(19) LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de l'Économie

(11) N° de publication : **LU101410**

(12) **BREVET D'INVENTION** **B1**

(21) N° de dépôt: LU101410

(22) Date de dépôt: 25/09/2019

(51) Int. Cl.:
G06F 16/28, H04L 29/08, H04L 29/12

(30) Priorité:

(43) Date de mise à disposition du public: 25/03/2021

(47) Date de délivrance: 25/03/2021

(73) Titulaire(s):
PHOENIX CONTACT GmbH & Co. Kg – 32825 Blomberg (Allemagne)

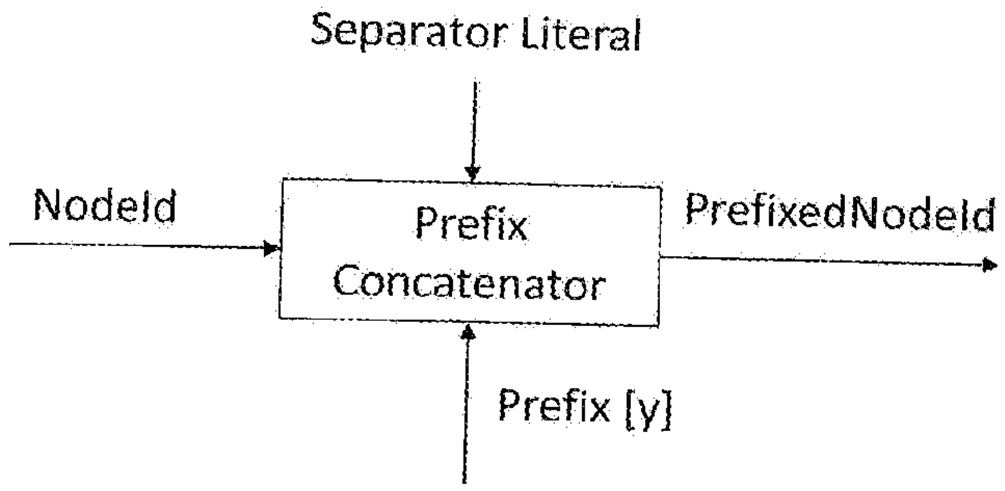
(72) Inventeur(s):
THEIS Oliver – 32689 Kalletal (Allemagne)

(74) Mandataire(s):
PHOENIX CONTACT GMBH & CO. KG – 32825 Blomberg (Allemagne)

(54) Verfahren zum Bereitstellen und Validieren alternativer Objektbezeichner sowie zum Ermitteln von Referenzen alternativer Objektbezeichner innerhalb einer OPC-UA-basierenden Kommunikationsumgebung.

(57) Vorgeschlagen wird ein Verfahren zum Bereitstellen alternativer Objektbezeichner für Objekte eines Adressraums eines Servers (S1, S2) innerhalb einer OPC-UA-basierenden Kommunikationsumgebung (K), wobei jedes Objekt des Adressraums des Servers (S1, S2) durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist, wobei wenigstens ein Zusatzbezeichner definiert und in einem Verzeichnis der für diesen Adressraum des Servers (S1, S2) definierten Zusatzbezeichner gespeichert wird, und wobei für jedes Objekt des Adressraum des Servers (S1, S2) ein das Objekt ebenfalls eindeutig bezeichnender alternativer Objektbezeichner gebildet wird, indem der Zusatzbezeichner mit dem herkömmlichen Objektbezeichner des jeweiligen Objekts nach einem bestimmten Schema kombiniert wird. Vorgeschlagen werden ferner ein Verfahren zum Validieren alternativer Objektbezeichner und ein Verfahren zum Ermitteln von Referenzen alternativer Objektbezeichner sowie ein Server, ein System und ein Computerprogrammprodukt ausgebildet zur Durchführung wenigstens eines der Verfahren.

Fig. 2



Verfahren zum Bereitstellen und Validieren alternativer Objektbezeichner sowie zum Ermitteln von Referenzen alternativer Objektbezeichner innerhalb einer OPC-UA-basierenden Kommunikationsumgebung

LU101410

Beschreibung

Die vorliegende Erfindung betrifft ein Verfahren zum Bereitstellen alternativer Objektbezeichner für Objekte eines Adressraums eines Servers innerhalb einer OPC-UA-basierenden Kommunikationsumgebung, ein Verfahren zum Validieren solcher alternativen Objektbezeichner sowie ein Verfahren zum Ermitteln von Referenzen solcher alternativer Objektbezeichner. Die Erfindung betrifft einen Server, ein System und ein Computerprogrammprodukt, welche ausgebildet sind zur Durchführung wenigstens eines dieser Verfahren.

