Schaffung einer kompakten Darstellung eines Modells eines geometrischen Gegenstands.

[0015] Eine zweite Aufgabe der Erfindung besteht in der Schaffung einer kompakten Darstellung eines Modells eines geometrischen Gegenstands, die eine Modellierung komplexer Gegenstände ermöglicht, welche bewegliche und/oder drehbare Teile aufweisen.

[0016] Eine dritte Aufgabe der Erfindung besteht in der Schaffung einer Modellstruktur und einer Darstellung der Modellstruktur, die sehr schnell erzeugt und interpretiert werden kann.

[0017] Eine vierte Aufgabe der Erfindung besteht in der Schaffung einer Modellstruktur und einer Darstellung der Modellstruktur, die zweckdienlich zum Darstellen von Gegenständen ist, welche aus modularen Spielbaublöcken oder Elementen bestehen.

[0018] Eine fünfte Aufgabe der Erfindung besteht in der Schaffung einer Darstellung einer Modellstruktur, die zweckdienlich für eine Verteilung über ein Computernetz ist, im Sinne relativ niedriger Bandbreitenanforderungen und niedriger Verarbeitungszeit, wenn das Modell Visualisierung ausgesetzt wird, erzeugt oder manipuliert wird.

[0019] Eine sechste Aufgabe der Erfindung besteht in der Schaffung einer Modellstruktur und einer Darstellung der Modellstruktur, die keine zusätzliche Meta-Beschreibung eines geometrischen Gegenstands erfordern, wenn das Modell einer Manipulierung unterzogen wird.

[0020] Der Erfindung zufolge werden die in den obigen Absätzen genannten Aufgaben innerhalb des Umfangs der Erfindung erfüllt, wie er durch die Patentansprüche definiert ist. Insbesondere, wenn ein Modell eines geometrischen Gegenstands aufweist: Bits in einer ersten Datenstruktur zum Identifizieren eines ersten Satzes von Elementen aus einer Sammlung von Darstellungen geometrischer Formen für eine Sammlung von Elementen und zum Darstellen von Positionen der Elemente mittels ganzzahliger Koordinaten in einem ersten Koordinatensystem; Bits in einer zweiten Datenstruktur, die zum Darstellen einer räumlichen Umwandlung eines zweiten Koordinatensystems in Bezug zu dem ersten Koordinatensystem codiert wird.

[0021] Dadurch wird eine sehr kompakte Darstellung eines geometrischen Gegenstands erhalten, während das Modell eine Modellierung komplexer, geometrischer Strukturen mit beweglichen Teilen ermöglicht. Tatsächlich unterscheidet das Modell zwei-

schen starren Teilen und beschränkt die Position der Elemente innerhalb solcher starren Teile zum Erhalten einer kompakten Darstellung. Wenn jedoch eine räumliche Umwandlung solcher starren Teile oder Segmente dargestellt werden soll, kann eine nicht-beschränkte oder weniger beschränkte Darstellung des gegenseitigen Verhältnisses angewendet werden. Dadurch wird auch eine relativ flexible Modelldarstellung bereitgestellt.

[0022] Aufgrund der ganzzahligen Darstellung von Positionen der Elemente ist es möglich, das Modell wieder und wieder auf einem Computer zu verarbeiten, ohne Abrundfehler innerhalb der jeweiligen Koordinatensysteme einzubringen. Daher werden eine große Menge von Elementen innerhalb eines Segments einschließende Modelle nicht aufgrund solcher Abrundfehler verzerrt.

[0023] Aufgrund der Tatsache, dass das Modell nur Verweise auf die geometrischen Darstellungen der Elemente aufweist, kann das Modell mit einem ausgewählten Satz von Elementen interpretiert werden, ohne das Modell zu modifizieren. Dadurch kann ein spezielles, graphisches Format ausgewählt werden, um zweckdienlich auf einem speziellen Computer interpretiert zu werden, z.B. in dem Sinne, eine ausreichend schnelle Bilderzeugungszeit zu liefern, wenn das Modell anzuzeigen ist.

[0024] Aufgrund der ganzzahligen, beschränkten Darstellung der Positionen von Elementen innerhalb des jeweiligen Koordinatensystems werden mögliche Positionen der Elemente implizit gegeben, d.h. Drehungen, freie Verschiebungen etc. sind nicht innerhalb eines Koordinatensystems möglich.

[0025] Da die kompakte Darstellung des Modells durch Beschränkungen auf den Elementen erhalten wird, wird keine zusätzliche Komprimierung benötigt, und die Modellstruktur kann daher sequentiell erzeugt und sequentiell durch Schreiben/Lesen einer Abfolge von Feststellungen erzeugt werden, die zum Darstellen des Modells verwendet werden.

Kurze Beschreibung der Zeichnungen

[0026] Die Erfindung soll im folgenden vollständiger in Verbindung mit einer bevorzugten Ausführungsform und unter Bezugnahme auf die Zeichnungen erklärt werden, in denen:

[0027] Fig. 1 die hierarchische Struktur eines Modells eines geometrischen Gegenstands gemäß einer Modellierungssprache virtueller Realität des Standes der Technik zeigt;

[0028] Fig. 2a einen ersten Gegenstand zeigt, der einer Modellierung gemäß der Erfindung unterzogen wird;

[0029] Fig. 2b einen zweiten Gegenstand zeigt, der einer Modellierung gemäß der Erfindung unterzogen wird;

[0030] Fig. 3 die hierarchische Struktur eines Modells eines geometrischen Gegenstands gemäß der Erfindung zeigt;

[0031] Fig. 4 zwei Familien von Elementen zeigt;

[0032] Fig. 5 einen ersten Schritt in einem Prozess zum Aufbauen eines Modells eines dritten, geometrischen Gegenstands zeigt;

[0033] Fig. 6 einen zweiten Schritt in einem Prozess zum Aufbauen eines Modells des dritten, geometrischen Gegenstands zeigt;

[0034] Fig. 7 einen dritten Schritt in einem Prozess zum Aufbauen eines Modells des dritten, geometrischen Gegenstands zeigt;

[0035] Fig. 8 einen vierten, gemäß der Erfindung zu modellierenden Gegenstand zeigt;

[0036] Fig. 9 ein Clientcomputersystem zum Entwerfen, Speichern, Manipulieren und Teilen geometrischer Modelle gemäß der Erfindung zeigt;

[0037] Fig. 10 ein Servercomputersystem zum Behandeln geometrischer Konstruktionen gemäß der Erfindung zeigt;

[0038] Fig. 11 ein Ablaufdiagramm für ein Verfahren zum gleichzeitigen Aufbauen von Segmenten und Modellen zeigt, während die Segmente und Modelle einer Manipulierung unterzogen werden;

[0039] Fig. 12 ein Ablaufdiagramm für ein Verfahren zum Zugreifen auf und Teilen einer gemeinsamen Welt zeigt, die für mehrere Clients zugänglich ist, und

[0040] Fig. 13 ein Ablaufdiagramm für ein Verfahren zum Übertragen von Modellinformationen zeigt.

Ausführliche Beschreibung einer bevorzugten Ausführungsform

[0041] Fig. 1 zeigt die hierarchische Struktur eines Modells eines geometrischen Gegenstands gemäß einer Modellierungssprache virtueller Realität des Standes der Technik. Eine solche Modellierungssprache virtueller Realität ist einer Fachperson im technischen Gebiet als VRML bekannt. Die Grundelemente von VRML sind zweidimensionale, nicht-konkave Polygone, die durch ihre Koordinaten in einem dreidimensionalen, rechteckigen Koordinatensystem definiert sind. Wenn die Achsen eines dreidimensionalen, rechteckigen Koordinatensystems als die x-Achse, y-Achse bzw. z-Achse bezeichnet werden, wird

ein zweidimensionales, konvexes Polygon, das ein Rechteck sein kann, durch die Koordinaten der vier Ecken des Rechtecks definiert. Das heißt, vier Sätze von (x, y, z) Koordinaten. Aufgrund der polygonorientierten Struktur von VRML muss ein Designer, der ein Modell virtueller Realität eines fiktiven oder existierenden Gegenstands der realen Welt entwirft, den Gegenstand in Polygone unterteilen. Ein genaues und feines VRML-Modell eines Gegenstands mit einer gekrümmten und glatten Oberfläche muss daher aus einer riesigen Anzahl von Polygonen zusammengesetzt werden, um das Modell mit einer feinen und glatten Oberfläche erscheinen zu lassen. Es kann jedoch ein größeres VRML-Modell des Gegenstands aus einer kleineren Anzahl von Polygonen zusammengesetzt werden, wodurch die Komplexität des Modells reduziert wird. Damit der Designer einen jeglichen Gegenstand modellieren kann, muss es möglich sein, die zweidimensionalen Polygone frei in dem dreidimensionalen Koordinatensystem (im folgenden ist dies auch als der "dreidimensionale Raum" oder nur "Raum" bezeichnet) zu platzieren. In einem Computer werden die Koordinaten der Polygone durch Zahlen des sogenannten Typs "Real", d.h. Gleitkommazahlen, dargestellt.

[0042] Zum Behandeln selbst recht einfacher Modelle werden die Polygone in einer nach unten umgedrehten Baumstrukturhierarchie **101** angeordnet. Jeder Zweig **102** des Baums ist an einen Knotenpunkt angeschlossen, um die Hierarchie zu definieren. Jeder Knotenpunkt auf einer gegebenen Ebene der Hierarchie ist ein gemeinsamer Bezugspunkt für andere Knotenpunkte oder Polygone auf einer niedrigeren Ebene der Hierarchie. Die Hierarchie wird nach unten durch Polygone beendet, die die niedrigste Ebene der Hierarchie darstellen.

[0043] Die Polygone **103** und **104** werden durch die Koordinaten x, y, z definiert, die sich auf denselben Knotenpunkt **107** mittels der Schnittflächen **105** und **106** beziehen. Auf den Knotenpunkt **107** kann auch durch andere Knotenpunkte **108** mit Polygonen (nicht gezeigt) Bezug genommen werden. Mittels des Schnittstellenteils **109** des Knotenpunkts **107** ist es möglich, die Polygone und Knotenpunkte durch gemeinsame Umwandlung auf eine niedrigere Ebene der Hierarchie zu übersetzen, zu drehen und zu skalieren, die direkt oder indirekt mit dem Knotenpunkt **107** verbunden sind. Es ist daher möglich, einen Satz von Polygonen zu gruppieren und diese Gruppe mittels einer einzigen Umwandlung zu behandeln, die eine Übersetzung, Drehung und Skalierung einschließt.

[0044] Alle in der Definition der Polygone und Knotenpunkte verwendeten Parameter, das heißt die Koordinaten, Übersetzungen, Drehungen und Skalierungen, sind von dem Typ Real und sind daher als Gleitkommazahlen in einem Computer dargestellt.

Diese Darstellung wird gewählt, um eine jegliche Modellierungsaufgabe zu handhaben. Diese Darstellung erfordert jedoch eine riesige Speicherkapazität und ist tatsächlich ein Hindernis, wenn es um die Speicherung und Übertragung von Modellen geht – einschließlich Echtzeitmanipulierungen von Modellen.

[0045] Fig. 2a zeigt einen einfachen Gegenstand, der ein erstes und ein zweites Segment aufweist, die einer Modellierung gemäß der Erfindung unterzogen werden. Ein Modell des Gegenstands umfasst zwei Segmente **201** und **202**, die jeweils Elemente **203**, **204**, **205** bzw. **206**, **207** aufweisen. Die Achsen x_1, y_1 und z_1 erstrecken sich über ein Koordinatensystem **209** und gehören zum Segment **201**, und die Achsen x_2, y_2 und z_2 erstrecken sich über ein Koordinatensystem **211** und gehören zu dem Segment **202**.

[0046] Es wird festgestellt, dass die Elemente **203**, **204** und **205** des ersten Segments nur gemäß den Modulen **208** des Koordinatensystems **209** platziert werden. Gleichmaßen werden die Elemente **206** und **207** des zweiten Segments nur gemäß den Modulen **210** des Koordinatensystems **211** platziert.

[0047] Der Zweck der Unterteilung des Gegenstands in zwei Segmente wird aus dem Folgenden deutlich werden.

[0048] Fig. 2b zeigt den in Fig. 2a gezeigten, einfachen Gegenstand, wobei das zweite Segment in Bezug zum ersten Segment gedreht ist. In dieser Darstellung ist es klar, dass zwei verschiedene und nicht-zusammenfallende Koordinatensysteme existieren, die durch die Achsen x_1, y_1, z_1 bzw. x_2, y_2, z_2 definiert sind. Elemente innerhalb des Segments **201** können nur gemäß den Modulen **208** des Gitters **209** platziert werden. Dementsprechend können Elemente innerhalb des Segments **202** nur gemäß den Modulen **210** des Gitters **211** platziert werden. Die Elemente innerhalb eines Segments werden dadurch in Bezug zu ihrer Position eingeschränkt. Die Gitter sind als zweidimensionale Gitter gezeigt, aber typischerweise werden die Gitter dreidimensionale Gitter sein, was die Anordnung der Elemente in drei Dimensionen ermöglicht. Ferner sind die Ausrichtungen der Elemente innerhalb eines Segments auf Drehung in Schritten von 90 Grad in Bezug zu den drei Achsen (x_1, y_1, z_1 bzw. x_2, y_2, z_2) der Segmente beschränkt.

[0049] Das Segment **202** wird gemäß dem Gitter **209** des Segments **201** platziert, aber es wird eine Drehung des Segments **202** in Bezug zur Achse z_1 zugelassen. Das einfache Modell wird nun erweitert, und es ist möglich, fortgeschrittene Gegenstände mit einer Abfolge rotierbarer Segmente zu modellieren. Es soll festgestellt werden, dass trotz der Tatsache, dass dieses Drehmerkmal in das einfache Modell eingebracht wird, die Darstellung nur mit einer einzigen Zahl erweitert wird, die den Drehwinkel darstellt, da

die Position des Segments **202** auf das Gitter **209** beschränkt ist, und da die Elemente **204** und **205** auch auf ein Gitter, d.h. das Gitter **211**, beschränkt sind.

[0050] Durch graduelles Hinzufügen von mehr Elementen zu den Segmenten und Verbinden neuer Segmente mit Elementen mit den existierenden Segmenten **201** und **202** ist es möglich, das Modell zu erweitern. Tatsächlich können sehr komplexe, geometrische Gegenstände auf diese Weise trotz der schwerwiegenden Begrenzungen in Form von Position und Ausrichtungsbeschränkungen modelliert werden. Diese schwerwiegenden Beschränkungen sind der Schlüssel zu einer sehr kompakten Darstellung eines Modells, das einen komplexen, geometrischen Gegenstand modelliert.

[0051] Insbesondere wird ein Modell gemäß der Erfindung den Modellen virtueller Realität des Standes der Technik überlegen sein, wenn ein einer Modellierung unterzogener Gegenstand relativ wenige, starre Teile aufweist, die sich in Bezug zueinander bewegen oder drehen können, und wenn die starren Teile mit vielen Elementen modelliert werden müssen. Ein Beispiel eines solchen Gegenstands kann z.B. eine Brücke mit zwei Brückenflügeln sein. Ein Modell dieses Gegenstands kann nur zwei bewegliche, starre Teile aufweisen, das heißt, die Brückenflügel, aber die jeweiligen Brückenflügel und die Brückenflügel haltenden Strukturen können starre Strukturen, z.B. Zäune, Fußwege und Türme, aufweisen, die mittels vieler Elemente zu modellieren sind.

[0052] Im Folgenden ist eine Beschreibung darüber angeführt, wie die Darstellung gemäß der Erfindung tatsächlich erzeugt wird und wie andere Merkmale dem erfindungsgemäßen Modell hinzugefügt werden können.

[0053] Fig. 3 zeigt die hierarchische Struktur eines Modells eines geometrischen Gegenstands gemäß der Erfindung.

[0054] Das Ziel der Hierarchie besteht nicht nur in der Schaffung logischer Gruppierungen geometrischer Grundelemente, die einer gemeinsamen Umwandlung unterzogen werden können, sondern auch in der Hinzufügung verschiedener Typen von Beschränkungen auf Parameter, die in den jeweiligen Ebenen der Hierarchie verwendet werden, wodurch es ermöglicht wird, eine zweckdienliche Darstellung der Parameter angesichts der Beschränkungen und im Sinne der Erzeugung einer kompakten Darstellung des Modells auszuwählen.

[0055] Es sollte festgestellt werden, dass die Hierarchie das räumliche Verhältnis zwischen Elementen betrifft – nicht die tatsächliche Darstellung geometrischer Formen für die Elemente. Die tatsächliche Darstellung der Form von Elementen kann ausgewählt

werden, um in einem jeglichen, geeigneten Format, wie z.B. VRML oder 3DMF, vorzuliegen.

[0056] Ein Modell gemäß einer Ausführungsform der Erfindung kann mittels einer Hierarchie **316** organisiert werden, die vier Grundebenen aufweist: "Weltebene", "Modellebene", "Segmentebene" und "Elementebene".

[0057] Auf der Weltebene ist ein Weltblock **315** definiert, der eine virtuelle Welt darstellt. Die primäre Aufgabe des Weltblocks ist es, einen dreidimensionalen Raum mit einem Ursprung zu definieren, wobei der Ursprung ein gemeinsamer Bezugspunkt für die unteren Schichten der Hierarchie darstellt.

[0058] Auf der Modellebene kann eine Anzahl von Modellblöcken **311**, **314** vorliegen, die Modelle in der virtuellen Welt darstellen. Modellblöcke umfassen Felder **312** und **313** und Blöcke in Form von Segmentblöcken **309**, **307**. In einer bevorzugten Ausführungsform sind die Felder in dem Modellblock "Übersetzen", "Drehen" und "Skalieren". Die Felder definieren die Position, Ausrichtung und die Größe des Modells in der virtuellen Welt. Die Position, Ausrichtung und die Größe des Modells können frei definiert werden, d.h. das Modell ist nicht durch irgendwelche den Feldern auferlegte Beschränkungen begrenzt. In einem Computer können die Felder Übersetzen, Drehen und Skalieren mittels von Zahlen des Typs Real, d.h. eine Gleitkommadarstellung, dargestellt werden.