IEC 62541 ist ein mehrteiliger Standard für die OPC Unified Architecture (OPC UA), welcher z.B. Protokolle für die sogenannte Machine-to-Machine (M2M) Kommunikation definiert, also für den automatisierten Informationsaustausch zwischen Endgeräten wie Maschinen oder Automaten.

OPC-UA unterliegt einem Client-Server-Modell, d. h. Server stellen Informationen zur Verfügung, die Clients von den Servern abfragen. Die Funktionalität eines Servers wird den Clients anhand sogenannter Services verfügbar gemacht. Ein Service ermöglicht den Zugriff auf bestimmte Serveraspekte mittels für den jeweiligen Service definierter Anfrage- und Antwortnachrichten (request/response), die zwischen dem Client und dem Server ausgetauscht werden. Dabei werden die Informationen von einem Server den Clients in einem Adressraum des Servers bereitgestellt.

Teil 3 des IEC 62541 Standards („Address Space Model“) definiert ein objektorientiertes Konzept eines Adressraumes, insbesondere gemäß der zum Anmeldetag gültigen Version, wobei gemäß Teil 4 des Standards („Services“), insbesondere gemäß der zum Anmeldetag gültigen Version, Objekte durch Knoten (nodes) repräsentiert werden. Knoten haben zum einen Attribute und zum anderen Referenzen, welche auf andere Knoten zeigen und die Kanten eines gerichteten Graphen repräsentieren, so dass es Vorwärts- und Rückwärtsreferenzen geben kann.

Clients in einer UPC-UA-basierenden Kommunikationsumgebung können einen solchen durch einen Server bereitgestellten Adressraum durchsuchen (browse), um ausgehend von einem bekannten Wurzelknoten den eindeutigen Bezeichner eines anderen Knotens (nachfolgend auch NodeID genannt) zu ermitteln. Mittels der spezifizierten Dienste (services) wie „DataAccess“ erhalten OPC-UA-Clients beispielsweise lesenden oder schreibenden Zugriff auf Attributwerte eines Knotens eines Adressraums.

NodeIDs setzen sich in der Regel aus den drei Elementen „namespaceIndex“, „identifierType“ und „identifier“ zusammen. Das Element „identifierType“ definiert den Typ des eindeutigen Bezeichners, welcher NUMERIC, STRING, GUID oder BYTESTRING sein kann. Für die Eindeutigkeit jeder NodeID muss jedem Knoten des Adressraums (address space) als Element „identifier“ ein eindeutiger Identifikator gemäß des definierten Bezeichnertyps festgelegt sein. Um Adress- bzw. Bezeichnerkollisionen zu vermeiden und den Adressraum weiter zu unterteilen, kann ein Adressraum mehrere Namensräume (name space) umfassen. Als Element „namespaceIndex“ der NodeID muss für jeden Knoten des Adressraums der jeweilige Namensraum des Knotens angegeben sein. Ein Verzeichnis aller Namensräume bzw.

Namensraumindizies eines Adressraums ist als Attribut zu einem bekannten Knoten wie dem Wurzelknoten hinterlegt.

Eine mögliche Anwendung eines OPC-UA-Servers ist es zum Beispiel im Bereich der Automatisierungstechnik, OPC-UA-Clients den Zugriff auf Variablen einer speicherprogrammierbaren Steuerung (programmable logic control, PLC) bereitzustellen. Beispielsweise werden bei Steuerungen auf Basis der PLCnext Technology der Anmelderin die Variablen der Komponenten als Knoten eines Adressraumes eines OPC-UA-Servers aufgeführt, wobei für die eindeutigen Knotenbezeichner der Bezeichnertyp STRING festgelegt ist (vgl. Figur 8). Das Verzeichnis aller Namensräume bzw. Namensraumindizies des Adressraums ist als Attribut „NamespaceArray" des Knotens „Server" hinterlegt (vgl. Figur 9).

Problematisch wurde es bisher aber, wenn für den selben Datenbestand eines OPC-UA-Adressraums mehrere Informationsmodelle wie zum Beispiel verschiedene semantische Repräsentationen insbesondere dynamisch implementiert werden sollten oder im Laufe der Zeit Änderungen im Adressraum vorgenommen werden sollten, die Auswirkungen auf bestehende NodeIDs hatten.

Die bereits erwähnten Namensräume erlauben zwar eine Strukturierung eines Adressraums. Sie sind aber statischer Natur und würden ein Replizieren von Knoten bzw. Knoteninstanzen in einen anderen Namensraum erfordern, um für den selben Datenbestand eines OPC-UA-Adressraums mehrere Informationsmodelle zu implementieren.

Das Konzept der OPC-UA-Sichten (views) erlaub zwar das Organisieren und Zuschneiden von Adressräumen auf spezifische Aufgaben und Anwendungsfälle. Sie dienen aber eher als Einstiegspunkte oder Filter für einen Adressraum.