[0059] Auf der Segmentebene kann eine Anzahl von Segmentblöcken **307**, **309** vorliegen, die Segmente in einem Modell darstellen. Die Segmentblöcke umfassen Felder **308**, **310** und Blöcke **301**, **302** in Form von Elementblöcken und/oder untergeordneten Segmentblöcken **305**. Die Felder **310** in dem Segmentblock sind Versatz, Drehen und Achse. Die Felder definieren die Position einer Achse innerhalb des Segments, Drehung im Sinne eines Winkels um die Achse und die Ausrichtung der Achse.

[0060] Die Position der Achse definiert daher eine Verbindung, die zum Verbinden des Segmentblocks **305** mit einem untergeordneten Segmentblock **305** verwendet wird.

[0061] Die Position der Achse innerhalb des Segments ist auf Positionen begrenzt, die durch die Beschränkungen des Segments zugelassen werden. Solche Beschränkungen können mittels eines dreidimensionalen Koordinatensystems festgelegt werden, in dem Elemente in modularen Schritten positioniert werden können. Die Winkeldrehung um die Achse ist in einer Grundausführungsform der Erfindung nicht beschränkt. Die Ausrichtung der Achse ist auf Schritte von 90 Grad um drei orthogonale Achsen des Koordinatensystems beschränkt, siehe z.B. Fig. 2b.

[0062] Das Feld Versatz kann drei Parameter aufweisen, die mittels Zahlen des Typs Ganzzahl in einem Computer dargestellt werden können. Alternativ kann ein Feld Übersetzung mit drei Parametern des Typs Ganzzahl verwendet werden.

[0063] Ferner kann das Feld Drehen drei Parameter aufweisen, die eine Gruppe von drei Zahlen des Typs Real umfassen, und eine Achse kann mittels einer Gruppe von drei Zahlen des Typs Kurzganzzahl dargestellt werden oder alternativ nur durch Anzeigen einer der drei Achsen in dem Koordinatensystem mittels einzelner Bits dargestellt werden.

[0064] Auf der Elementebene kann eine Anzahl von Elementblöcken **301**, **302** vorliegen, die Elemente innerhalb eines Segments darstellen. Elementblöcke weisen keine untergeordneten Blöcke, sondern nur Felder **303**, **304** auf. Daher beenden Elementblöcke die Hierarchie nach unten. Die Felder in dem Elementblock sind Versatz, Ausrichtung und Formverweis. Bezüglich Position und Ausrichtung sind Elemente die am stärksten begrenzten Teile eines Modells. Das Feld Versatz ist ein dreidimensionaler Versatz, der auf diskrete Werte beschränkt ist. Wenn die Beschränkungen mittels eines dreidimensionalen Koordinatensystems festgelegt sind, in dem Elemente in modularen Schritten positioniert werden können, kann das Feld Versatz mittels einer Gruppe von Zahlen des Typs Ganzzahl dargestellt werden. Ferner ist das Feld Ausrichtung in der gleichen Weise wie das Feld Achse in den Segmentblöcke beschränkt, d.h. auf Schritte von 90 Grad um drei orthogonale Achsen des Koordinatensystems beschränkt. Daher kann die Achse mittels einer Gruppe von drei Zahlen des Typs Kurzganzzahl dargestellt werden oder alternativ nur durch Anzeigen einer von drei Achsen in dem Koordinatensystem dargestellt werden.

[0065] Mittels des Felds Formverweis schließen Elemente einen Verweis auf eine geometrische Beschreibung ein, die interpretiert und für Visualisierungszwecke verwendet werden kann. Der Formverweis ist vorzugsweise eine Bezugsziffer, die in einer Nachfrage an eine eine geometrische Beschreibung der einzelnen Elemente aufweisende Datenbank verwendet werden kann. Typischerweise wird ein Modell auf einem Satz verschiedener Elementtypen basieren. Die Datenbank weist eine geometrische Beschreibung der einzelnen Elemente gemäß einer jeglichen bevorzugten, geometrischen Beschreibung auf. Bevorzugte, geometrische Beschreibungen können VRML, 3DMF etc. sein. Ferner kann in der Datenbank eine Anzahl verschiedener Typen geometrischer Beschreibungen für jedes der Elemente gespeichert sein. Es ist dadurch möglich, eine geometrische Beschreibung auszuwählen, die in irgendeinem Sinne geeignet sein kann, z.B. im Sinne der Schaffung eines guten Austauschs zwischen Erhalten einer vernünftig schnellen Visualisierung des Mo-

dells und Erhalten einer ausreichend hohen Auflösung der Visualisierung. Es sollte hervorgehoben werden, dass nur die Position der Elemente auf ganzzahlige Koordinaten beschränkt ist – nicht die Koordinaten anderer Typen von Parametern, die zum Darstellen der Formen der Elemente verwendet werden.

[0066] Die Elementblöcke **301**, **302** können auch andere Felder aufweisen, die einen Verweis auf ein Material liefern, aus dem das Element besteht; einen Verweis auf eine Dekoration oder Struktur. Solche Felder können als "Materialverweis" und "Dekorationsversion" bezeichnet werden. Wie für die geometrische Beschreibung kann eine Beschreibung von Materialien und Dekoration oder Struktur in einer Datenbank in einem geeigneten Format gespeichert werden.

[0067] Daher basiert die Wirkung von Beschränkung einer Modellstruktur gemäß der Erfindung, verglichen z.B. mit der VRML-Modellstruktur, auf der Tatsache, dass bei den meisten Modellierungsaufgaben die Strukturen auf der niedrigsten Ebene der Hierarchie am häufigsten verwendet werden. Durch Beschränken genau dieser Strukturen der niedrigsten Ebenen und Anpassen einer Darstellung dieser Strukturen, um nur gerade ausreichend zu sein, ist es daher möglich, eine sehr kompakte Modelldarstellung zu erhalten. Wenn die Hierarchie insgesamt betrachtet wird, ermöglicht der Modellblock eine freie Umwandlung, ermöglicht der Segmentblock eine teilweise beschränkte Umwandlung, während der Elementblock nur eine sehr beschränkte Umwandlung zulässt.

[0068] Nach Definieren einer möglichen, geometrischen Interpretation von Elementen und Segmenten und nach Definieren einer Hierarchie, in der Elemente, Segmente, Modelle und Welten organisiert sind, besteht das Thema nun darin, eine bevorzugte Ausführungsform einer entsprechenden Codestruktur zu definieren, die zum Darstellen eines tatsächlichen Modells verwendet wird.

[0069] Es sollte festgestellt werden, dass eine Grundausführungsform der Erfindung ein einziges Segment aufweisen kann, mit Elementen, die durch Koordinaten in einem beschränkten Koordinatensystem erzwungene Positionen in einer freien Welt, d.h. in einem willkürlichen Koordinatensystem, aufweisen.

Codestruktur

[0070] Die Codestruktur umfasst zwei Typen von Grundelementen: Blöcke und Felder. Ein Block ist durch einen Blocknamen und Klammern {} definiert, die seinen Inhalt umschließen. Ein Block kann eine Anzahl von Feldern und andere untergeordnete Blöcke einschließen. Ein Feld ist durch einen Feldnamen

und eine Anzahl von dem Feldnamen nachfolgenden Parametern definiert.

Blöcke und Felder

[0071] Im Folgenden werden Blöcke mit ihrem Blocknamen, gefolgt durch ihren Kontext in Klammern, geschrieben. Kontext ist der Block, in dem sie existieren können; die obere Ebene wird als Datei bezeichnet. Der Ausdruck Datei wird als ein Verweis z.B. auf eine elektronische Datei verwendet, die auf einem computerlesbaren Medium, wie z.B. im Speicher eines Computers, auf einer CD-ROM, auf einer digitalen Videodisk (DVD), auf einer Floppydisk, auf einer Festplatte, in einem flüchtigen Speicher (z.B. Direktzugriffsspeicher), etc., gespeichert ist.

[0072] Jeder Block hat eine kurze Beschreibung. Zum Reduzieren der Komplexität dieser Definition werden untergeordnete Blöcke nur mit ihrem Blocknamen geschrieben.

Welt (Datei)

[0073] Die strukturell wichtigen Grundelemente sind die Blöcke in der Hierarchie, d.h. Weltblock, Modellblock, Segmentblock und Elementblock. Der Weltblock bezieht sich auf eine Datei "Datei", die eine Beschreibung der Welt enthält.

[0074] Die Welt ist durch die folgende Syntax definiert:

```
Welt:
{
Optionen {...};
Modell {...};
}
```

[0075] In einer virtuellen Welt gemäß der Erfindung kann eine Anzahl von Modellen existieren, die in der Anweisung "Modell {...}" enthalten ist. Es ist jedoch möglich, andere Merkmale in der Welt festzulegen, die gemäß der Erfindung beschrieben oder nicht beschrieben werden können. Solche Merkmale können Hintergrund, Ton, vordefinierte Ereignisse etc. sein.

[0076] Ansonsten wird der Weltblock zum Festlegen eines gemeinsamen Verweises für die Modelle (und andere Merkmale) in der Welt verwendet. Vorzugsweise liegt dieser Verweis in Form eines Koordinatensystems vor.

Modell (Welt)

[0077] Der Modellblock kann eine Anzahl von Modellen oder eine Anzahl von Segmentblöcken enthalten. Die Felder in den Modellblöcken sind: "Drehen", "Übersetzen" und "Skalieren":

```
Modell:
{
```

```
Drehen [Real; Real; Real];
Übersetzen [Real; Real; Real];
Skalieren [Real; Real; Real];
Segment {...};
}
```

[0078] Es sollte erwähnt werden, dass alle Dreh- und Ausrichtungparameter auf dem gut bekannten Dreifingersystem von Koordinaten basieren, das heißt, der Daumen folgt positiven Werten einer der Achsen, und positive Winkelwerte werden in der Richtung der anderen Finger gezählt.

[0079] Das Feld Drehen weist drei Parameter auf, die bestimmen, wie das Modell in der Welt ausgerichtet werden soll. Vorzugsweise sind die Parameter als gut bekannte Euler-Winkel spezifiziert, d.h. zuerst wird das Modell gemäß einem ersten der drei Parameter gedreht, die einen Winkel um eine erste Achse in einem rechteckigen Koordinatensystem spezifizieren; zweitens wird das Modell gemäß dem zweiten Parameter gedreht, das einen Winkel um eine zweite Achse spezifiziert, welche Ausrichtung aufgrund der ersten Drehung des Modells um die erste Achse geändert wird. Diese Verfahrensweise wird verfolgt, bis das Modell, wie gewünscht, um die drei Parameter in dem Feld Drehen gedreht worden ist. Es können jedoch andere Verfahrensweisen zum Spezifizieren der Ausrichtung und/oder Drehung eines Gegenstands angewendet werden.

[0080] In der Syntax für das Feld Drehen in der obigen Definition des Modells sind die Parameter vom Typ "Real"; es können jedoch andere Typen, wie z.B. Ganzzahl, wobei ein gegebener Auswahltyp z.B. die möglichen Werte [0, 10, 20, ..., 350] Grad aufweist, oder andere Typen verwendet werden.

[0081] Das Feld Übersetzen umfasst auch drei Parameter des Typs Real. Jeder dieser Parameter spezifiziert eine Übersetzung des Modells entlang einer von drei orthogonalen Achsen in einem Koordinatensystem. Daher ermöglicht Übersetzen Übersetzungen, die nicht durch die Module des modularen Koordinatensystems begrenzt sind.

Segment (Modell, Segment)

[0082] Der Segmentblock enthält einen durch Elementblöcke und, wahlweise, untergeordnete Segmentblöcke gebildeten Körperteil. Die Felder im Segment sind "Versatz", "Ausrichtung", "Achse" und "Drehen":

```
Segment:
{
Versatz [Ganzzahl; Ganzzahl; Ganzzahl];
Ausrichtung [2 Bit; 2 Bit; 2 Bit];
Achse [3 Bit];
Drehen [Real];
Element {...};
```



```
Segment {...};
}
```

[0083] Das Feld Versatz beschreibt die Position eines Ursprungs des aktuellen Segments in bezug zum Ursprung eines anderen Segments, das – in der Hierarchie – dem aktuellen Segment übergeordnet ist. Wenn kein übergeordnetes Segment existiert, wird das Feld Versatz weggelassen. Das Feld Versatz umfasst drei Parameter des Typs Ganzzahl zum Beschreiben der Position. Durch Beschränken dieser Position auf Ganzzahlwerte wird eine wesentliche Reduzierung in der Datenmenge erhalten, die zum Darstellen eines vollständigen Modells benötigt wird.

[0084] Das Feld Ausrichtung (in **Fig. 3** weggelassen) beschreibt die Ausrichtung des Segments, d.h., ob das Segment, z.B. mit der Unterseite nach oben, um 90 Grad nach links gedreht etc. ausgerichtet ist. Das Feld Ausrichtung umfasst drei Parameter von 2 Bit zum Beschreiben dieser Ausrichtung in 90-Grad-Schritten um eine oder mehrere der drei Achsen in dem Koordinatensystem.

[0085] Das Feld Achse beschreibt die Ausrichtung einer Achse, um die das Segment gedreht werden kann. Nur ein Parameter von 3 Bit wird zum Spezifizieren einer der drei Achsen in dem Koordinatensystem verwendet, da die Ausrichtung in bezug zum Ursprung des aktuellen Segments beschrieben ist.

[0086] Das Feld Drehen beschreibt eine Drehung des aktuellen Segments um die vorhergehend definierte Achse, die in dem Feld Achse definiert wurde. Das Feld Drehen umfasst einen einzigen Parameter des Typs Real. Dieser Parameter kann zum Spezifizieren eines statischen Winkels des aktuellen Segments verwendet werden, oder der Parameter kann modifiziert werden, um eine Manipulierung zu simulieren, z.B. in Form einer Drehung des aktuellen Segments mit einer spezifizierten Winkelgeschwindigkeit.

[0087] Daher wird durch die oben definierten Beschränkungen eine wesentliche Reduzierung in der Datenmenge erreicht, die zum Darstellen eines vollständigen Modells benötigt wird. Dieser Vorteil wird erreicht, ohne einen Benutzer nicht zufrieden zu stellen, der ein erfindungsgemäßes Modell erzeugt, da die Beschränkungen in einem modularen Bausystem oder einer modularen, virtuellen Realität erwartet werden. Wenn die Beschränkungen jedoch ein Begrenzungsfaktor in einem Teil einer Modellierungsaufgabe sind, kann z.B. ein Feld Übersetzen, wie in dem Modellblock definiert, angewendet werden, um die Beschränkungen zu überwinden oder zu lösen. Das heißt, eine Verschiebung unter Verwendung nicht-ganzer Zahlen, z.B. des Typs Real, kann verwendet werden.

[0088] Die Anweisungen "Element {...}" und "Segment {...}" zeigen an, dass das betreffende Segment untergeordnete Elemente und Segmente gemäß der Hierarchie eines erfindungsgemäßen Modells aufweisen kann.

Element (Segment)

[0089] Der Elementblock ist das geometrische Grundelement gemäß der Erfindung. Er bezieht sich auf einen geometrischen Körper mittels des Felds "Formverweis". Es können jedoch andere Verweisfelder definiert werden: "Materialverweis" und "Dekorationsversion". Andere Felder sind "Ausrichtung" und "Versatz":

```
Element:
{
  Versatz [Ganzzahl; Ganzzahl; Ganzzahl];
  Ausrichtung [2 Bit; 2 Bit; 2 Bit];
  Formverweis [Ganzzahl];
  Materialverweis [Ganzzahl];
  Dekorationsverweis [Ganzzahl];
}
```

[0090] Das Feld Versatz wird zum Definieren einer Position des Elements innerhalb eines Segments und gemäß den Modulen des modularen Gitters verwendet. Daher können Parameter des Typs Ganzzahl verwendet werden. Alternativ können Parameter eines eine Kette von alphanumerischen Zeichen aufweisenden Typs in den Feldern Formverweis, Materialverweis und Dekorationsverweis verwendet werden.

[0091] Die Ausrichtung des Elements wird mittels des Felds Ausrichtung und drei Parametern von 2-Bit definiert. Daher ist die Ausrichtung des Elements auf das modulare Gitter des Segments beschränkt.

[0092] Zum Behandeln eines speziellen Modells in einer effizienten Weise enthält der Elementblock einen Verweis auf eine geometrische Beschreibung des Elements anstelle der vollständigen, geometrischen Beschreibung, die eine große Menge geometrischer Daten enthalten kann. Daher kann die Erscheinung eines Modells durch Auswechseln eines Parameters Formverweis in dem Elementblock gegen einen anderen Parameter Formverweis geändert werden.

[0093] Es wird nun mit einem Beispiel eines speziellen Modells gemäß der Erfindung fortgefahren.

[0094] **Fig. 4** zeigt zwei Familien von Elementen. Eine erste Familie, die die Elemente **401**, **402** und **403** aufweist. Eine zweite Familie, die die Elemente **404**, **405** und **406** aufweist. Jedes der Elemente hat eine geometrische Beschreibung, die mit einem Verweis so verknüpft ist, dass sie geometrisch ausgespielt werden kann, wenn ein Element mit einem spe-

zifizierten Verweis angefordert wird. Die geometrische Beschreibung kann durch Entwerfen der Elemente mittels einer computergestützten Entwurfssoftware geliefert werden, wodurch Daten in einem speziellen Format erzeugt werden, das auf einem geeigneten Medium gespeichert und von diesem ausgelesen werden kann. Ein solches, spezielles Format kann z.B. VRML, 3DMF etc. sein. Auf diese Weise können die Daten zum visuellen Interpretieren eines erfindungsgemäßen, virtuellen Modells ausgelesen werden.

[0095] Jedes der Elemente ist mit einer solchen Verweisposition und -Ausrichtung ausgelegt, dass, wenn die Elemente in einem Modell verwendet werden, ihre Position und Ausrichtung in dem Modell in Bezug zu der Verweisposition und -Ausrichtung beschrieben werden kann. Zum Definieren dieser Verweisposition und -Ausrichtung werden die Elemente zu Darstellungszwecken in dieser Offenbarung nur in einem orthogonalen Koordinatensystem mit Achsen X, Y, Z platziert.

[0096] Trotz der verschiedenen Erscheinungen der einzelnen Elemente hat jedes der Elemente ein jeweiliges Bezugsmodul, das einem Modul in dem modularen Gitter eines Segments entspricht. Dieses Bezugsmodul ist mit Strichellinien/einem umschließenden Kasten angezeigt.