Eine Aufgabe der vorliegenden Erfindung liegt somit darin, eine Möglichkeit zur Beseitigung wenigstens eines der genannten Nachteile anzugeben.

Als Lösung schlägt die Erfindung ein Verfahren mit den Merkmalen des unabhängigen Patentanspruchs 1, ein Verfahren mit den Merkmalen des unabhängigen Patentanspruchs 6 sowie ein Verfahren mit den Merkmalen des unabhängigen Patentanspruchs 9 vor. Ferner schlägt die Erfindung einen Server gemäß des unabhängigen Patentanspruchs 14, ein System gemäß des unabhängigen Patentanspruchs 15 sowie ein Computerprogrammprodukt gemäß des unabhängigen Patentanspruchs 16 vor.

Weitere vorteilhafte Ausgestaltungen der erfindungsgemäßen Steuerungsanordnung sind in den abhängigen Patentansprüchen angegeben.

Der Erfindung liegt die Idee zugrunde, durch ein neues Adressierungsschema, welches insbesondere zusätzlich zum herkömmlichen Adressierungsschema implementiert werden kann, mehrere verschiedene Repräsentationen desselben Datenbestands eines OPC-UA-Adressraums zu ermöglichen. Dies erlaubt verschiedene Informationsmodelle und insbesondere den Zugriff auf Objekte bzw. Knoten in verschiedenen Kontexten. Das neue Adressierungsschema basiert auf alternativen Objektbezeichnern, welche im Wesentlichen die bestehenden herkömmlichen Objektbezeichner mit einem Zusatzbezeichner kombinieren, wobei der alternative Bezeichner als Repräsentant, Alias oder Synonym des herkömmlichen Bezeichners eines Objekts bzw. Knotens zu verstehen ist. Ein wesentlicher Vorteil der Erfindung, welcher später noch näher erläutert wird, ist dabei, dass ein OPC-UA-Server zur Implementierung dieses neuen Adressierungsschemas im Wesentlichen keine zusätzlichen Rechen- oder Speicherressourcen benötigt.

Vorgeschlagen wird demnach ein Verfahren zum Bereitstellen alternativer Objektbezeichner für Objekte eines Adressraums eines Servers innerhalb einer OPC-UA-basierenden Kommunikationsumgebung (K), wobei jedes Objekt des Adressraums des Servers durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist. Dabei wird wenigstens ein Zusatzbezeichner definiert und zweckmäßiger Weise in einem Verzeichnis der für diesen Adressraum des Servers definierten Zusatzbezeichner gespeichert. Zudem wird dabei für jedes Objekt des Adressraums des Servers ein das Objekt ebenfalls eindeutig bezeichnender alternativer Objektbezeichner gebildet, indem der Zusatzbezeichner mit dem herkömmlichen Objektbezeichner des jeweiligen Objekts nach einem bestimmten Schema kombiniert wird.

Sowohl bei diesem wie auch bei den anderen erfindungsgemäßen Verfahren können die Objekte durch Knoten repräsentiert sein, so dass es sich bei den Objektbezeichnern um Knotenbezeichner (NodeIDs) handelt. Ferner kann ein Adressraum in Namensräume unterteilt sein, wobei das jeweilige Verfahren für die Objekte bzw. Knoten eines Namensraums oder mehrerer, insbesondere aller, Namensräume eines Adressraums angewandt bzw. ausgeführt werden kann.

Vorteilhaft ist dabei, dass ein Anwender die alternativen Objektbezeichner als solche klar erkennen und von den herkömmlichen Objektbezeichnern einfach unterscheiden kann.

Die alternativen Objektbezeichner bieten eine große Flexibilität hinsichtlich im Laufe der Zeit gegebenenfalls erforderlicher Änderungen im Adressraum, wie die Vergabe neuer Bezeichner für bestehende Objekte bzw. Knoten, denn solche Änderungen wirken sich Dank des neuen Adressierungsschemas nicht auf die herkömmlichen, sondern nur auf die alternativen Objekt- bzw. Knotenbezeichner aus.

Von Vorteil ist dabei außerdem, dass die alternativen Objektbezeichner aus OPC-UA-Protokollsicht gleich behandelt werden können wie die herkömmlichen Objektbezeichner.

Gemäß einer ersten Ausbildung des Verfahrens gibt das Schema vor, dass wenn der herkömmliche Objektbezeichner vom Bezeichnertyp STRING ist, der Zusatzbezeichner ebenfalls vom Bezeichnertyp STRING ist und zwei Zusatzbezeichnerteile umfasst, wobei der erste Zusatzbezeichnerteil eine frei definierbare Zeichenkette ist und der zweite Zusatzbezeichnerteil ein frei definierbares Separatorzeichen ist.