[0097] Es soll nun zu den speziellen Elementen übergegangen werden. Das Element **401** ist ein L-förmiges Element, das drei Gittermodule besetzt. Diesem Element ist die Typidentifizierung "T401" zugeordnet. Das Element **402** ist ein kubisches Element, das eine einzelne Gittereinheit besetzt. Diesem Element ist die Typidentifizierung "T402" zugeordnet. Das Element **403** ist ein T-förmiges Element, das 4 Gittereinheiten besetzt. Diesem Element ist die Typidentifizierung "T403" zugeordnet.

[0098] Dementsprechend ist das Element **404** ein L-förmiges Rohrelement, das drei Gittermodule besetzt. Diesem Element ist die Typidentifizierung "T404" zugeordnet. Das Element **405** ist ein gerades Rohrelement, das eine einzelne Gittereinheit besetzt. Diesem Element ist die Typidentifizierung "T405" zugeordnet. Das Element **406** ist ein T-förmiges Rohrelement, das 4 Gittermodule besetzt.

[0099] Diesem Element ist die Typidentifizierung "T406" zugeordnet.

[0100] Daher ist es dann möglich, eine geometrische Beschreibung für jedes der Elemente **401–406** zu speichern und eine geometrische Beschreibung für ein spezielles Element aus der virtuellen Modellierungssprache gemäß der Erfindung mittels der Typidentifizierung, z.B. eines Verweises, auszulesen.

[0101] Vorzugsweise sind die Elemente in den jeweiligen Familien derart ausgelegt, dass sie mit anderen Elementen in ihren jeweiligen Familien verbunden werden können. Dies kann visuell durch solches Entwerfen der Elemente angezeigt werden, dass sie eine erkennbare Schnittstelle in Form einer Oberfläche aufweisen, die mit einer Ebene zwischen zwei benachbarten Gittermodulen zusammenfällt. Bei Betrachtung der röhrenförmigen Elemente **404**, **405** und **406**, kann eine erkennbare Schnittstelle in Form der Enden der Rohre vorliegen. Andere Elemente sind z.B. in den US-Patenten 4,213,403; 4,176,493; 5,795,210; 4,124,949; 5,645,463 gezeigt.

[0102] Fig. 5 zeigt einen ersten Schritt in einem Prozess zum Bauen eines Modells eines geometrischen Gegenstands. Das Element **501** des Typs "T401" ist in einem Segment mit einem orthogonalen Koordinatensystem angeordnet, das durch die Achse Y1, Y1 und Z1 definiert ist. Die Achsen sind nur zu Darstellungszwecken vorgesehen und können gezeigt werden oder nicht gezeigt werden, wenn das Modell Visualisierung unterzogen wird.

[0103] In einer bevorzugten Ausführungsform des Modells virtueller Realität ist definiert, dass das kleinstmögliche Modell mindestens ein Segment aufweisen soll; deshalb ist das Element **501** in einem Segment angeordnet. Es wird festgestellt, dass das Element **501** eine Ausrichtung in Bezug zu den Achsen X1, Y1 und Z1 hat, die sich von der Bezugsausrichtung in der Definition des Typs des Elements unterscheidet, siehe Fig. 4. Diese Ausrichtung ist vorzugsweise als eine Mehrzahl von 90-Grad-Schritten des Elements beschrieben. Das Element **501** wird durch Drehen des Elements 270 Grad um eine Achse parallel zur X1-Achse ausgerichtet, wobei die Winkel gemäß einem Dreifingersystem von Koordinaten gemessen werden (d.h. man lässt den Daumen der X-Achse folgen und nimmt positive Werte in einer Kreisrichtung, die den Fingern folgt – diese Definition wird im Folgenden durchgehend verwendet werden).

[0104] Der bevorzugte Code, der zum Darstellen des Segments mit einem einzelnen Element **501** verwendet wird, ist im Folgenden angeführt:

```
CODEANFANG:
Segment
{
  Element
  {
    Formverweis (T401);
    Ausrichtung (270, 0, 0);
  }
}
CODEENDE
```

[0105] Der Definition der Codesyntax zufolge, wird ein Segment mit einem Element definiert. Ferner wird mittels des Felds Formverweis definiert, dass der Typ

des Elements "T401" ist. Die Ausrichtung des Elements wird mittels des Felds Ausrichtung und der Parameter (270, 0, 0) festgestellt, die angeben, dass das Element 270 Grad in Bezug zur X1-Achse gedreht wird. Es soll festgestellt werden, dass die Parameter in diesem Beispiel als Grad angegeben sind, aufgrund der 90-Grad-Beschränkung kann diese Darstellung jedoch nur in 2 Bits pro Parameter codiert werden.

[0106] Daher kann dieser Code durch eine visuelle Interpretierieranwendung interpretiert werden, die eine geometrische Beschreibung des Elements lädt, die Position und Ausrichtung des Elements in Übereinstimmung mit der in dem obigen Code gegebenen Anweisung berechnet und eine gewünschte Projektion auf eine Ebene oder Sphäre zum Anzeigen durch Anzeigemittel berechnet.

[0107] Die Größe des Gitters wird als relativ grob, verglichen mit der Auflösung, ausgewählt, die zum Anzeigen des Elements verwendet wird. Typischerweise ist die Größe des Gitters ausgelegt, um gerade eben in der Lage zu sein, die Verbindung zu handhaben oder einfach Ausrichtung in irgendeinem Sinne von Elementen innerhalb einer Familie von Elementen zu ermöglichen.

[0108] Trotz der Tatsache, dass die Elemente in einem Koordinatensystem mittels ganzzahliger Koordinaten positioniert werden, kann die Definition oder Darstellung der geometrischen Formen der Elemente nicht-ganzzahlige Koordinaten aufweisen. Der wichtige Aspekt besteht darin, dass die Position der Elemente mittels Ganzzahlen dargestellt wird.

[0109] Fig. 6 zeigt einen zweiten Schritt in einem Prozess zum Bauen eines Modells eines geometrischen Gegenstands. In diesem Schritt werden mehr Elemente hinzugefügt, und die Wirkung des virtuellen Modells gemäß der Erfindung wird deutlicher zum Vorschein kommen. Nun weist der geometrische Gegenstand drei Elemente **601**, **602** und **603** der jeweiligen Typen "T401", "T403" und "T402" auf. Die Position und Ausrichtung der jeweiligen Elemente sind mittels des folgenden Codes angegeben:

```
CODEANFANG:
Segment
{
  Element
  {
    Formverweis (T401);
    Ausrichtung (270, 0, 0);
  }
  Element
  {
    Formverweis (T403);
    Ausrichtung (0, 0, 270);
    Versatz (3, 0, 0);
  }
}
```

```
Element
{
  Formverweis (T402);
  Versatz (3, 2, 0);
}
}
CODEENDE
```

[0110] Der erste Teil des Codes, der das Element **601** definiert, ist bereits eingeführt. Wie deutlich werden wird, werden die anderen Elemente (**602** und **603**) in dem Segment in Bezug zu diesem Element (**601**) definiert.

[0111] Der nächste Teil des Codes definiert das Element **602** des Typs "T403". Wie zu sehen ist, hat dieses Element nicht die gleiche Ausrichtung wie in der Definition des Typs des Elements; deshalb ist es erforderlich, ein Ausrichtungsfeld hinzuzufügen, das spezifiziert, dass das Element 270 Grad um die Z1-Achse gedreht ist. Ferner wird das Element **602** um einen ganzzahligen Versatzwert von drei Modulen entlang der X1-Achse versetzt.

[0112] Der letzte Teil des Codes definiert die Position des Elements **603** des Typs "T402". Dieses Element ist um einen ganzzahligen Versatzwert von drei Modulen entlang der X1-Achse und einen ganzzahligen Versatzwert von zwei entlang der Y1-Achse versetzt.

[0113] Da Elemente innerhalb eines Segments nur in Übereinstimmung mit den Modulen des Gitters platziert werden können, können zum Definieren von Position und Ausrichtung der Elemente verwendete Parameter sehr effizient dargestellt werden, z.B. mittels Ganzzahlwerten. Auf diese Weise können bei Bedarf weitere Elemente dem Segment hinzugefügt werden.

[0114] Im Fall von Manipulierung wird einfach bestimmt, dass Elemente innerhalb des Segments aufgrund der durch das modulare Gitter gegebenen Beschränkungen nicht in Bezug zueinander manipuliert werden können.

[0115] Fig. 7 zeigt einen zweiten Schritt in einem Prozess zum Bauen eines Modells eines geometrischen Gegenstands gemäß der Erfindung. In diesem Schritt werden noch weitere Elemente an das Modell angehängt und die Wirkung des erfindungsgemäßen, virtuellen Modells wird noch deutlicher erscheinen. Ferner werden in diesem Schritt Elemente hinzugefügt, die nicht unbedingt an die innerhalb eines Segments gegebenen Positions- und Ausrichtungsbeschränkungen gebunden sind. Dies erfordert eine erweiterte Beschreibung des Modells; aufgrund der Tatsache, dass diese erweiterte Beschreibung nur verwendet wird, wenn sie benötigt wird (d.h. beim Beschreiben z.B. von Drehungen), wird die Kompaktheit

und Zweckdienlichkeit des gesamten Modells jedoch aufrechterhalten.

fachsten zu verstehende betrachtet.

Schritt 1

[0116] Nun weist der geometrische Gegenstand fünf Elemente **701**, **702**, **703**, **704** und **705** der jeweiligen Typen "T401", "T403", "T402", "T403" und "T401" auf. Die Position und Ausrichtung der jeweiligen Elemente sind mittels des folgenden Codes gegeben:

```
CODEANFANG:
Segment
{
Element
{
Formverweis (T401);
Ausrichtung (270, 0, 0);
}
Element
{
Formverweis (T403);
Ausrichtung (0, 0, 270);
Versatz (3, 0, 0);
}
Element
{
Formverweis (T402);
Versatz (3, 2, 0);
}
Segment
{
Versatz (0, 0, 2);
Ausrichtung (90, 0, 0);
Drehen (2, 2, 5);
Achse (0, 0, 1);
Element
{
Formverweis (F403);
Versatz (0, -1, 0);
}
Element
{
Formverweis (T401);
Versatz (-3, -1, 0);
Ausrichtung (270, 0, 0);
}
}
CODEENDE
```

[0117] Der erste Teil des Codes definiert die in **Fig. 6** eingeführten Elemente **701**, **702** und **703**. Der nächste Teil des Codes definiert ein weiteres Segment, das 45 Grad in Bezug zu einer Achse parallel zu der Z-Achse (d.h. der vertikalen Achse) gedreht wurde und Elemente **704** und **705** aufweist.

[0118] Der Teil des Codes, der die Elemente **704** und **705** definiert, kann auf verschiedene Weisen erzeugt werden; die folgende, schrittweise Vorgehensweise mit Schritten **1–4** wird jedoch als die am ein-

[0119] In diesem Schritt der Codeerzeugung wird bestimmt (z.B. als Reaktion auf einen Benutzerbefehl), dass ein neues Segment dem Modell hinzugefügt werden soll. Unter Berücksichtigung dieser Vorgabe wird das Element **704** als das erste Element in dem neuen Segment hinzugefügt, das ein modulares Gitter (nicht gezeigt) aufweist, welches mit dem Gitter oder Koordinatensystem zusammenfällt, wie es in der Definition des Elements (siehe **Fig. 4**) des Typs "T403" definiert ist. Daher müssen keine Versatz- oder Ausrichtungsfelder spezifiziert werden. Anschließend wird das Element **705** dem neuen Segment hinzugefügt. Dieses Element ist jedoch um einen Ganzzahlwert von -3 entlang der X1-Achse versetzt, und die Ausrichtung ist um 270 Grad um die X-Achse in Bezug zur Ausrichtung des Elements in der Definition des Elements des Typs "T401" versetzt. Dies wird durch die Versatz- und Ausrichtungsfelder in dem folgenden Code definiert:

```
CODEANFANG:
Segment
{
Element
{
Formverweis (T403);
}
Element
{
Formverweis (T401);
Versatz (-3, 0, 0);
Ausrichtung (270, 0, 0);
}
}
CODEENDE
```

Schritt 2

[0120] In diesem Schritt der Codeerzeugung wird der Ursprung des neuen Segments bestimmt, um dort angeordnet zu sein, wo das neue Segment mit dem alten Segment verbunden ist, d.h. am Boden des "T-förmigen" Elements **704**. Deshalb werden die Positionen der Elemente **704** und **705** neu berechnet, um mit diesem neuen Ursprung des Segments in Übereinstimmung zu sein:

```
CODEANFANG:
Segment
{
Element
{
Formverweis (T403);
Versatz (0, -1, 0); //hinzugefügt
}
Element
{
Formverweis (T401);
```

```

Versatz (-3, -1, 0); //modifiziert
Ausrichtung (270, 0, 0);
}
}
CODEENDE

```

[0121] Es wird festgestellt, dass dies beinhaltet, dass ein Versatzfeld für das Element **704** hinzugefügt wird und dass das Versatzfeld für das Element **705** zum Reflektieren dieses neuen Ursprungs modifiziert wird.

[0122] Diese Neuberechnung muss auch durchgeführt werden, wenn das neue Segment mit dem alten Segment an einer anderen Position in dem neuen Segment verbunden ist, z.B. wie in **Fig. 2b** gezeigt ist. Da jedoch die Versatzparameter des Typs Ganzzahl sind, kann diese Neuberechnung sehr schnell, verglichen mit einer Situation, durchgeführt werden, in der Parameter des Typs Real verwendet werden würden. Trotz der Tatsache, dass viele Addier- und Subtrahieroperationen in einem praktischen Modell auszuführen sein werden, werden aufgrund der ganzzahligen Darstellung keine Abrundungsfehler eingebracht werden.

Schritt 3

[0123] In diesem Schritt der Codeerzeugung wird spezifiziert, wo das neue Segment mit dem alten Segment verbunden werden soll. Dies wird durch Spezifizieren festgestellt, wie der Ursprung des neuen Segments in Bezug zum Ursprung des alten Segments übersetzt wird. Dies erfolgt durch Einfügen des folgenden Versatzfelds in den zweiten Segmentblock des in Schritt 2 oben gegebenen Codes:

Versatz (0, 0, 2).

[0124] Auf diese Weise wird der Ursprung des neuen Segments in einen ganzzahligen Wert von 2 entlang der Z1-Achse übersetzt.

Schritt 4

[0125] Nach Definieren der Position des neuen Segments und der Elemente innerhalb des neuen Segments kann die Ausrichtung des neuen Segments in Bezug zum alten Segment spezifiziert werden. Dies erfolgt durch Einsetzen des folgenden Felds "Drehen" in den in Schritt 2 oben angeführten Code:

Drehen (90, 225, 0).

[0126] Daher wird das neue Segment erstens 90 Grad um die X-Achse gedreht, und dann wird das Segment zweitens um 225 Grad um die Y-Achse gedreht, die, wie vorhergehend definiert, bereits 90 Grad gedreht wurde.

[0127] Im Fall von Manipulierung wird, wie vorhergehend aufgeführt, einfach bestimmt, dass Elemente innerhalb der Segmente aufgrund der durch das modulare Gitter gegebenen Beschränkungen nicht in Bezug zueinander manipuliert werden können. Es wird jedoch auch einfach bestimmt, dass die Segmente sich aufgrund der impliziten Definition der Segmente in Bezug zueinander drehen dürfen. Daher muss keine zusätzliche Meta-Beschreibung, d.h. eine Definition möglicher Manipulierungen (d.h. Bewegungen oder Drehungen) definiert werden. Wenn erwünscht ist, die Segmente in Bezug zueinander und ohne Beschränkung durch die jeweiligen, modularen Gitter zu übersetzen, kann ein einfaches Übersetzungsfeld hinzugefügt werden, das die Übersetzung definiert.

[0128] **Fig. 8** zeigt einen dritten zu modellierenden, einfachen Gegenstand gemäß der Erfindung. Dieser Gegenstand kann mit dem gleichen Modell beschrieben werden, wie für **Fig. 7** oben beschrieben wurde, wobei nur der Typ der jeweiligen, mittels der Feldparameter "Formverweis" beschriebenen Elemente gegen die jeweiligen Typen der Elemente der in **Fig. 4** definierten, zweiten Familie von Elementen ausgetauscht wurde. Das Modell weist Elemente **801, 802, 803, 804** und **805** auf, die jeweils den Elementen **701, 702, 703, 704** bzw. **705** in **Fig. 7** entsprechen.

[0129] Es wird daher dargestellt, dass eine andere Familie von Elementen in dem Modell ohne Verlust der effizienten Beschreibung verwendet werden kann. In einer bevorzugten Ausführungsform ist die Familie von Elementen an eine Klasse von Modellierungssparadigmen angepasst, so dass die Elemente gut bekannte Gegenstände der realen Welt darstellen, die sich als geeignet für eine gegebene Modellierungsaufgabe herausgestellt haben.

[0130] Ferner ist es gemäß der obigen Offenbarung einer bevorzugten Ausführungsform und gemäß einer Aufgabe der Erfindung einer Fachperson in diesem Gebiet klar, dass eine Meta-Beschreibung, d.h. eine Beschreibung möglicher Manipulierungen eines geometrischen Gegenstands, implizit mittels einer geometrischen Beschreibung durch Modifizieren der Parameter der jeweiligen Anweisungen gegeben ist. Bei Betrachtung beispielsweise der Verbindung von zwei Segmenten wird eine solche Verbindung mittels eines Felds Achse und eines Felds Drehen angegeben. Durch Modifizieren des Drehwinkels wird die geometrische Beschreibung geändert, während die Manipulierung, d.h. Drehung, stattfindet. In einem solchen Fall wird die Drehachse in der geometrischen Beschreibung definiert; so wird auf diese Weise die Meta-Beschreibung in der geometrischen Beschreibung eingebettet. Wenn es um Elemente innerhalb eines Segments geht, ist es offensichtlich, dass diese Elemente keiner Drehung unterzogen werden können, da keine Parameter in der geometrischen Be-

schreibung eine Meta-Beschreibung für Drehung zulassen. In einer praktischen Ausführungsform wird ein Benutzer jedoch bei Bedarf mit Steuerungen zum Anwenden neuer Meta-Beschreibungen versorgt. Tatsächlich sind solche neuen Meta-Beschreibungen einfach anzuwenden, da nur ein begrenzter Satz von Beschreibungen in der Modellstruktur vorliegt, und da die Parameter auf begrenzte Sätze von Werten beschränkt sind.

[0131] Nach Definieren der Struktur von Modellen gemäß der Erfindung, von Modifikationen und Manipulationen der Modelle soll ein Computersystem zum Ausführen dieser Aufgaben offenbart werden.

[0132] Fig. 9 zeigt ein Clientcomputersystem zum Entwerfen, Speichern, Manipulieren und Teilen geometrischer Konstruktionen gemäß der Erfindung. Das allgemein mit **901** gekennzeichnete Clientcomputersystem kann als ein alleinstehendes System oder als ein Client in einem Client/Server-System verwendet werden. Der Client umfasst einen Speicher **902**, der teilweise als ein flüchtiges und ein nicht-flüchtiges Speichermittel realisiert ist, z.B. eine Festplatte und ein Direktzugriffsspeicher (RAM). Der Speicher umfasst einen Prozedur- oder Programmcodeinterpretierer **907**, Codegenerator **908**, Benutzeroberflächen (UI)-Ereignis-Handler **909** und eine Modellierungsanwendung **910**, die durch die zentrale Verarbeitungseinheit **903** ausführbar ist. Ferner weist der Speicher Modelldaten **911** auf.

[0133] Der Codeinterpretierer **907** ist ausgelegt, um den ein Modell gemäß der Erfindung definierenden Code zu lesen und zu interpretieren. In einer bevorzugten Ausführungsform ist der Codeinterpretierer ausgelegt, um ein erfindungsgemäßes Modell zu lesen und ein solches Modell in ein bekanntes, graphisches Format für Präsentation auf einer Computeranzeige zu konvertieren. Wenn die vorhergehend definierte Codestruktur zum Darstellen eines Modells eines Gegenstands einer Fachperson in diesem Gebiet bekannt ist, kann diese Konvertierung durch Anwenden gut bekannter, graphischer Prinzipien realisiert werden, die im Feld von Graphikverarbeitung bekannt sind. Eine nicht-erschöpfende Liste von durch den Codeinterpretierer **907** erfüllten Funktionen ist im folgenden aufgeführt.

Funktion 1

[0134] Wenn ein Weltblock gelesen wird, wird der Codeinterpretierer (CI) ein Koordinatensystem aufbauen, auf das sich anschließende Anweisungen beziehen werden.

Funktion 2

[0135] Wenn ein Modellblock gelesen wird, wird der CI vorbereiten, ein Segmente und Elemente aufwei-

sendes Modell in dem durch die Funktion 1 definierten Koordinatensystem zu platzieren. Dies erfolgt gemäß den Feldern Drehen und Übersetzen.

Funktion 3

[0136] Wenn ein Segmentblock gelesen wird, wird der CI vorbereiten, ein Element aufweisendes Segment gemäß der in Funktion 2 spezifizierten Position und Ausrichtung und gemäß einem Ausrichtungsfeld zu platzieren, das die Ausrichtung des Segments spezifiziert. Zusätzlich werden die Felder Versatz, Achse und Drehen zum Bestimmen der Position untergeordneter Segmente gelesen.

Funktion 4

[0137] Wenn ein Elementblock gelesen wird, wird die CI eine geometrische Beschreibung eines Elements, das durch einen Elementtypverweis in einem Formverweisfeld spezifiziert ist, durch Zugriff auf eine Datenbank lesen, die z.B. als die Modelldaten **911** gespeichert ist.

[0138] Das Ergebnis der aufeinanderfolgend und/oder iterativ zitierten Funktionen 1, 2, 3 und 4 wird als Daten geliefert, die auf eine Ebene oder Sphäre projiziert werden, oder als Daten, die einer Projektion unterzogen werden können.

[0139] Der Benutzeroberflächen-Ereignis-Handler **909** ist ausgelegt, um eine Benutzerinteraktion mit einer Benutzeroberfläche in richtige Benutzerbefehle zu konvertieren, die durch den Codegenerator **908** erkennbar sind. Ein Satz möglicher und erkennbarer Befehle kann aufweisen: Holen eines Elements aus einer Bibliothek von Elementen; Verbinden eines Elements mit einem anderen Element; Lösen eines Elements; Ablegen eines Elements; Manipulieren eines Segments, z.B. Initiierung einer Drehung, etc. Zusammen mit jedem Befehl ist ein Satz jeweiliger Parameter verknüpft, wie sie vorhergehend in jedem der Felder definiert wurden.

[0140] Der Codegenerator **908** ist zum Modifizieren des Codes ausgelegt, der ein aktuelles, erfindungsgemäßes Modell, wie oben beschrieben, und als Reaktion auf Benutzerbefehle beschreibt. Eine nicht-erschöpfende Liste von durch den Codegenerator **908** erfüllten Funktionen ist im folgenden angeführt: (es wird angenommen, dass ein Segment vorhergehend spezifiziert wird).

Funktion 5

[0141] Wenn ein Element aus einer Bibliothek von Elementen ausgewählt wird, wird sein Typ durch das Feld Formverweis spezifiziert.

Funktion 6

[0142] Wenn das Element mit einem Element innerhalb eines Segments verbunden ist, wird das Segment identifiziert, und die Position und die Ausrichtung des verbundenen Elements innerhalb des Segments wird bestimmt und mittels der Felder Versatz und Ausrichtung dargestellt, die in die Darstellung des Modells eingesetzt werden.

Funktion 7

[0143] Wenn ein Element gelöscht wird, wird der die Position und Ausrichtung des Elements darstellende Code gelöscht.

[0144] Als eine gleichzeitige oder anschließende Aufgabe kann der Codeinterpretierer zum Darstellen des Ergebnisses des Codegenerators ausgeführt werden.

[0145] Die Modellierungsanwendung **910** ist ausgelegt, um Speicher, Dateien, die Benutzeroberfläche etc. zu steuern.

[0146] Ein Benutzer **905** kann mit dem Clientcomputersystem **901** mittels der Benutzeroberfläche **906** zusammenwirken.

[0147] Zum Laden von Modellen, geometrischen Beschreibungen oder anderen Daten weist das Clientcomputersystem eine Eingabe/Ausgabeeinheit (I/O) **904** auf. Die Eingabe/Ausgabeeinheit kann als eine Schnittstelle zu verschiedenen Typen von Speichermedien und verschiedenen Typen von Computernetzen, z.B. dem Internet, verwendet werden. Ferner kann die Eingabe/Ausgabeeinheit (I/O) **904** zum Austauschen von Modellen mit anderen Benutzern, z.B. interaktiv verwendet werden.

[0148] Der Datenaustausch zwischen dem Speicher **902**, der zentralen Verarbeitungseinheit (CPU) **903**, der Benutzeroberfläche (UI) **906** und der Eingabe/Ausgabeeinheit (I/O) **904** wird mittels des Datenbusses **912** erreicht.

[0149] Fig. 10 zeigt ein Servercomputersystem zum Behandeln geometrischer Konstruktionen gemäß der Erfindung. Das allgemein mit **1001** bezeichnete Servercomputersystem wird zum Bedienen einer Anzahl von Clientcomputersystemen verwendet. Der Server weist einen Speicher **1002** auf, der teilweise als ein flüchtiges und ein nicht-flüchtiges Speichermittel, z.B. eine Festplatte und ein Direktzugriffsspeicher (RAM), realisiert ist. Der Speicher umfasst Prozeduren oder Programme "Codeinterpretierer" **1005**, "Codegenerator" **1006**, "Sitzungshandler" **1007**, die durch die zentrale Verarbeitungseinheit **1003** ausführbar sind. Ferner umfasst der Speicher "Modelldaten" **1008** z.B. in Form geometrischer Beschreibungen,

gen, Beschreibungen von Modellen etc.

[0150] Zum Laden von Modellen, geometrischen Beschreibungen oder anderer Daten oder zum Verteilen von Daten auf Clientcomputersysteme weist das Servercomputersystem eine Eingabe/Ausgabeeinheit (I/O) **1004** auf. Die Eingabe/Ausgabeeinheit (I/O) kann als eine Schnittstelle zu verschiedenen Typen von Speichermedien und verschiedenen Typen von Computernetzen, z.B. dem Internet, verwendet werden.

[0151] Datenaustausch zwischen dem Speicher **1002**, der zentralen Verarbeitungseinheit (CPU) **1003** und der Eingabe/Ausgabeeinheit **1004** wird mittels des Datenbusses **1009** bereitgestellt.

[0152] Es wird nun mit einer detaillierten Beschreibung der Funktionen/Computerprogramme fortgefahren, auf die in der Beschreibung der Fig. 9 und 10 Bezug genommen wurde.

[0153] Fig. 11 zeigt ein Ablaufdiagramm für ein Verfahren zum gleichzeitigen Konstruieren von Segmenten und Modellen, während die Segmente und Modelle Manipulierung unterzogen werden. Dieses Ablaufdiagramm beschreibt auch die Operation einer Prozedur, die auf einem Client- und/oder Servercomputer zum Behandeln eines erfindungsgemäßen Modells ausgeführt wird.

[0154] In Schritt **1101** werden Benutzerbefehle empfangen, die mittels Benutzersteuerungen, welche Teil der Benutzeroberfläche **1102** sind, eingegeben werden. Benutzersteuerungen können viele verschiedene Typen aufweisen, z.B. eine Computertastatur, Joystick oder Maus in Interaktion mit einer Benutzeroberfläche **1121** aufweisen, wodurch Signale an ein Programm geliefert werden, um eine Benutzerinteraktion mit der Benutzeroberfläche darzustellen. Zum Beispiel kann ein Benutzer ein auf der Benutzeroberfläche **1121** angezeigtes Element auswählen und es mittels einer Computermouse unter Verwendung der gut bekannten Drag-and-Drop-Operation zu einer gewünschten Position bewegen. Daher wird ein diese Bewegung des Elements darstellender Benutzerbefehl erzeugt. Eine solche Operation der Benutzeroberfläche ist auch als ereignisgesteuerte Programmierung bekannt.

[0155] Der Benutzerbefehl wird in dem Block **1113** des Ablaufdiagramms empfangen und zum Erzeugen, Aktualisieren oder Manipulieren von Modelldaten **1112** gemäß der Erfindung verwendet. Die Modelldaten **1112**, die ein Ergebnis der Verarbeitung in dem Block **1113** darstellen, können zu einem Kommunikationsnetz über eine Eingabe/Ausgaberichtung (I/O) **1117** übertragen werden. Daher ist es möglich, Modelle mit anderen Benutzern auszutauschen; Modelle zu teilen; Modelle von einem Server zu laden;

Modelle auf einem Server zu speichern etc. Typischerweise werden die Modelldaten **1112** mit den die Erscheinung gespeicherter Elemente darstellenden Daten mittels des Umsetzers **1118** gemischt. Der Umsetzer verwendet den vorhergehend definierten "Formverweis" zum Erhalten einer geometrischen Darstellung dieser Erscheinung von der Datenbank **1119**. Ferner werden in dem Umsetzer **1118** die geometrische Darstellung und die vorhergehend definierten Blöcke und Felder, die die Form eines speziellen Modells beschreiben, kombiniert und in ein Format konvertiert, das durch ein 3D-Realitätssimulationsgerät **1120** interpretiert werden kann. Das 3D-Realitätssimulationsgerät **1120** liefert das Modell darstellende Informationen, so dass das Modell auf einer Benutzeranzeige **1121** visualisiert werden kann.

[0156] Es folgt nun eine detaillierte Beschreibung des Blocks **1113** des Ablaufdiagramms.

[0157] Wenn ein Benutzerbefehl empfangen wird, wird bestimmt, ob der Benutzerbefehl eine Elementoperation, eine Segmentoperation oder eine Modelloperation betrifft. Eine Elementoperation wird in Schritt **1116** erfasst, wenn ein Element dem Modell hinzugefügt wird, von dem Modell entfernt oder aus einer ersten zu einer zweiten Position innerhalb des Modells bewegt wird. Eine Segmentoperation wird in Schritt **1115** erfasst, wenn ein Segment in dem Modell bewegt, entfernt oder manipuliert wird oder alternativ, wenn ein neues Segment dem Modell hinzugefügt wird. Eine Modelloperation wird in Schritt **1114** erfasst, wenn ein Modell manipuliert, d.h. frei bewegt wird, ohne auf das Gitter beschränkt zu sein, oder wenn z.B. ein Segment frei platziert wird, auch ohne auf das Gitter beschränkt zu sein.

[0158] In Übereinstimmung mit den erfassten Operationen werden Modelldaten erzeugt, aktualisiert oder manipuliert.

[0159] Wenn der Benutzer ein Element aus einer Bibliothek von Elementen auswählt, die z.B. die in **Fig. 4** gezeigten Elemente aufweist, wird Schritt **1110** "Hole neues Element" eingegeben, und das ausgewählte Element wird zum Beispiel als der Typ "T401" identifiziert. Anschließend wird in Schritt **1111** das ausgewählte Element mit einem anderen Element verbunden, das sich bereits in der Welt befindet, oder alternativ wird das Element allein platziert, wodurch es als ein Segment behandelt wird. Wenn das ausgewählte Element verbunden ist, wird der Code, der dieses Element, seine Position und Ausrichtung darstellt, an einen existierenden Code angehängt oder ein neuer Code wird erzeugt.

[0160] Wenn ein anderes Element in der Welt existiert, kann der Benutzer entscheiden, dieses Element auszuwählen und das Element aus seiner aktuellen Position zu bewegen oder zu lösen. Wenn ein sol-

ches Ereignis erfasst wird, wird der Teil des das Elements darstellenden, existierenden Codes entfernt und zeitweilig gespeichert.

[0161] Das ausgewählte Element kann entweder abgelegt oder zu einer anderen Position verbunden/bewegt werden. Wenn das Ereignis, dass das Element abgelegt wurde, erfasst wird, wird das Element von der Benutzeroberfläche verschwinden, und der temporäre Code wird in Schritt **1109** gelöscht. Wenn alternativ das Ereignis erfasst wird, dass das Element mit einer anderen Position verbunden ist, wird ein das Element, seine neue Position und Ausrichtung darstellender Code erzeugt und in den existierenden Code eingefügt.

[0162] Zum Beschleunigen des Prozesses zum Erzeugen oder Editieren eines existierenden Modells kann ein gesamtes Segment mit anderen Segmenten und Elementen als ein einziger Gegenstand behandelt werden.

[0163] Wenn ein Segment in der Welt existiert, kann der Benutzer entscheiden, dieses Segment auszuwählen und das Segment aus seiner aktuellen Position mittels einer Aktion zu bewegen oder zu lösen, die für alle Segmente und Elemente innerhalb des Bewegung oder Lösung ausgesetzten Segments gemeinsam ist. Wenn ein solches Ereignis erfasst wird, wird der Teil des existierenden Codes, der das gesamte Segment darstellt, entfernt und zeitweilig gespeichert.

[0164] Wenn das Ereignis erfasst wird, dass das gesamte Segment abgelegt werden soll, wird das gesamte Segment von der Benutzeroberfläche verschwinden, und der temporäre Code wird in Schritt **1106** gelöscht werden. Wenn alternativ das Ereignis erfasst wird, dass das Element mit einer anderen Position verbunden ist, wird ein das Segment, seine neue Position und Ausrichtung darstellender Code erzeugt und in den existierenden Code eingefügt.

[0165] Darüber hinaus können Segmente manipuliert und Modelle können bewegt werden. Wenn Segmente in der Welt existieren, können sie manipuliert werden, d.h. Elemente oder andere Segmente innerhalb des der Manipulierung unterzogenen Segments werden nicht geändert, aber das Segment kann z.B. um einen speziellen Winkel gedreht oder festgelegt werden, um mit einer speziellen Winkelgeschwindigkeit zu rotieren. Dies kann durch Erfassen eines spezifizierten Ereignisses von der Benutzeroberfläche und Modifizieren des Codes in Schritt **1104** für das der Manipulierung ausgesetzte Segment erfolgen, z.B. durch Modifizieren der Parameter in dem Feld Drehen eines Segmentblocks.

[0166] Wenn Modelle in der Welt existieren, können sie in ähnlicher Weise zum Manipulieren eines Seg-

ments bewegt werden, wobei jedoch stattdessen die Parameter in den Feldern Drehen und Übersetzen eines Modellblocks in Schritt **1103** modifiziert werden.

[0167] Bei einem vordefinierten Ereignis kann die Prozedur in Schritt **1112** beendet oder angehalten werden, das z.B. das Speichern erzeugter Segmente und Modelle aufweist.

[0168] Fig. 12 zeigt ein Ablaufdiagramm für ein Verfahren zum Zugreifen auf und Teilen einer gemeinsamen Welt, die für mehrere Clients zugänglich ist. In einer bevorzugten Ausführungsform der Erfindung hält ein Server ein Modell einer Welt. Das Modell kann an eine Anzahl von Clients verteilt werden, die Segmente und Modelle in der Welt manipulieren oder auf andere Weise ändern dürfen, oder neue Modelle und Segmente können der Welt hinzugefügt werden. Durch jeweilige Clients vorgenommene Änderungen können zu anderen Clients übertragen werden, die die selbe Welt teilen. So wird ein jeweiliger Client erfahren, dass seine lokale Kopie der Welt sich ändern wird, wenn andere Clients Änderungen an dem Modell der Welt vornehmen. Ein solches Update der lokalen Kopie der Welt des Clients muss sehr schnell und effizient ausgeführt werden, wenn eine große Anzahl von Benutzern unterstützt werden soll, und übermäßige Update-Zeit sollte vermieden werden. Dies wird mit der vorliegenden Erfindung ermöglicht und ist der Schlüssel zum Liefern einer für mehrere Clients zugänglichen, gemeinsamen Welt, die sich aufgrund der eigenen Interaktion verschiedener Clients mit der Welt ständig zu ändern scheint.