In einer zweckmäßigen Weiterbildung gibt das Schema vor, dass der alternative Objektbezeichner ebenfalls vom Bezeichnertyp STRING ist und aus den drei Teilen erster Zusatzbezeichnerteil, zweiter Zusatzbezeichnerteil und herkömmlicher Objektbezeichner des jeweiligen Objekts in dieser oder umgekehrter Kombinationsreihenfolge gebildet wird.

Somit ist das neue Adressierungsschema sehr einfach implementierbar für Adressräume mit herkömmlichen Objektbezeichnern vom Bezeichnertyp STRING.

Gemäß einer anderen Ausbildung des Verfahrens gibt das Schema vor, dass wenn der herkömmliche Objektbezeichner vom Bezeichnertyp NUMERIC ist, der Zusatzbezeichner ebenfalls vom Bezeichnertyp NUMERIC und ein Vielfaches eines Basiswertes ist, wobei der Basiswert größer ist als der um 1 erhöhte größte numerische Wert unter allen herkömmlichen Objektbezeichnern der Objekte des Adressraums des Servers.

In einer zweckmäßigen Weiterbildung gibt das Schema vor, dass der alternative Objektbezeichner ebenfalls vom Bezeichnertyp NUMERIC ist und durch Addieren des Zusatzbezeichners und des herkömmlichen Objektbezeichners des jeweiligen Objekts gebildet wird. In diesem Fall

entspricht also der alternative Objektbezeichner der Summe der numerischen Werte des Zusatzbezeichners und des herkömmlichen Objektbezeichners.

Somit ist das neue Adressierungsschema ebenfalls sehr einfach implementierbar für Adressräume mit herkömmlichen Objektbezeichnern vom Bezeichnertyp NUMERIC.

Vorgeschlagen wird außerdem ein Verfahren zum Validieren eines alternativen Objektbezeichners innerhalb einer OPC-UA-basierenden Kommunikationsumgebung, wobei jedes Objekt eines Adressraums eines Servers durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist und zusätzlich durch wenigstens einen alternativen Objektbezeichner eindeutig bezeichnet ist, umfassend die Schritte:

- Entnehmen eines alternativen Objektbezeichners aus einer Anfrage eines Clients,
- Teilen des alternativen Objektbezeichners nach einem bestimmten Schema unter Anwendung eines Teilungskriteriums, so dass anschließend ein herkömmlicher Objektbezeichner und ein Zusatzbezeichner vorliegen,
- Beurteilen des herkömmlichen Objektbezeichners als valide, wenn er in einem Verzeichnis der in diesem Adressraum des Servers vorhandenen herkömmlichen Objektbezeichner ermittelt werden kann,
- Beurteilen des Zusatzbezeichners als valide, wenn er in einem Verzeichnis der für diesen Adressraum des Servers definierten Zusatzbezeichner ermittelt werden kann,
- Beurteilen des alternative Objektbezeichners als valide, wenn zuvor sowohl der herkömmliche Objektbezeichner als auch der Zusatzbezeichner als valide beurteilt wurden,
- Beurteilen des alternativen Objektbezeichners als nicht valide, wenn zuvor der herkömmliche Objektbezeichner und/oder der Zusatzbezeichner als nicht valide beurteilt wurden.

Da die Anfrage eines OPC-UA-Clients grundsätzlich irgendeinen Objektbezeichner enthalten könnte, muss zunächst, insbesondere seitens des OPC-UA-Server, validiert werden, dass der angefragte alternative Objektbezeichner tatsächlich ein gültiger ist, indem validiert wird, dass der darin enthaltene Zusatzbezeichner gültig ist und der darin enthaltene herkömmliche Objektbezeichner tatsächlich Teil des Adressraums ist.

Gemäß einer ersten Ausbildung des Verfahrens gibt das Schema vor, dass wenn der alternative Objektbezeichner vom Bezeichnertyp STRING ist, als Teilungskriterium ein Separatorzeichen verwendet wird und das Separatorzeichen innerhalb der gesamten Zeichenkette des alternativen Objektbezeichners als zweiter Zusatzsbezeichnerteil identifiziert wird, und wobei der Teil der Zeichenkette vor dem Separatorzeichen als erster Zusatzbezeichnerteil und der Teil der Zeichenkette nach dem Separatorzeichen als herkömmlicher Objektbezeichner identifiziert und weiterverarbeitet wird oder der Teil der Zeichenkette vor dem Separatorzeichen als herkömmlicher Objektbezeichner und der Teil der Zeichenkette nach dem Separatorzeichen als erster Zusatzbezeichnerteil identifiziert und weiterverarbeite wird.

Gemäß einer anderen Ausbildung des Verfahrens gibt das Schema vor, dass wenn der alternative Objektbezeichner vom Bezeichnertyp NUMERIC ist, als Teilungskriterium ein Basiswert verwendet wird und der alternative Objektbezeichner durch den Basiswert dividiert wird, wobei der Ganzzahlquotient dieser Division, insbesondere nach erneuter Multiplikation mit dem Basiswert Base als Zusatzbezeichner und der Rest dieser Division als herkömmlicher Objektbezeichner identifiziert und weiterverarbeitet wird.