[0169] Im ersten Schritt **1201** des Ablaufdiagramms "Lade Welt" wird eine Log-on-Sitzung zwischen einem Client und dem Server ausgeführt, die das Identifizieren des Namens des Clients, ein richtiges Datenübertragungsprotokoll etc. umfasst. Im nächsten Schritt **1202** wird die Beschreibung der Welt durch den Server zum Client gesendet. Vor, während oder nach dieser Datenübertragung tauschen der Client und der Server Informationen zum Feststellen aus, ob alle Komponenten, d.h. Modelle, Segmente, Elemente, etc. in dem Modell dem Client bekannt sind. Diese Überprüfung wird in Schritt **1203** ausgeführt. Wenn irgendwelche unbekannten Modelle in der Welt vorhanden sind, wird eine richtige Beschreibung der relevanten Modelle in Schritt **1204** übertragen. Wenn keine unbekannten Modelle vorliegen, wird der Client so eingeschätzt, dass er eine Kopie der Welt hat und daher Modelle und Segmente manipulieren oder auf andere Weise Änderungen in der Welt vornehmen darf.

[0170] Im Folgenden wird ein Verfahren zum Versorgen eines Clients mit weiteren Informationen gemäß der hierarchischen Struktur eines erfindungsgemäßen Modells dargestellt. Es wird angenommen, dass Modelle, Segmente, Elemente und andere

Komponenten eines Modells eine eindeutige Identifikationsnummer haben. Daher kann der Server beim Client anfragen, ob z.B. ein spezielles Modell vorliegt; wenn das spezielle Modell vorliegt, müssen keine weiteren Daten übertragen werden; ansonsten kann der Server beim Client anfragen, ob ein spezielles Segment (e) des speziellen Modells vorliegt (vorliegen). Auf diese Weise müssen nur unbekannte Modellkomponenten zum Client übertragen werden.

[0171] Fig. 13 zeigt ein Ablaufdiagramm für ein Verfahren zum Erhalten unbekannter Modelle, Segmente und Elemente. Es ist jedoch nur gezeigt, wie unbekannte Modelle zu erhalten sind, da das Ablaufdiagramm allgemein auf das Holen von Modellen, Segmenten und Elementen durch aufeinanderfolgendes Untersuchen der jeweiligen Ebenen der Modellhierarchie anwendbar ist.

[0172] Im ersten Schritt **1301** des Ablaufdiagramms "Hole Modell" wird das Verfahren zum Holen eines Modells eingegeben. In Schritt **1302** wird überprüft, ob das Modell beim Client in irgendeiner Form erhältlich ist; wenn dies der Fall ist, wird das Modell im Schritt **1305** lokal zu einer Laufzeit-Datenbank übertragen. Die Laufzeit-Datenbank ist für das Halten von Komponenten zuständig, die Teil der Welt sind, mit der der Client interagieren möchte. Wenn das Modell alternativ nicht lokal verfügbar ist, aber fern erhältlich ist, d.h. am Server, wird das Modell vom Server zu der beim Client angeordneten Laufzeit-Datenbank übertragen. Diese Entscheidung wird in Schritt **1303** getroffen, und Daten werden in Schritt **1305** übertragen. Wenn die Entscheidung in Schritt **1303** ist, dass das Modell auch fern nicht verfügbar ist, muss ein Fehler in Schritt **1304** behandelt werden.

[0173] Wenn die Modelle, die vom Server zum Client übertragen werden, dem Client nicht bekannte Segmente aufweisen, werden diese Segmente gleichermaßen gegen die unbekannten Modelle ausgetauscht. Es wird in Schritt **1306** überprüft, ob irgendwelche unbekannten Segmente in den ausgetauschten Modellen vorliegen. Wenn unbekannte Segmente vorhanden sind, werden diese in Schritt **1308** ausgetauscht; ansonsten darf der Client fortfahren, das Weltmodell in Schritt **1307** zu modifizieren.

[0174] Die bevorzugte Ausführungsform, wie sie in den Fig. 9, 10, 11, 12 und 13 gezeigt ist, kann in Hardware, Software oder einer Kombination aus diesen realisiert werden.

[0175] Die Software kann mittels eines jeglichen Datenspeicher- oder Datenübertragungsmediums verteilt werden. Das Medium umfasst Floppydisk, CD-ROM, Minidisk, Kompaktdisk oder ein Netz. Das Netz kann z.B. das Internet sein. Über das Netz kann eine Softwareausführungsform der Erfindung durch Download eines Computerprogramms verteilt wer-

den.

[0176] Ferner können geometrische Beschreibungen von Elementen, Teile von Modellen oder vollständige Modelle mittels eines jeglichen der oben genannten Datenspeicher- oder Datenübertragungsmedien verteilt oder ausgetauscht werden.

[0177] Es sollte festgestellt werden, dass, obwohl die Darstellung der für die jeweiligen Felder definierten Parameter mittels einfach zu lesender Buchstaben und Ziffern angegeben ist, die Darstellung eines Modells gemäß der Erfindung in keiner Weise auf eine solche Darstellung begrenzt ist. Tatsächlich wird typischerweise eine komprimierte oder codierte Darstellung der Parameter verwendet werden. Dies kann mittels eines bekannten Verfahrens innerhalb des Gebiets von Datenkomprimierung erfolgen.

[0178] Es soll darüber hinaus festgestellt werden, dass die Anweisungen in der Codestruktur in Bits in einer Sendezeichendarstellung konvertiert werden können. Die Sendezeichendarstellung kann daher zwischen Computern als eine kompakte Datenstruktur oder ein Datenpaket übertragen werden.

[0179] In den folgenden, zitierten Verweisstellen sind Elemente offenbart, die vorzugsweise durch eine geometrische Beschreibung (z.B. VRML, 3DMF etc.) zu beschreiben sind und als Elemente in einem erfindungsgemäßen Modell verwendet werden.

[0180] Das der Interlego AG abgetretene US-Patent 5,645,463 offenbart ein Spielbauelement, das eine Grundplatte und eine Scheibe aufweist, die schwenkbar an einer Kreisöffnung in der Grundplatte angebracht ist. Sowohl die Grundplatte als auch die Scheibe weist in einem modularen Gitter angeordnete Verbindungsmittel auf. In einer bevorzugten Ausführungsform wird das Bauelement der Erfindung zufolge modelliert, indem die Grundplatte als ein erstes Segment oder ein Teil eines ersten Segments genommen wird und die Scheibe als ein zweites Segment eines Teils eines zweiten Segments genommen wird. Von der vorhergehenden Beschreibung der Erfindung wird deutlich werden, dass es möglich ist, eine Rotation der Scheibe in der Grundplatte mittels eines Achsenfelds und eines Drehfelds zu modellieren. Die Drehachse ist modelliert, um sich in der Mitte der Grundplatte mit einer Richtung senkrecht zu der Ebene der Grundplatte zu befinden.

[0181] Das der Interlego AG abgetretene US-Patent 4,214,403 offenbart ein Spielgelenkelement, das ein Paar kastenförmiger Spielbaublöcke aufweist, welche gelenkartig mittels einer röhrenförmigen Buchse in einer Ecke eines ersten der beiden Blöcke und eines komplementären Drehzapfens in einem zweiten der beiden Blöcke verbunden sind. In einer bevorzugten Ausführungsform wird das Bauelement gemäß

der Erfindung modelliert, indem der erste Block als ein erstes Segment und der zweite Block als ein zweites Segment genommen wird. Es ist möglich, eine Drehung des ersten Blocks in Bezug zum zweiten Blocks mittels eines Achsenfelds und eines Drehfelds zu modellieren. Die Drehachse wird so modelliert, um sich in der Mitte des Drehzapfens mit einer dem Drehzapfen folgenden Richtung zu befinden. In der Beschreibung von 4,214,403 erscheint, dass beide der Blöcke mit anderen Blöcken in einer starren Verbindungsweise verbunden werden können. Diese anderen Blöcke können als Elemente jeweils in dem ersten oder zweiten Segment modelliert werden.

[0182] Das der Interlego AG abgetretene US-Patent 5,645,463 offenbart ein weiteres Spielbauelement, das mehrere Typen von Verbindungsmitteln, z.B. Löcher zum Halten von Achsen, Verbindungsstützen und Ansätze, sowie Greifmittel zum Einschnappen von Elementen in gegenseitige Verbindung aufweist. Es ist ferner möglich, dieses Spielbauelement gemäß der Erfindung einschließlich von Modellen von Elementen zu modellieren, mit denen es verbunden werden kann. Andere Elemente können mit dem offenbarten Element mittels der oben genannten Verbindungsmittel verbunden werden. Wenn ein solches Element gemäß der Erfindung modelliert wird, ist es möglich, das Element als ein Segment zu spezifizieren, das die gegenseitige Verbindung mit anderen Segmenten an Positionen unterstützt, die mit der Position des Verbindungsmittels zusammenfallen. Dies erfolgt durch Anhängen der Achsenfelder und Drehfelder, wie oben beschrieben ist.

[0183] Das Magic Mold Corporation abgetretene US-Patent 5,061,219 offenbart ein Konstruktions-spielbauelement, das mehrere Typen von Verbindungsmitteln, wie zum Beispiel Löcher zum Halten von Achsen, nabenartige Verbinder, die eine oder mehrere Greiffassungen aufzuweisen haben, und Strukturelemente z.B. mit strebenartigem Aufbau aufweist, deren Endabschnitte konfiguriert sind, um in den Greiffassungen aufgenommen zu werden. Das Konstruktionsbauelement kann als ein Element mit vier möglichen Ausrichtungen in Bezug zu einer ersten Achse, vier mögliche Ausrichtungen in Bezug zu einer zweiten Achse, die orthogonal zu der ersten Achse ist, und acht mögliche Ausrichtungen in Bezug zu einer dritten Achse modelliert werden, die orthogonal zu der ersten und zweiten Achse ist. Das Mittelloch des nabenartigen Verbinders kann als ein Bezugspunkt zum Beschreiben der Position des Spielelements verwendet werden.

[0184] Ferner Konstruktionsspielbauelemente, die im US-Patent 5,423,707; 5,137,486; 5,199,919; 5,421,762, und 5,238,438 offenbart sind.

[0185] Es ist offensichtlich, dass die Erfindung zum Modellieren anderer modularer Bausysteme als die

im Vorhergehenden gezeigten verwendet werden kann.

[0186] Obwohl die Erfindung unter Bezugnahme auf ein rechteckiges Koordinatensystem beschrieben wurde, können andere Koordinatensysteme für die gemäß der Erfindung verwendeten Definitionen verwendet werden.

[0187] Die Erfindung ist nicht auf den gezeigten Code begrenzt; dieser Code ist als eine bevorzugte Ausführungsform gewählt worden. Daher ist die Erfindung nicht auf die speziellen Block- und Feldanweisungen (z.B. Segmentblock, Feld Übersetzen etc.) begrenzt. Die Darstellung von Parametern (z.B. ein Versatzwert von 4, ein Drehwinkel von 90 Grad, etc.) wird in dieser Offenbarung als eine einfach zu lesende und zu verstehende Darstellung in dieser Offenbarung verstanden. Es ist jedoch einer Fachperson in diesem Gebiet gut bekannt, diese Darstellung in eine ausreichende Darstellung umzuwandeln, die zum Speichern in einer Datenstruktur auf einem computerlesbaren Medium passend ist.

[0188] Die Datenstruktur kann genommen werden, um eine direkte Darstellung der Block- und Feldanweisungen einschließlich zugehöriger Parameter und Klammern zum Identifizieren des Umfangs eines Blocks zu sein. Dadurch kann die Datenstruktur z.B. als eine direkte ASCII-Darstellung gespeichert werden. Alternativ kann die Datenstruktur als eine codierte Listenartikelstruktur genommen werden, in der jeder Artikel einen Block oder ein Feld darstellt, die durch einen Binärcode identifiziert ist und in der mit den Feldern verknüpfte Parameter z.B. in einem binär codierten Format gespeichert werden. Es können jedoch andere geeignete Datenstrukturen zum Speichern der benötigten Informationen gemäß der Erfindung gewählt werden. Durch Nehmen der Datenstruktur als eine direkte Darstellung des Codes, der in den Definitionen von Blöcken und Feldern verwendet wird und in den Beispielen verwendet wird, wird die Hierarchie der Blöcke und Felder aufgrund des durch die Klammern gesetzten Umfangs der Blöcke automatisch in die Datenstruktur eingebettet werden. Diese Hierarchie kann jedoch auf andere Weisen spezifiziert werden, indem z.B. für jedes Element identifiziert wird, von welchem übergeordneten Segment, Modell und welcher Welt es einen Teil darstellt und gleichermaßen für Segmente spezifiziert wird, von welchem Modell und welcher Welt sie ein Teil sind. Alternativ ist es möglich, die Hierarchie durch Identifizieren eines übergeordneten Blocks zu spezifizieren.

[0189] In der Beschreibung sollte der Ausdruck "geometrischer Gegenstand" als ein Gegenstand der realen Welt oder einer virtuellen Welt verstanden werden, der die Modellierung gemäß der Erfindung unterzogen wird.

[0190] In der Beschreibung sollte der Ausdruck "Manipulierung" sehr breit interpretiert werden und die Aktion aufweisen, Teile der geometrischen Interpretation eines Modells zu bewegen und/oder zu drehen.

[0191] In der Beschreibung ist der Ausdruck "Meta-Beschreibung" die aktuelle Beschreibung einer Manipulation.

[0192] Der allgemeine Ausdruck "modellieren" oder "ein Modell gemäß der Erfindung" sollte nicht mit dem Ausdruck "Modellblock" verwechselt werden.

[0193] Die Grundidee der Erfindung besteht darin, dass das Modell sehr hart beschränkt wird, was – prima facie – die Fähigkeiten des Modellierungskonzepts zu begrenzen scheint. Dies ermöglicht es jedoch tatsächlich, eine sehr kompakte Darstellung eines Modells auszuwählen, welche wiederum zu einer sehr flexiblen Modellstruktur dadurch führt, dass ihre Darstellung sehr kompakt und einfach zugänglich ist, wodurch eine schnelle und effiziente Verarbeitung beim Modifizieren, Interagieren, oder Manipulieren eines erfindungsgemäßen Modells zugelassen wird. Ferner ist es möglich, ein Modell eines recht komplexen Gegenstands sehr schnell über ein Übertragungsnetz zu übertragen.

Patentansprüche

1. Verfahren zum Erzeugen eines computerlesbaren Modells eines geometrischen Gegenstands, wobei das Modell auf einem Computer mit Zugang zu einer Datenbank erzeugt wird, die Darstellungen von geometrischen Formen für eine Sammlung von Elementen einschließt, wobei das Verfahren die Schritte aufweist:

- Codieren von Bits in einer ersten Datenstruktur zum Identifizieren eines ersten Satzes von Elementen aus der Sammlung und zum Darstellen von erzwungenen Positionen der Elemente mittels ganzzahliger Koordinaten in einem ersten Koordinatensystem;
- Codieren von Bits in einer zweiten Datenstruktur zum Darstellen einer räumlichen Umwandlung des zweiten Koordinatensystems in Bezug zu einem ersten Koordinatensystem;
- Nehmen der ersten und zweiten Datenstruktur als ein computerlesbares Modell eines geometrischen Gegenstands.

2. Verfahren nach Anspruch 1, das ferner die folgenden Schritte aufweist:

- Codieren von Bits in einer dritten Datenstruktur zum Identifizieren eines zweiten Satzes von Elementen aus der Sammlung und zum Darstellen von Positionen der Elemente mittels ganzzahliger Koordinaten in einem dritten Koordinatensystem und Einschließen dieser Bits in das computerlesbare Modell;
- Codieren von Bits in der zweiten Datenstruktur zum

Darstellen einer erzwungenen, räumlichen Umwandlung des dritten Koordinatensystems in Bezug zum zweiten Koordinatensystem.

3. Verfahren nach einem der Ansprüche 1 und 2, das ferner den Schritt aufweist:

– Codieren der Bits in der zweiten Datenstruktur zum Darstellen einer Position einer Achse und zum Darstellen einer Drehung des ersten Koordinatensystems um die Achse und in Bezug zum zweiten Koordinatensystem.

4. Verfahren nach einem der Ansprüche 1 bis 3, bei dem der Schritt zum Codieren von Bits in der dritten Datenstruktur ferner den Schritt aufweist: Codieren der Bits in der zweiten Datenstruktur zum Darstellen der Position der Achse mittels ganzzahliger Koordinaten in dem zweiten Koordinatensystem.

5. Verfahren nach einem der Ansprüche 1 bis 4, bei dem der Schritt zum Codieren von Bits in der dritten Datenstruktur ferner den Schritt aufweist: Codieren der Bits in der dritten Datenstruktur zum Darstellen einer Ausrichtung der Achse durch Identifizieren einer Richtung entlang einer Achse in einem Satz vordefinierter, zueinander orthogonaler Achsen.

6. Verfahren nach einem der Ansprüche 2 bis 5, das ferner den Schritt aufweist: Codieren von Bits in der ersten und dritten Datenstruktur zum Darstellen von Ausrichtungen der Elemente in dem ersten bzw. dritten Koordinatensystem.

7. Verfahren nach einem der Ansprüche 1 bis 6, das ferner den Schritt aufweist: Codieren von Bits zum Darstellen von Ausrichtungen der Elemente durch Identifizieren einer Richtung entlang einer Achse in einem Satz vordefinierter, zueinander orthogonaler Achsen.

8. Verfahren nach einem der Ansprüche 1 bis 7, das ferner die Schritte aufweist: Codieren von Bits in einer vierten Datenstruktur zum Darstellen einer räumlichen Umwandlung des ersten und zweiten Koordinatensystems in Bezug zu einem vierten Koordinatensystem und Einschließen der vierten Datenstruktur in das computerlesbare Modell eines geometrischen Gegenstands.

9. Verfahren nach einem der Ansprüche 1 bis 8, bei dem jede der ganzzahligen Koordinaten ein dreidimensionales Volumen identifizieren kann und umgekehrt, und bei dem die Größe des dreidimensionalen Volumens angepasst wird, um eine Positionsdarstellung von zwei Elementen mittels der ganzzahligen Koordinaten derart zu ermöglichen, dass die beiden Elemente zum Bilden eines kohärenten, geometrischen Gegenstands verbunden werden.

10. Verfahren nach einem der Ansprüche 1 bis 9,

bei dem die Elemente Darstellungen geometrischer Formen aufweisen, die gut bekannten Spielbauelementen ähneln.

11. Verfahren nach einem der Ansprüche 1 bis 10, bei dem die Elemente Darstellungen geometrischer Formen aufweisen, von denen zumindest ein Teil der Formen ausgelegt ist, um Verbindungsmitteln oder Visualisierungen möglicher Verbindungen von Elementen zu ähneln.

12. Verfahren nach einem der Ansprüche 1 bis 11, bei dem die Darstellungen geometrischer Formen für eine Sammlung von Elementen Koordinaten aufweisen, die durch Zahlen des Typs gegeben sind, der real in einem Koordinatensystem ist.

13. Verfahren nach einem der Ansprüche 1 bis 12, das ferner die Schritte aufweist: Codieren der Bits in der dritten Datenstruktur zum Darstellen der ganzzahligen Koordinaten in Bezug zu einem Ursprung in dem zweiten Koordinatensystem, wobei der Ursprung eine Koordinate der Verbindung mit Elementen in dem ersten Koordinatensystem definiert; Definieren eines neuen Ursprungs im dritten Koordinatensystem als Reaktion auf die neue Koordinate der Verbindung in dem zweiten Koordinatensystem; Berechnen neuer Positionen der Elemente innerhalb des dritten Koordinatensystems infolge des neuen Ursprungs; Codieren der Bits in der dritten Datenstruktur zum Darstellen des Ergebnisses der Berechnung.

14. Verfahren nach einem der Ansprüche 3 bis 13, das ferner die Schritte aufweist: Codieren der Bits in der ersten Datenstruktur zum Darstellen der Position der Achse; dadurch Definieren einer Koordinate der Verbindung mit Elementen in dem zweiten Koordinatensystem im ersten Koordinatensystem; Definieren einer neuen Koordinate der Verbindung in dem ersten Koordinatensystem zum Darstellen einer neuen Position von Elementen innerhalb des ersten Koordinatensystems in Bezug zum zweiten Koordinatensystem, und Codieren der Bits in der ersten Datenstruktur zum Darstellen der neuen Position der Achse, wobei die neue Position die neue Koordinate der Verbindung ist.

15. Verfahren nach einem der Ansprüche 1 bis 14, das ferner die Schritte aufweist: Ausführen des Verfahrens auf einem Computer mit einer Benutzerschnittstelle, die Benutzersteuerungen einschließt, welche eine Interaktion zwischen einem Benutzer und der Benutzerschnittstelle darstellende Signale liefert, wobei die Interaktion Modifikation und/oder Manipulation einer geometrischen Darstellung des computerlesbaren Modells aufweist, und Konvertieren der Signale in ein computerlesbares Modell oder ein modifiziertes, computerlesbares Modell.

16. Verfahren nach einem der Ansprüche 1 bis 15, das ferner den Schritt aufweist: Übertragen der codierten Bits oder des computerlesbaren Modells über ein an den Computer angeschlossenes Kommunikationsnetz.

17. Verfahren nach Anspruch 16, das ferner den Schritt aufweist: Empfangen der codierten Bits oder des computerlesbaren Modells auf einem Computer über ein an den Computer angeschlossenes Kommunikationsnetz.

18. Verfahren nach einem der Ansprüche 1 bis 17, das ferner die Schritte aufweist:
 – Verbinden mit einer lokalen Datenbank und einer Ferndatenbank für Datenübertragung;
 – erstens für jedes Element durch Konsultieren der lokalen Datenbank Prüfen, ob die Darstellung der geometrischen Form lokal erhältlich ist;
 – zweitens, wenn die Darstellung der geometrischen Form lokal erhältlich ist, Auslesen der geometrischen Form von der lokalen Datenbank, oder alternativ, wenn die geometrische Form nicht lokal erhältlich ist, Auslesen der geometrischen Form von der Ferndatenbank, wenn die geometrische Form in der Ferndatenbank vorliegt.

19. Verfahren nach einem der Ansprüche 1 bis 18, das ferner die Schritte aufweist: Erzeugen anzeigbarer Daten, die die geometrischen Formen darstellen, wie ihr räumliches Verhältnis gemäß der ersten, zweiten und dritten Datenstruktur beschrieben ist.

20. Computerlesbares Medium mit einem darauf aufgezeichneten Programm, wobei das Programm – bei Ausführung – den Computer veranlassen soll, das Verfahren gemäß einem der Ansprüche 1 bis 19 auszuführen.

21. Computersystem mit Mitteln zum Ausführen des Verfahrens gemäß einem der Ansprüche 1 bis 19.

22. Computerlesbares Modell eines geometrischen Gegenstands mit Datenstrukturen zum Verwenden des Verfahrens gemäß einem der Ansprüche 1 bis 19.

23. Verfahren zum Interpretieren eines computerlesbaren Modells eines geometrischen Gegenstands, wobei das Modell auf einem Computer mit einer Benutzerschnittstelle interpretiert wird, und der Computer Zugang zu einer Datenbank hat, die Darstellungen geometrischer Formen für eine Sammlung von Elementen einschließt, wobei das Verfahren die Schritte aufweist:

– Decodieren von Bits in einer ersten Datenstruktur zum Identifizieren eines ersten Satzes von Elementen von der Sammlung und zum Identifizieren von

Positionen der Elemente mittels ganzzahliger Koordinaten in einem ersten Koordinatensystem;

– Decodieren von Bits in einer zweiten Datenstruktur zum Identifizieren einer räumlichen Umwandlung des ersten Koordinatensystems in Bezug zum ersten Koordinatensystem;

– Auslesen der identifizierten Darstellungen geometrischer Formen und Erzeugen anzeigbarer Daten, die die geometrische Form darstellen, wie ihr räumliches Verhältnis gemäß der ersten und zweiten Datenstruktur beschrieben ist.

24. Verfahren nach Anspruch 23, das ferner den Schritt aufweist:

– Decodieren von Bits in einer dritten Datenstruktur zum Identifizieren eines zweiten Satzes von Elementen aus der Sammlung und Identifizieren von Positionen der Elemente mittels ganzzahliger Koordinaten in dem dritten Koordinatensystem und Einschließen dieser Bits in das computerlesbare Modell;

– Decodieren der Bits in der zweiten Datenstruktur zum Identifizieren einer erzwungenen, räumlichen Umwandlung des dritten Koordinatensystems in Bezug zum zweiten Koordinatensystem.

25. Verfahren nach einem von Anspruch 23 oder 24, das ferner die Schritte aufweist:

– Decodieren der Bits in der zweiten Datenstruktur zum Identifizieren einer Position einer Achse und zum Identifizieren einer Drehung des ersten Koordinatensystems um die Achse und in Bezug zum zweiten Koordinatensystem.

26. Verfahren nach einem der Ansprüche 23 und 24, das ferner den Schritt aufweist: Decodieren der Bits in der zweiten Datenstruktur zum Identifizieren der Position der Achse mittels ganzzahliger Koordinaten in dem zweiten Koordinatensystem.

27. Verfahren nach einem der Ansprüche 23 bis 26, das ferner den Schritt aufweist: Decodieren der Bits in der dritten Datenstruktur zum Identifizieren einer Ausrichtung der Achse durch Identifizieren einer Richtung entlang einer Achse in einem Satz vordefinierter, zueinander orthogonaler Achsen.

28. Verfahren nach einem der Ansprüche 24 bis 27, das ferner den Schritt aufweist: Decodieren von Bits in der ersten und dritten Datenstruktur zum Identifizieren von Ausrichtungen der Elemente in dem ersten bzw. dritten Koordinatensystem.

29. Verfahren nach Anspruch 28, das ferner den Schritt aufweist: Decodieren von Bits zum Identifizieren von Ausrichtungen der Elemente durch Lesen einer Identifizierung einer Richtung entlang einer Achse in einem Satz vordefinierter, zueinander orthogonaler Achsen.

30. Verfahren nach einem der Ansprüche 23 bis

29, das ferner die Schritte aufweist: Decodieren von Bits in einer vierten Datenstruktur zum Identifizieren einer räumlichen Umwandlung des ersten und zweiten Koordinatensystems in Bezug zu einem vierten Koordinatensystem; und Modifizieren der anzeigbaren Daten, um gemäß der räumlichen Umwandlung zu erscheinen.

31. Verfahren nach einem der Ansprüche 23 bis 30, bei dem jede der ganzzahligen Koordinaten ein dreidimensionales Volumen identifizieren kann und umgekehrt, und bei dem die Größe des dreidimensionalen Volumens ausgelegt ist, um eine anzeigbare Position von zwei Elementen mittels der ganzzahligen Koordinaten zu ermöglichen, so dass die beiden Elemente verbunden werden, um einen kohärenten geometrischen Gegenstand zu bilden.

32. Verfahren nach einem der Ansprüche 23 bis 31, bei dem die Elemente Darstellungen geometrischer Formen aufweisen, die gut bekannten Spielbauelementen ähneln.

33. Verfahren nach einem der Ansprüche 23 bis 32, bei dem die Elemente Darstellungen geometrischer Formen aufweisen, von denen zumindest ein Teil der Formen ausgelegt ist, um Verbindungsmittel oder Visualisierungen möglicher Zwischenverbindungen von Elementen zu ähneln.

34. Verfahren nach einem der Ansprüche 23 bis 33, bei dem die Darstellungen geometrischer Formen für eine Sammlung von Elementen Koordinaten aufweisen, die durch Zahlen des in einem Koordinatensystem realen Typs gegeben sind.

35. Verfahren nach einem der Ansprüche 23 bis 32, das ferner die Schritte aufweist:

- Decodieren der Bits in der dritten Datenstruktur zum Darstellen der ganzzahligen Koordinaten in Bezug zu einem Ursprung in dem zweiten Koordinatensystem, wobei der Ursprung eine Koordinate von Verbindung mit Elementen in dem ersten Koordinatensystem definiert;
- Definieren eines neuen Ursprungs in dem dritten Koordinatensystem als Reaktion auf eine neue Koordinate der Verbindung in dem zweiten Koordinatensystem;
- Berechnen neuer Positionen der Elemente innerhalb des zweiten Koordinatensystems infolge des neuen Ursprungs;
- Codieren der Bits in der dritten Datenstruktur zum Darstellen des Berechnungsergebnisses.

36. Verfahren nach einem der Ansprüche 24 bis 33, das ferner die Schritte aufweist:

- Decodieren der Bits in der ersten Datenstruktur zum Darstellen der Position der Achse, wodurch eine Koordinate von Verbindung in dem ersten Koordinatensystem definiert wird;

– Definieren einer neuen Koordinaten der Verbindung in dem ersten Koordinatensystem zum Darstellen einer neuen Position von Elementen innerhalb des ersten Koordinatensystems in Bezug zum zweiten Koordinatensystem;

– Codieren der Bits in der ersten Datenstruktur zum Darstellen der neuen Position der Achse, wobei die neue Position die neue Koordinate der Verbindung ist.

37. Verfahren nach einem der Ansprüche 27 bis 34, das ferner die Schritte aufweist:

- Ausführen des Verfahrens auf einem Computer mit einer Benutzerschnittstelle, die Benutzersteuerungen einschließt, welche eine Interaktion zwischen einem Benutzer und der Benutzerschnittstelle darstellende Signale liefert, wobei die Interaktion Modifikation und/oder Manipulation einer geometrischen Darstellung des computerlesbaren Modells aufweist;
- Konvertieren der Signale in ein computerlesbares Modell oder ein modifiziertes, computerlesbares Modell.

38. Verfahren nach einem der Ansprüche 23 bis 37, das ferner den Schritt aufweist: Konvertieren des computerlesbaren Modells in anzeigbare Daten durch Berechnen einer Projektion der geometrischen Formen, wie sie gemäß der Datenstrukturen erscheinen, auf eine Ebene oder gekrümmte Fläche.

39. Verfahren nach einem der Ansprüche 23 bis 38, das ferner den Schritt aufweist: Lesen des computerlesbaren Modells aus einem Binärdateityp, der das computerlesbare Modell darstellt.

40. Verfahren nach einem der Ansprüche 23 bis 37, das ferner die Schritte aufweist: Übertragen der Bits oder des computerlesbaren Modells über ein an den Computer angeschlossenes Kommunikationsnetz.

41. Verfahren nach Anspruch 40, das ferner die Schritte aufweist: Empfangen der Bits oder des computerlesbaren Modells auf einem Computer über ein an den Computer angeschlossenes Kommunikationsnetz.

42. Verfahren nach einem der Ansprüche 23 bis 41, das ferner die Schritte aufweist:

- Lösen der Datenbank in eine lokale Datenbank und eine Ferndatenbank;
- erstens, Prüfen für jedes Element unter Befragung der lokalen Datenbank, ob die Darstellung der geometrischen Form lokal erhältlich ist;
- zweitens, wenn die geometrische Form lokal erhältlich ist, Auslesen der geometrischen Form von der lokalen Datenbank, oder, wenn die geometrische Form nicht lokal erhältlich ist, Auslesen der geometrischen Form von der Ferndatenbank, wenn die Form in der Ferndatenbank vorliegt.

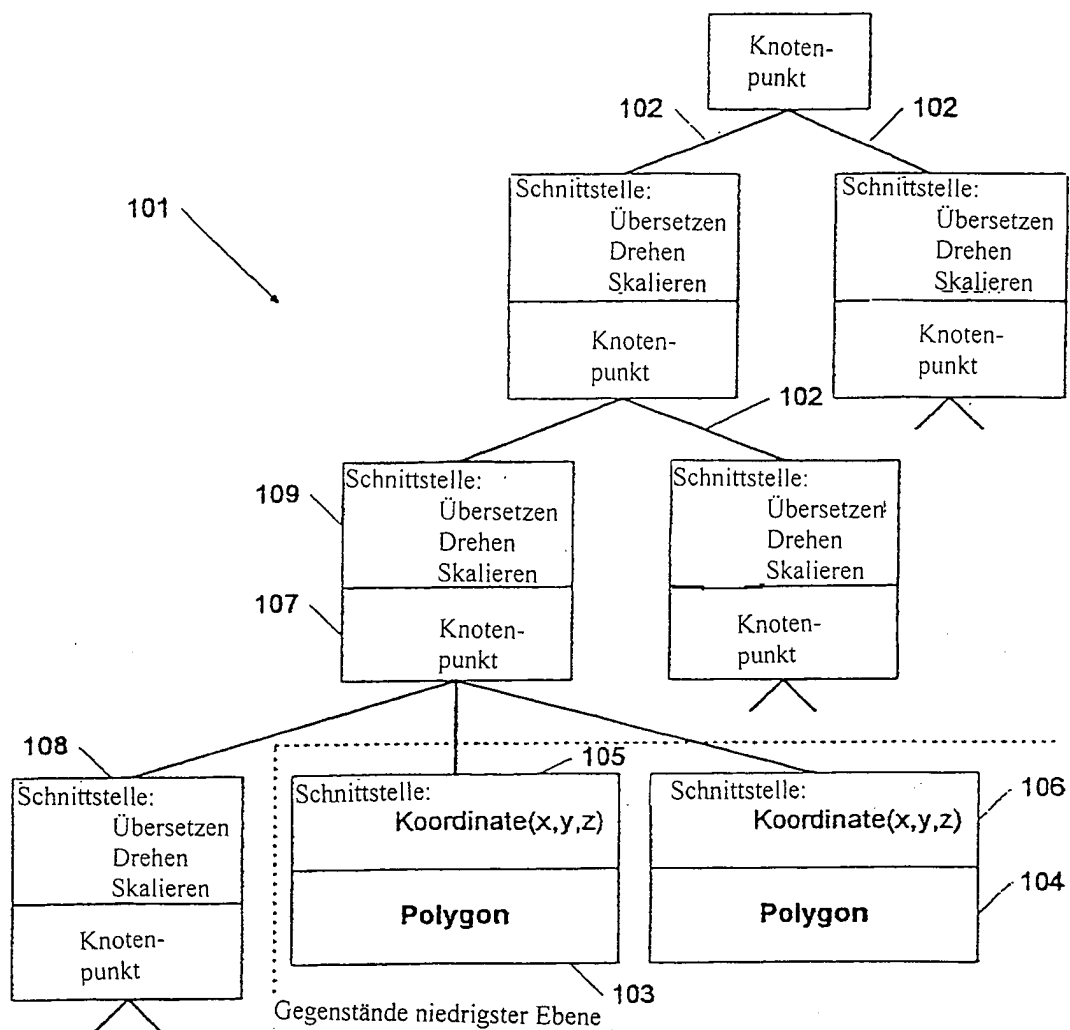
43. Computerlesbares Medium mit einem darauf aufgezeichneten Programm, wobei das Programm – bei Ausführung – den Computer veranlassen soll, das Verfahren gemäß einem der Ansprüche 23 bis 42 auszuführen.

44. Computersystem mit Mitteln zum Ausführen des Verfahrens gemäß einem der Ansprüche 23 bis 42.

45. Computerlesbares Modell eines geometrischen Gegenstands mit Datenstrukturen zur Verwendung des Verfahrens gemäß einem der Ansprüche 23 bis 42.