Somit können angefragte Objektbezeichner vom Bezeichnertyp STRING aber auch vom Bezeichnertyp NUMERIC validiert werden.

Wenn der alternative Objektbezeichner vom Bezeichnertyp STRING ist, kann der als Teilungskriterium zum Teilen des alternativen Objektbezeichners zu verwendende Separator zum Beispiel im Quellcode festgelegt sein, insbesondere im Quellcode für einen sogenannten NodeID-Parser. Alternativ kann das zu verwendende Teilungskriterium zum Beispiel auch als ein konfigurierbarer Parameter eingerichtet sein.

Auch wenn der alternative Objektbezeichner vom Bezeichnertyp NUMERIC ist, kann die als Teilungskriterium zum Teilen des alternativen Objektbezeichners zu verwendende Basis zum Beispiel im Quellcode festgelegt sein, insbesondere im Quellcode für einen sogenannten NodeID-Parser. Dabei sollte der Wert der Basis möglichst groß gewählt sein, so dass eine Überlappung der NodeIDs verschiedener Informationsmodelle eines Namensraums möglichst unwahrscheinlich ist. OPC UA spezifiziert UINT32 als Datentyp für numerische NodeIDs, so dass als maximaler Wert der Basis (2^32-1)/Y gewählt werden kann, wobei Y die maximal mögliche Anzahl verschiedener Dimensionen bzw. Informationsmodelle angibt. Soll hingegen der Wert der Basis minimal klein festgelegt sein, muss die maximale Anzahl der Nodes im Adressraum bekannt sein und beachtet werden. Alternativ kann das zu verwendende Teilungskriterium zum Beispiel auch als ein konfigurierbarer Parameter eingerichtet sein.

Wenn der alternative Objektbezeichner vom Bezeichnertyp STRING ist, kann ein Verzeichnis der definierten Zusatzbezeichner zum Beispiel als Liste oder Array im Quellcode festgelegt sein, insbesondere im Quellcode für einen sogenannten NodeID-Parser.

Auch wenn der alternative Objektbezeichner vom Bezeichnertyp NUMERIC ist, kann ein Verzeichnis der definierten Zusatzbezeichner zum Beispiel als Liste oder Array im Quellcode festgelegt sein, insbesondere im Quellcode für einen sogenannten NodeID-Parser.

Vorgeschlagen wird außerdem ein Verfahren zum Ermitteln von Referenzen eines alternativen Objektbezeichners innerhalb einer OPC-UA-basierenden Kommunikationsumgebung, wobei jedes Objekt eines Adressraums eines Servers durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist und zusätzlich durch wenigstens einen alternativen Objektbezeichner eindeutig bezeichnet ist, umfassend die Schritte:

- Entnehmen eines alternativen Objektbezeichners aus einer Anfrage eines Clients,
- Teilen des alternativen Objektbezeichners nach einem bestimmten Schema unter Anwendung eines Teilungskriteriums, so dass anschließend ein herkömmlicher Objektbezeichner und ein Zusatzbezeichner vorliegen,
- Ermitteln der dem herkömmlichen Objektbezeichner zugeordneten Referenzen in einem Verzeichnis der in diesem Adressraum des Servers vorhandenen Referenzen, wobei jede Referenz einen herkömmlichen Objektbezeichner eines zugeordneten anderes Objekts umfasst,
- für jede ermittelte Referenz Bilden eines alternativen Objektbezeichners durch Kombinieren des Zusatzbezeichners mit dem von der Referenz umfassten herkömmlichen Objektbezeichner des zugeordneten anderen Objekts nach einem bestimmten Schema.

Die Anfrage eines OPC-UA-Clients kann darauf gerichtet sein zu einem angefragten alternativen Objektbezeichner die Referenzen zu ermitteln. Dieses Verfahren erzeugt für jede ermittelte Referenz einen alternativen Objektbezeichner unter Verwendung des in dem angefragten alternativen

Objektbezeichner enthaltenen Zusatzbezeichners. Bei der Beantwortung der Anfrage wird also das vom Client verwendete Informationsmodell berücksichtigt. Dabei ist das Verfahren vorteilhafter Weise sowohl für Vorwärtsreferenzen als auch für Rückwärtsreferenzen anwendbar bzw. ausführbar.

Gemäß einer ersten Ausbildung des Verfahrens gibt das Schema vor, dass wenn der alternative Objektbezeichner, dessen Referenzen zu ermitteln sind, vom Bezeichnertyp STRING ist, als Teilungskriterium ein Separatorzeichen verwendet wird und das Separatorzeichen innerhalb der Zeichenkette des alternativen Objektbezeichners als zweiter Zusatzsbezeichnerteil identifiziert wird,
und wobei der Teil der Zeichenkette vor dem Separatorzeichen als erster Zusatzbezeichnerteil und der Teil der Zeichenkette nach dem Separatorzeichen als herkömmlicher Objektbezeichner identifiziert und weiterverarbeitet wird
oder der Teil der Zeichenkette vor dem Separatorzeichen als herkömmlicher Objektbezeichner und der Teil der Zeichenkette nach dem Separatorzeichen als erster Zusatzbezeichnerteil identifiziert und weiterverarbeite wird.

Gemäß einer Weiterbildung des Verfahrens gibt das Schema vor, dass für jede ermittelte Referenz der alternative Objektbezeichner ebenfalls vom Bezeichnertyp STRING ist und aus drei Teilen gebildet wird
und zwar in der Reihenfolge erster Zusatzbezeichnerteil, zweiter Zusatzbezeichnerteil und herkömmlicher Objektbezeichner des jeweiligen zugeordneten anderen Objekts
oder in der Reihenfolge herkömmlicher Objektbezeichner des jeweiligen zugeordneten anderen Objekts, zweiter Zusatzbezeichnerteil und erster Zusatzbezeichnerteil.

Gemäß einer anderen Ausbildung des Verfahrens gibt das Schema vor, dass wenn der alternative Objektbezeichner, dessen Referenzen zu ermitteln sind, vom Bezeichnertyp NUMERIC ist, als Teilungskriterium ein Basiswert verwendet wird und der alternative Objektbezeichner durch den Basiswert dividiert wird, wobei der Ganzzahlquotient dieser Division, insbesondere nach erneuter Multiplikation mit dem Basiswert Base, als Zusatzbezeichner und der Rest dieser Division als herkömmlicher Objektbezeichner weiterverarbeitet wird.

Gemäß einer Weiterbildung des Verfahrens gibt das Schema vor, dass für jede ermittelte Referenz der alternative Objektbezeichner ebenfalls vom Bezeichnertyp NUMERIC ist und durch Addieren des Zusatzbezeichners und des herkömmlichen Objektbezeichners des jeweiligen zugeordneten anderen Objekts gebildet wird.

Somit können für angefragte Objektbezeichner vom Bezeichnertyp STRING aber auch vom Bezeichnertyp NUMERIC effektiv Referenzen ermittelt werden.

Vorgeschlagen wird ferner ein Server, insbesondere OPC-UA-Server, welcher zur Durchführung wenigsten eines der zuvor beschriebenen erfindungsgemäßen Verfahren ausgebildet ist.

Vorgeschlagen wird zudem ein System umfassend eine OPC-UA-basierende Kommunikationsumgebung mit wenigstens einem erfindungsgemäßen Server wie zuvor beschrieben und mit wenigstens einem Client, welche über ein Netzwerk miteinander verbunden sind.

Vorgeschlagen wird schließlich auch ein Computerprogrammprodukt, umfassend Anweisungen, die bei Ausführung durch einen Computer, insbesondere einen Server, diesen veranlassen, wenigstens eines der zuvor beschriebenen erfindungsgemäßen Verfahren auszuführen.

Die zu den erfindungsgemäßen Verfahren einschließlich deren Ausführungen und Weiterbildungen angegebenen Merkmale und Vorteile gelten sinngemäß auch für den erfindungsgemäßen Server sowie für das System und das Computerprogrammprodukt.

Diese und weitere Merkmale und Vorteile der vorliegenden Erfindung ergeben sich auch aus den Ausführungsbeispielen, welche nachfolgend unter Bezugnahme auf die beiliegenden Zeichnungen näher erläutert werden. In schematischer Darstellung zeigt dabei

Figur 1 ein System mit einer OPC-UA-basierenden Kommunikationsumgebung;

Figur 2 ein erstes Beispiel eines Verfahrens zum Bereitstellen eines alternativen Objektbezeichners;

Figur 3 ein erstes Beispiel eines Verfahrens zum Validieren eines alternativen Objektbezeichners;

Figur 4 ein erstes Beispiel eines Verfahrens zum Ermitteln von Referenzen eines alternativen Objektbezeichners;

Figur 5 ein zweites Beispiel eines Verfahrens zum Bereitstellen eines alternativen Objektbezeichners;

Figur 6 ein zweites Beispiel eines Verfahrens zum Validieren eines alternativen Objektbezeichners;

Figur 7 ein zweites Beispiel eines Verfahrens zum Ermitteln von Referenzen eines alternativen Objektbezeichners;

Figur 8 einen Adressraum eines OPC-UA-Servers, worin die Komponentenvariablen einer Steuerung als Knoten

mit eindeutigen Knotenbezeichner des Bezeichnertyp STRING aufgeführt sind;