Es folgen 12 Blatt Zeichnungen

Anhängende Zeichnungen



Stand der Technik

Fig. 1

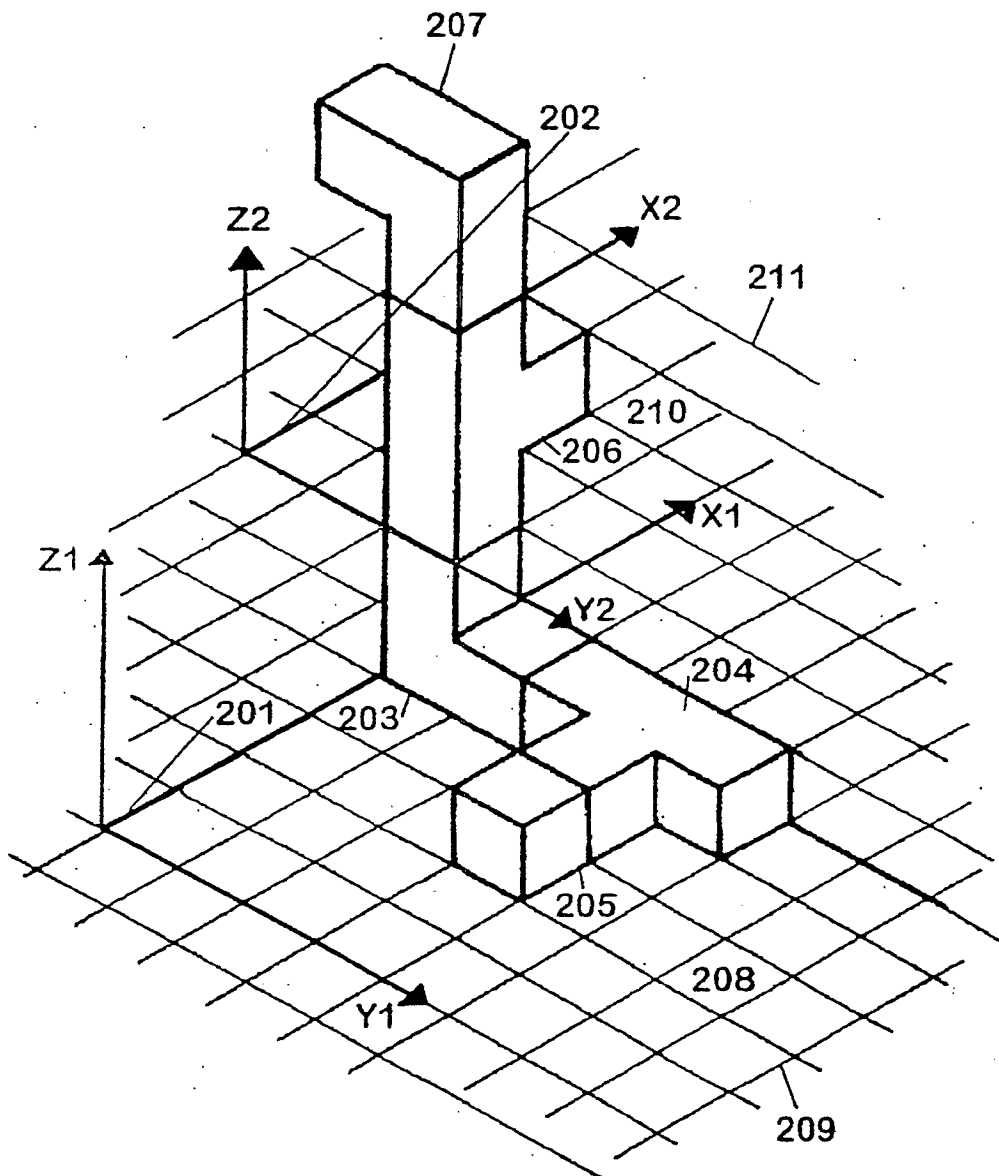


Fig. 2a

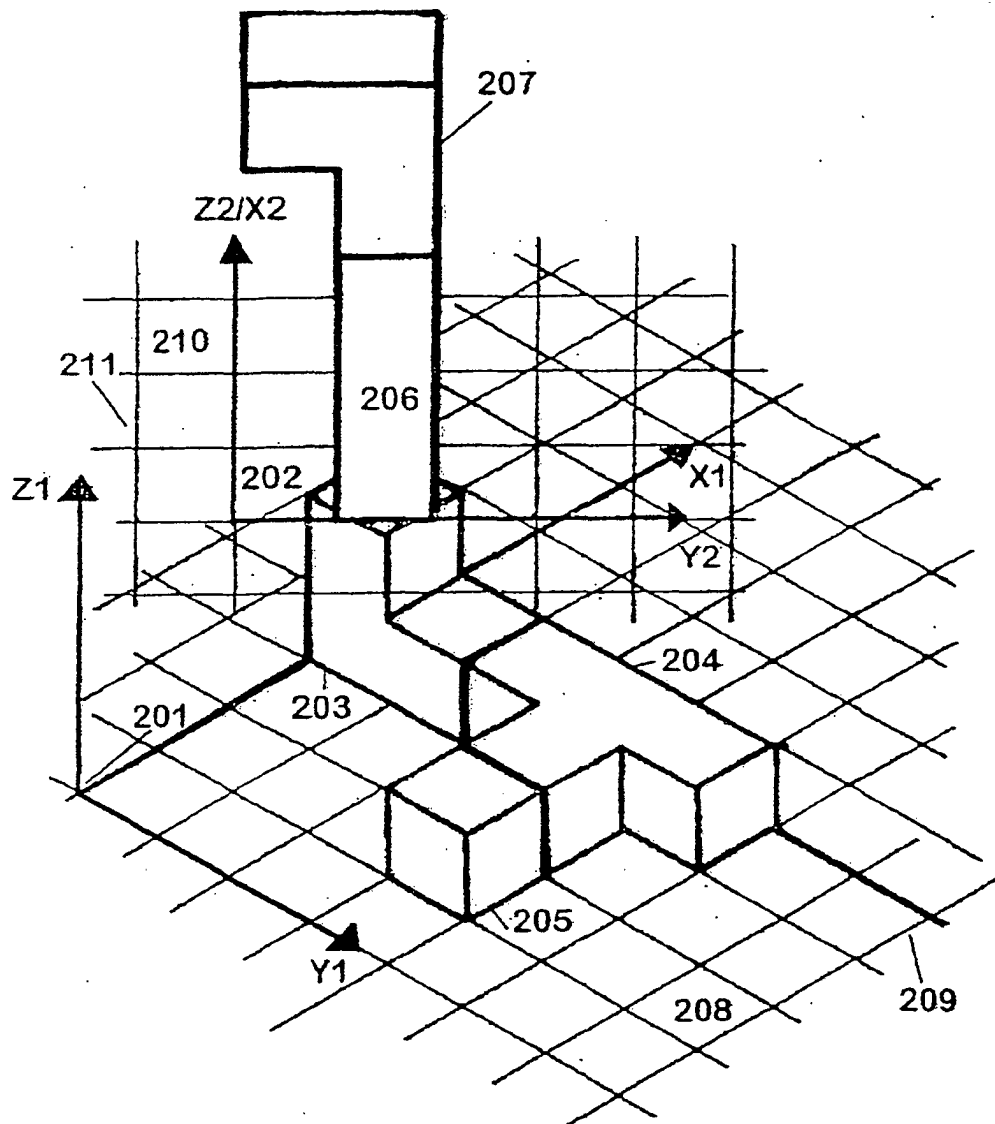


Fig. 2b

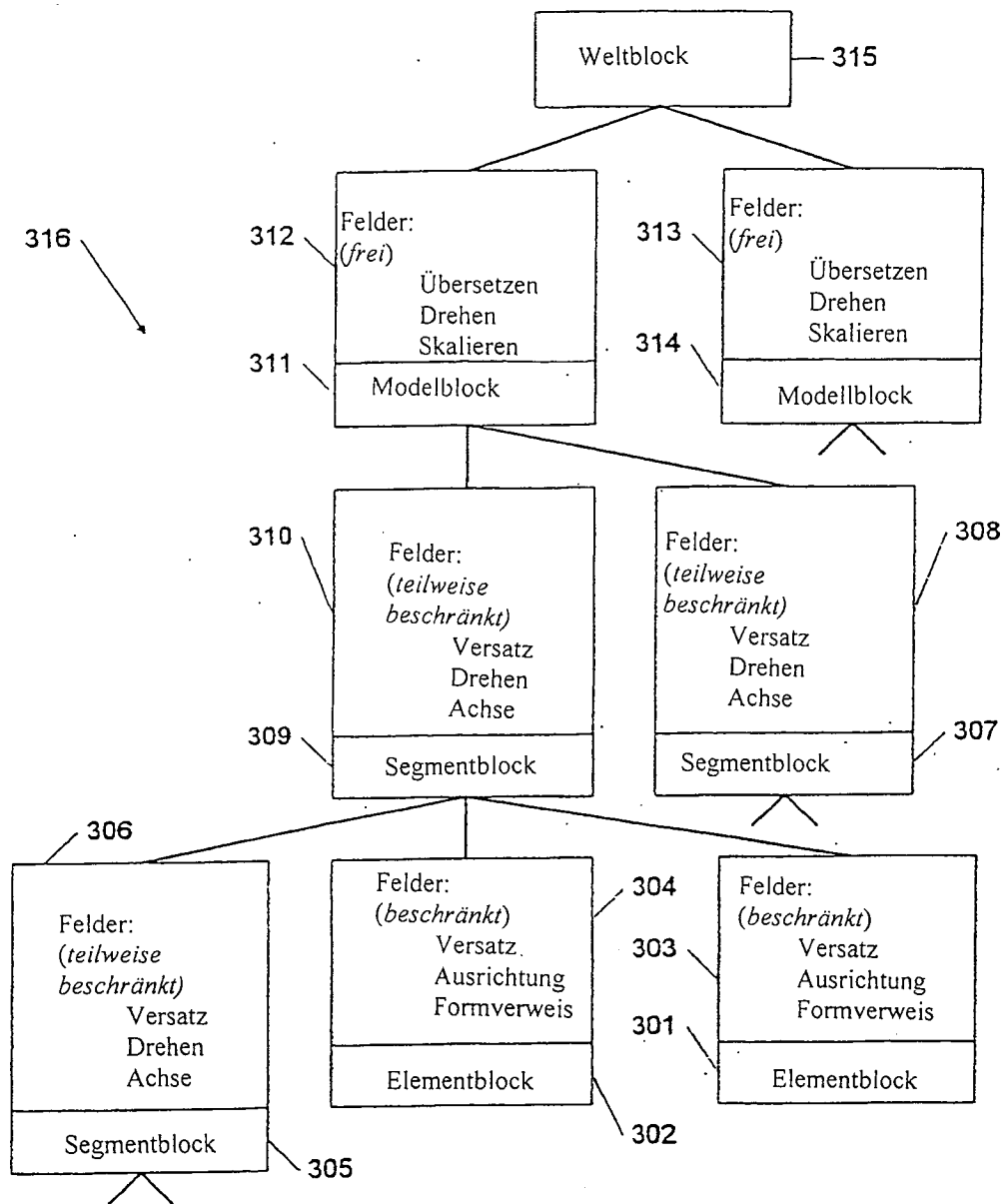


Fig. 3

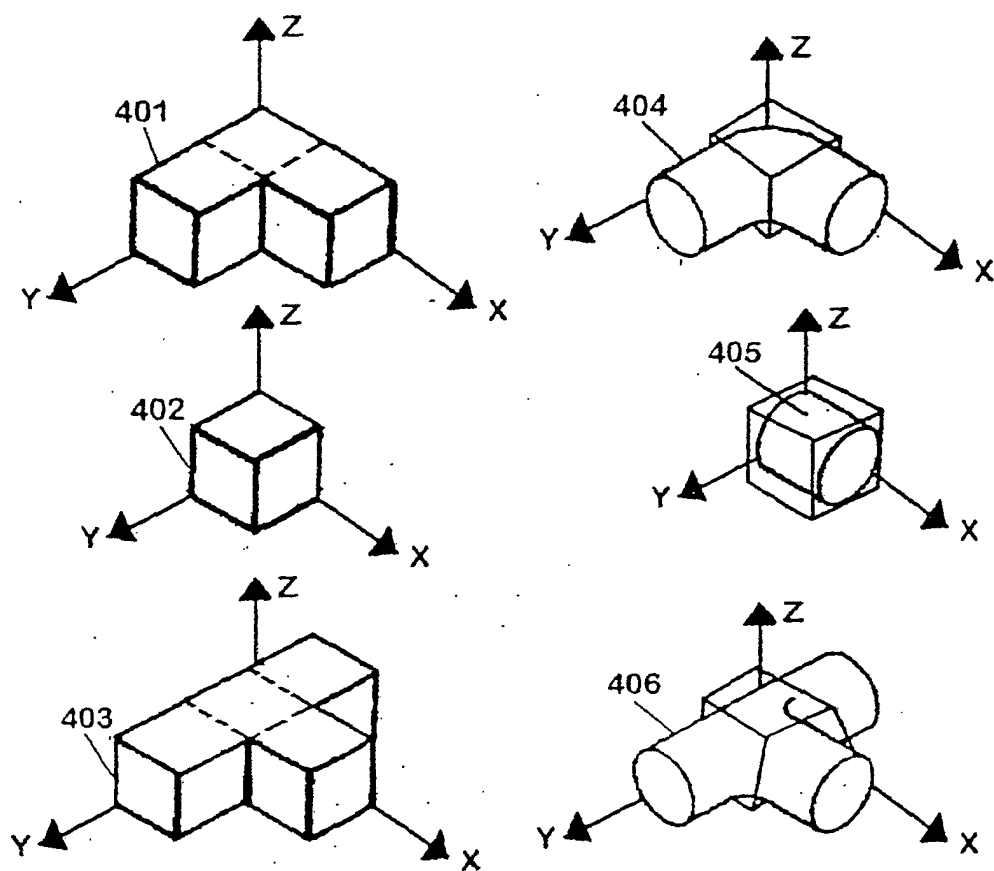


Fig. 4

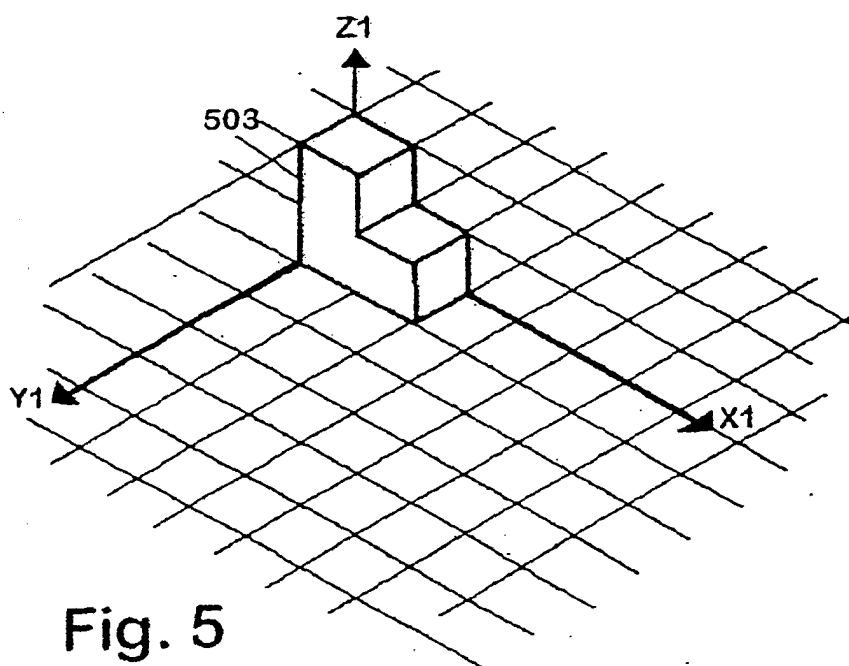


Fig. 5

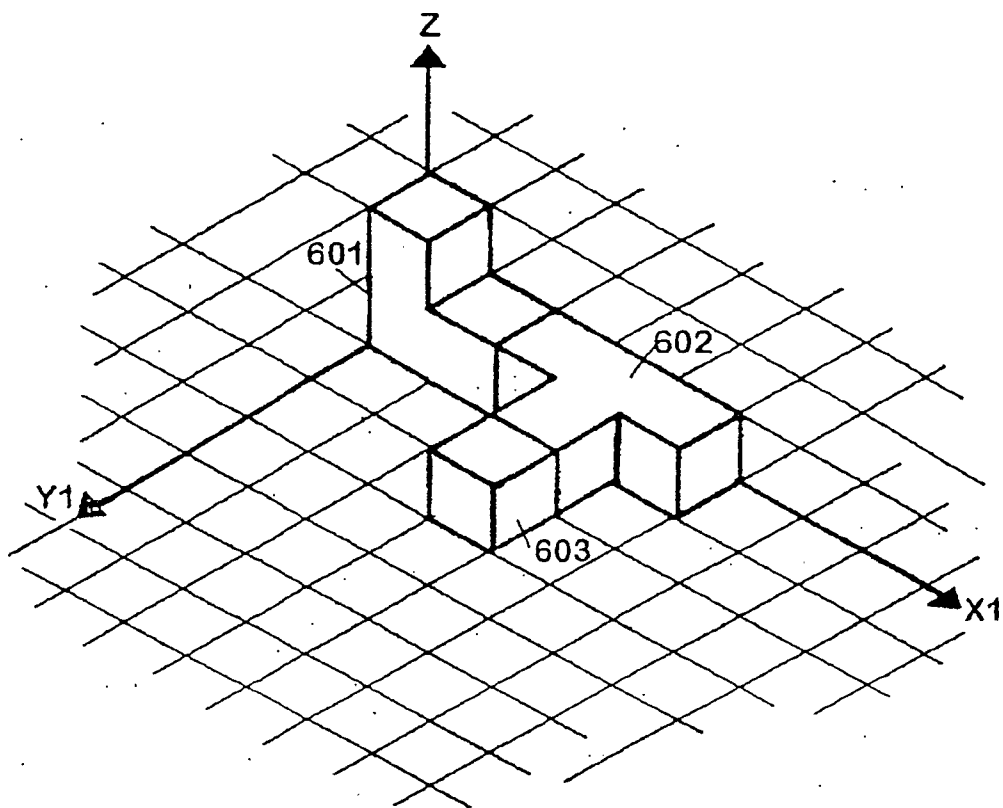


Fig. 6

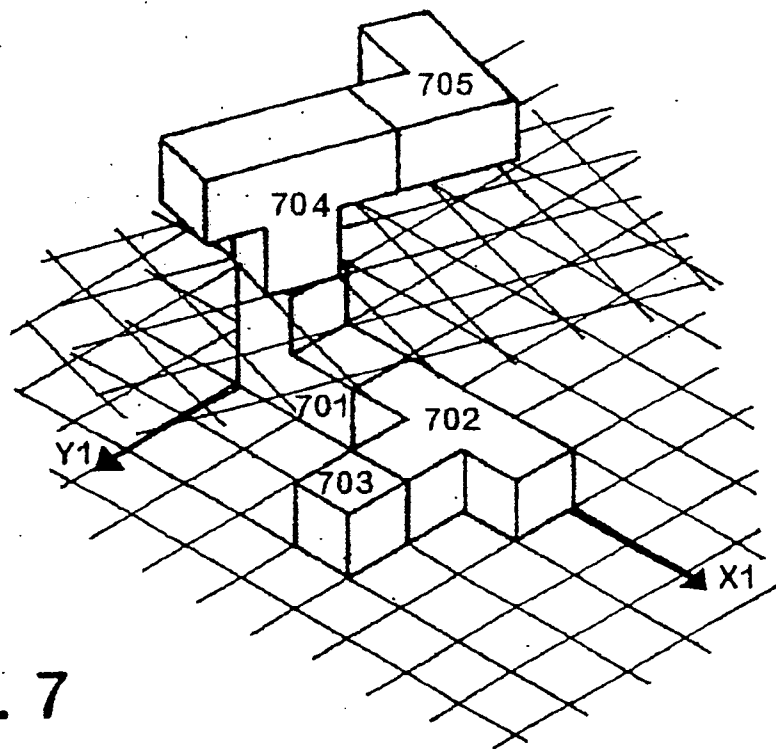


Fig. 7

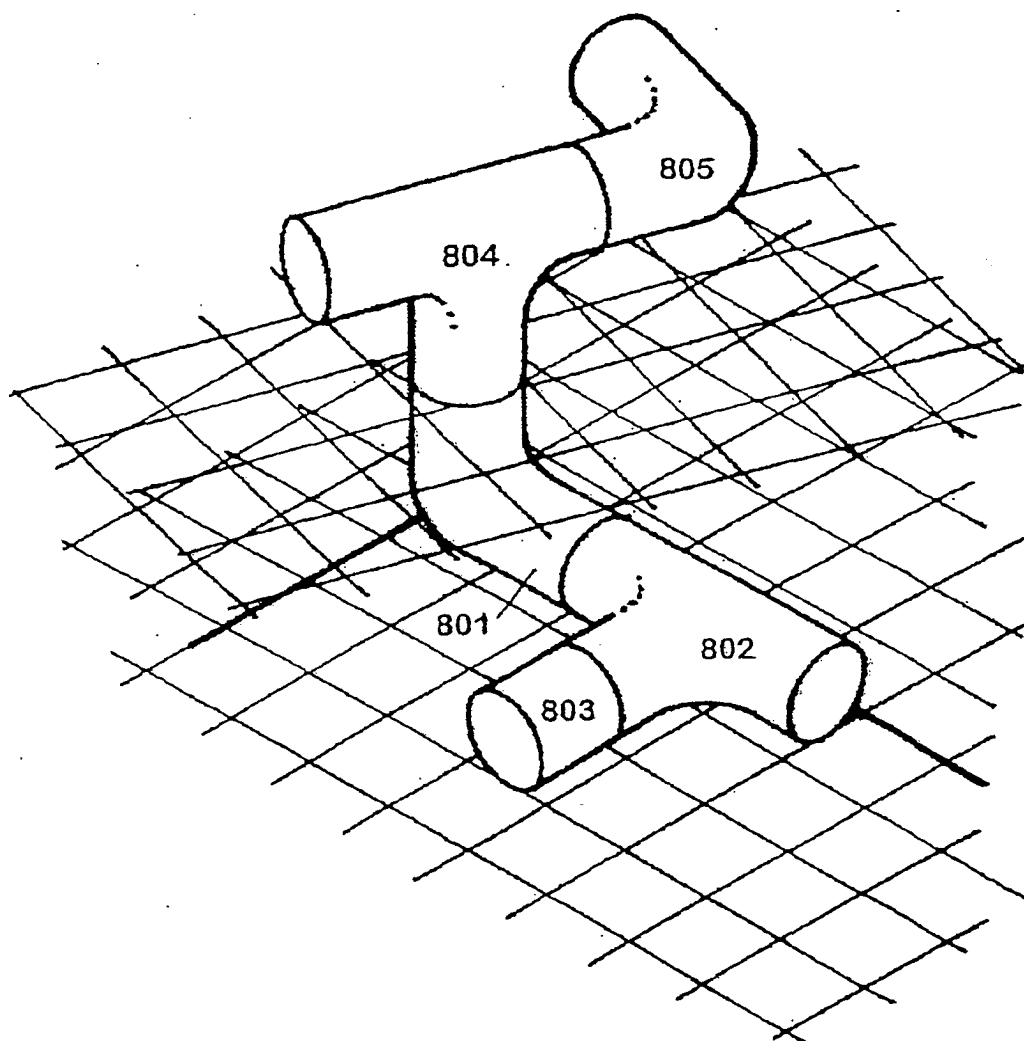


Fig. 8

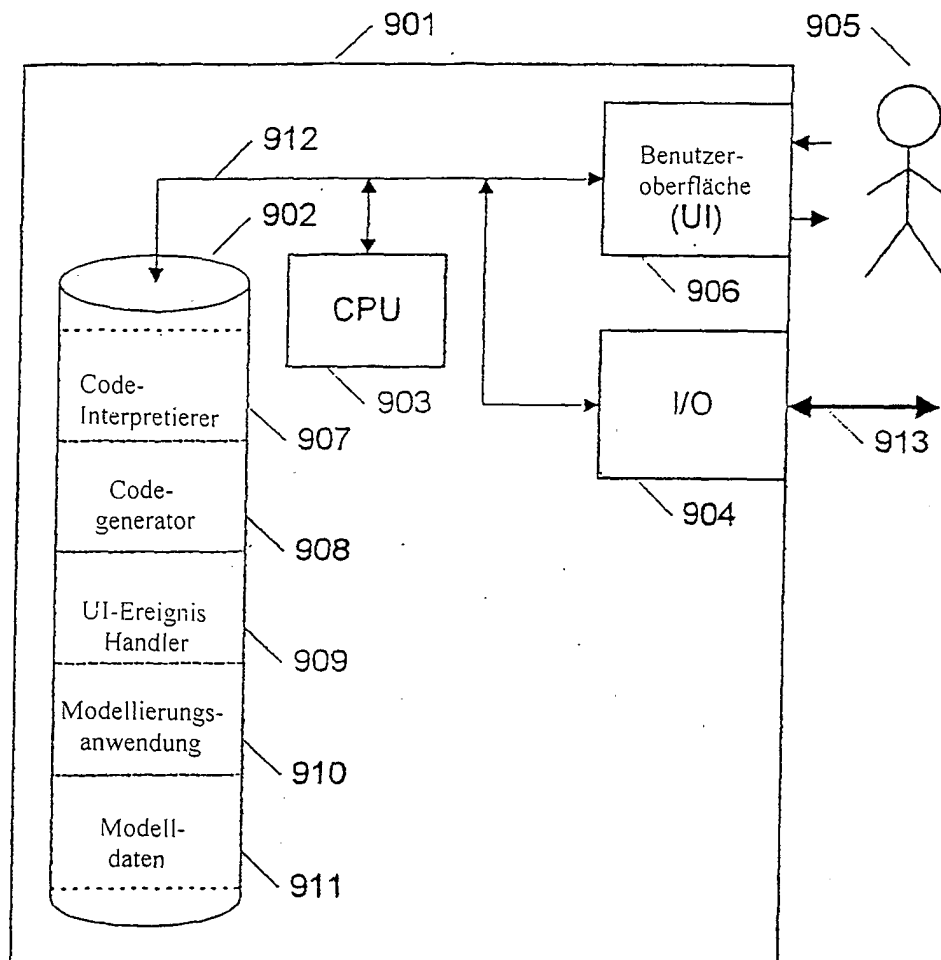


Fig. 9

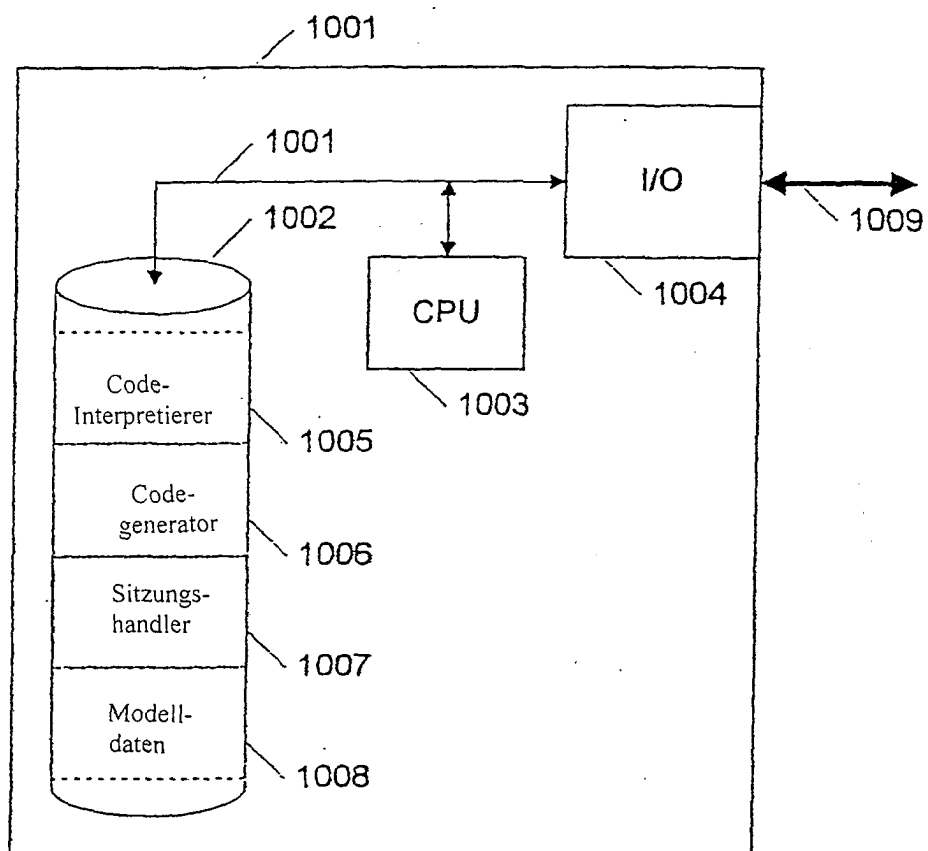
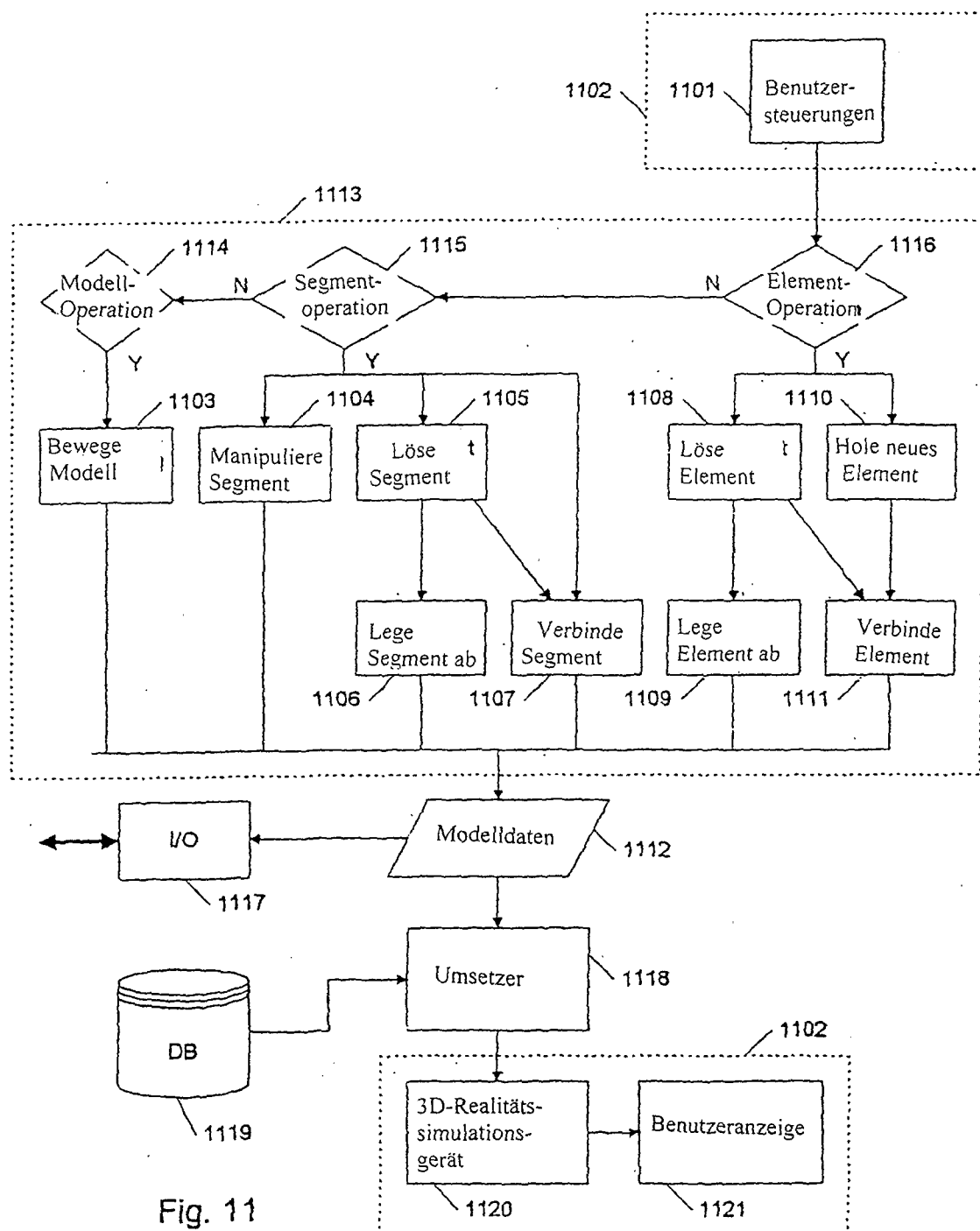


Fig. 10



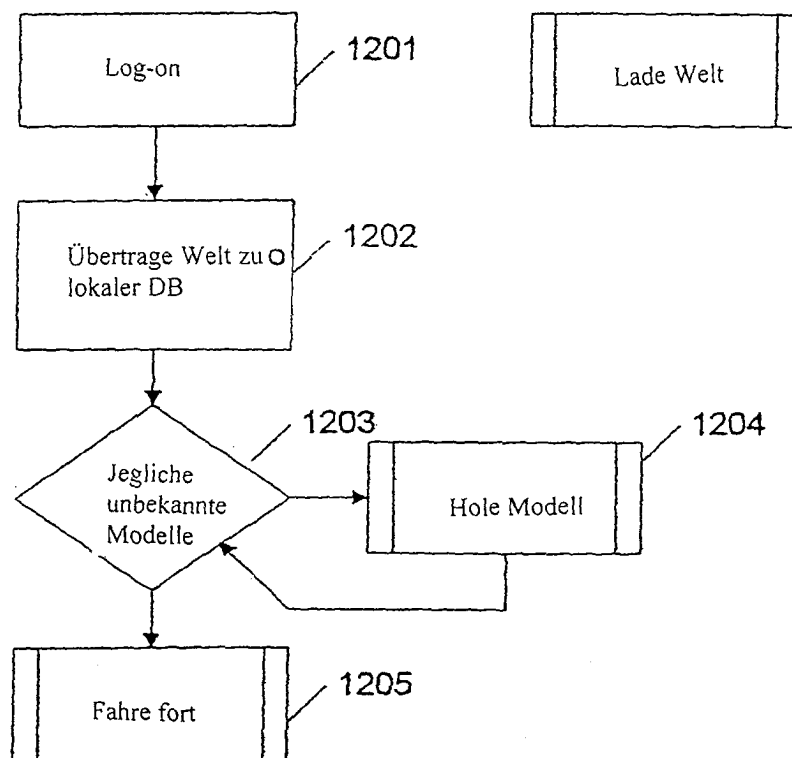


Fig. 12

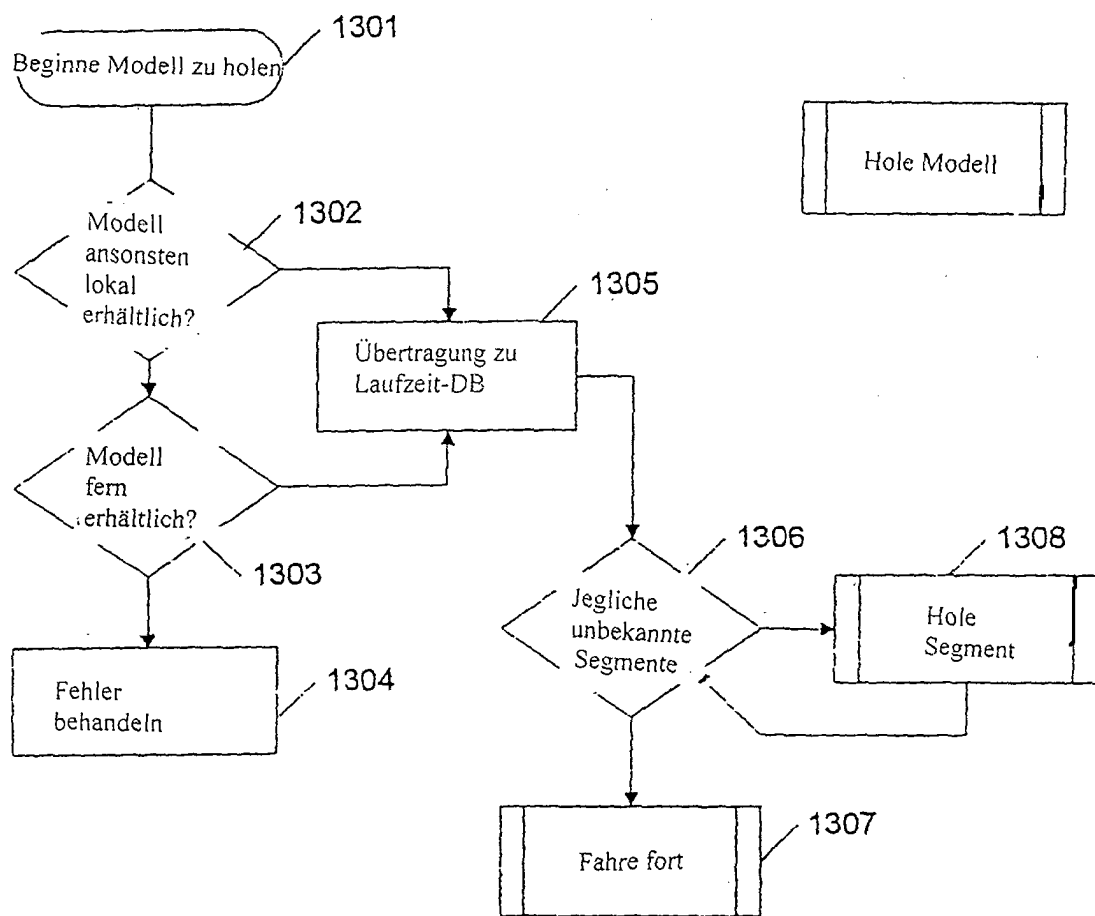


Fig. 13