Figur 9 einen Adressraum eines OPC-UA-Servers, worin das Verzeichnis aller Namensräume bzw. Namensraumindizies des Adressraums als Attribut „NamespaceArray" des Knotens „Server" hinterlegt ist;

Figur 10 ein erstes Beispiel einer Auflistung von herkömmlichen Objektbezeichnern und alternativen Objektbezeichnern für Objekte bzw. Knoten eines Adressraums, welche Variablen der Steuerungskomponente „Arp.PLC.Eclr" in unterschiedlichen Informationsmodellen repräsentieren;

Figur 11 ein zweites Beispiel einer Auflistung von herkömmlichen Objektbezeichnern und alternativen Objektbezeichnern für Objekte bzw. Knoten eines Adressraums, welche Variablen der Steuerungskomponente „Arp.PLC.Eclr" in unterschiedlichen Informationsmodellen repräsentieren.

Figur 1 zeigt beispielhaft ein System mit einer OPC-UA-basierenden Kommunikationsumgebung K, welches hier zwei Server S1 und S2 sowie zwei Clients C1 und C2 aufweist, welche über ein Netzwerk N miteinander verbunden sind. Dabei sind die Clients als OPC-UA-Clients und die Server als OPC-UA-Server ausgebildet. Die Server sind zudem zur Durchführung der erfindungsgemäßen Verfahren zur Bereitstellung, Validierung und Ermittlung von Referenzen alternativer Objektbezeichner ausgebildet, wobei die einzelnen Verfahren im Folgenden noch näher erläutert

werden. Die Server S1 und S2 stellen Ihre Informationen den Clients C1 und C2 in einem Adressraum bereit.

Die Figur 8 zeigt beispielhaft einen Teil des Adressraums des OPC-UA-Servers S1. Darin sind die Komponentenvariablen einer nicht dargestellten Steuerung der PLCnext Technology, mit der zumindest der Server S1 in Verbindung steht, als Knoten mit eindeutigen herkömmlichen Knotenbezeichnern des Bezeichnertyps STRING aufgeführt. Figur 9 zeigt einen anderen Teil des Adressraums des OPC-UA-Servers S1, wobei hier erkennbar ist, dass das Verzeichnis aller Namensräume bzw. Namensraumindizies des Adressraums als Attribut „NamespaceArray" des Knotens „Server" hinterlegt ist.

Der OPC-UA-Standard spezifiziert nicht, wie ein Server die Knoten zu verwalten hat. Es sei angenommen, dass der Adressraum in Namensräume N[x] unterteilt ist, wobei die herkömmlichen Knotenbezeichner NodeID in einem Namensraum N[x] vom Bezeichnertyp identifierType = STRING oder identifierType = NUMERIC sind. Es sei ferner angenommen, dass der Server eine als NodeIDArray bezeichnete Array-artige Struktur nutzt, um valide Knotenbezeichner NodeID zu speichern, und eine als Forward-/BackwardReferenceMap bezeichnete Kartenartige Struktur nutzt, um Referenzen zu speichern, d. h. Arrays von Knotenbezeichnern NodeIDs der Referenzen zu jedem Knotenbezeichner NodeID. Diese Annahmen wurden jedoch nur zur Vereinfachung der Beschreibung der Erfindung getroffen, jedoch nicht zur Einschränkung der Erfindung.

Die Erfindung zielt darauf ab, einen bestehenden Adressraum, insbesondere innerhalb eines Namensraums N[x], um eine Dimension y mit y = 1 … Y zu erweitern, und zwar bei geringen Kosten bzw. bei geringem Aufwand, d.h. ohne Replizierung von Knoteninstanzen in einem Speicher, insbesondere ohne neue Elemente zum NodeIDArray oder zur Forward-/BackwardReferenceMap hinzuzufügen. Somit

ermöglicht die Erfindung das Implementieren Y zusätzlicher Informatinsmodelle bzw. Kontexte für einen Datenbestand.

Hierzu schlägt die Erfindung ein neues, eingangs bereits erwähntes Adressierungsschema vor.

Im Fall eines herkömmlichen Knotenbezeichners NodeID vom Bezeichnertyp identifierType = STRING in einem Namensraum N[x] wird dabei ein neuer bzw. alternativer Knotenbezeichner PrefixedNodeID desselben Bezeichnertyps definiert, welcher wie in Figur 2 gezeigt gebildet wird durch die Kombination bzw. Verkettung (vgl. Prefix Concatenator) eines ersten Teils eines Zusatzbezeichners Prefix[y] und eines zweiten Teils eines Zusatzbezeichners in Form eines Separators Literal mit dem herkömmlichen Knotenbezeichner NodeID, wobei Prefix[y] ein Element der Menge Prefix[Y] mit dem Index y = 1:Y ist.

Im Fall eines herkömmlichen Knotenbezeichners NodeID vom Bezeichnertyp identifierType = NUMERIC in einem Namensraum N[x] wird ein neuer bzw. alternativer Knotenbezeichner BiasedNodeID desselben Bezeichnertyps definiert, welcher wie in Figur 5 gezeigt gebildet wird durch die Kombination bzw. Addition (vgl. Bias Adder) eines Zusatzbezeichners Bias[y] und des herkömmlichen Knotenbezeichners NodeID, wobei Bias[y] ein Element der Menge Bias[Y] mit dem Index y = 1:Y ist, und wobei Bias[y] das Produkt der Multiplikation eines Basiswerts Base mit y ist (vgl. Base Multiplier), wobei der Basiswert Base größer ist als der um 1 erhöhte größte numerische Wert unter allen herkömmlichen Knotenbezeichnern NodeID in dem Namensraum N[x].

Wenn beispielsweise der Namensraum N[x] 10 Knoten mit den Knotenbezeichnern NodeID = 1 … 9 umfasst, so kann der Basiswert Base = 100 sein, da 100 > max (0 … 9) gilt, so dass sich für Y = 3 alternative Knotenbezeichner Biased NodeID = (100 … 109, 200 … 209, 300 … 309) ergeben.

Allgemein ausgedrückt ist somit ein Verfahren zum Bereitstellen alternativer Objektbezeichner für Objekte eines Adressraums, insbesondere eines Namensraums, eines Servers S1, S2 innerhalb einer OPC-UA-basierenden Kommunikationsumgebung K vorgesehen, wobei jedes Objekt des Adressraums des Servers S1, S2 durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet, wobei wenigstens ein Zusatzbezeichner definiert und in einem Verzeichnis der für diesen Adressraum des Servers S1, S2 definierten Zusatzbezeichner gespeichert wird, und wobei für jedes Objekt des Adressraums des Servers S1, S2 ein das Objekt ebenfalls eindeutig bezeichnender alternativer Objektbezeichner gebildet wird, indem der Zusatzbezeichner mit dem herkömmlichen Objektbezeichner des jeweiligen Objekts nach einem bestimmten Schema kombiniert wird.

In einer ersten Ausbildung des Verfahrens gibt das Schema vor, dass wenn der herkömmliche Objektbezeichner vom Bezeichnertyp STRING ist, der Zusatzbezeichner ebenfalls vom Bezeichnertyp STRING ist und zwei Zusatzbezeichnerteile umfasst, wobei der erste Zusatzbezeichnerteil eine frei definierbare Zeichenkette ist und der zweite Zusatzbezeichnerteil ein frei definierbares Separatorzeichen ist. Außerdem gibt das Schema vor, dass der alternative Objektbezeichner ebenfalls vom Bezeichnertyp STRING ist und aus den drei Teilen erster Zusatzbezeichnerteil, zweiter Zusatzbezeichnerteil und herkömmlicher Objektbezeichner des jeweiligen Objekts in dieser Kombinationsreihenfolge gebildet wird (vgl. Figur 2).

Beispielhafte Ergebnisse dieses Verfahrens sind in den Figuren 10 und 11 dargestellt.

Zu sehen ist in Figur 10 eine Auflistung von herkömmlichen Objektbezeichnern und alternativen Objektbezeichnern für Objekte bzw. Knoten eines Adressraums, welche Variablen der

Steuerungskomponente „Arp.PLC.Eclr" in unterschiedlichen Informationsmodellen repräsentieren. Zum einen wurde der Zusatzbezeichner „PlcOpen.Programs" definiert, um ein Informationsmodell für Programmvariablen bereitzustellen. Zum anderen wurde der Zusatzbezeichner „PlcOpen.GlobalVars" definiert, um ein Informationsmodell für globale Variablen bereitzustellen. In beiden Fällen umfasst der Zusatzbezeichner zudem das Separatorzeichen „:".

In Figur 11 ist eine weitere Auflistung von herkömmlichen Objektbezeichnern und alternativen Objektbezeichnern für Objekte bzw. Knoten eines Adressraums zu sehen, welche Variablen der Steuerungskomponente „Arp.PLC.Eclr" in nochmals anderen Informationsmodellen repräsentieren. Zum einen wurde hier der Zusatzbezeichner „History.SessionA" definiert, für ein Informationsmodell zum Zugriff auf historische Werte dieser Variablen, welche in einer Session A mit einer ersten Samplingrate erfasst wurden. Zum anderen wurde der Zusatzbezeichner „History.SessionB" definiert, für ein Informationsmodell zum Zugriff auf historische Werte dieser Variablen, welche in einer Session B mit einer zweiten Samplingrate erfasst wurden. In beiden Fällen umfasst der Zusatzbezeichner zudem wieder das Separatorzeichen „:".

In den Beispielen gemäß der Figuren 2, 10 und 11 ist der Zusatzbezeichner als Prefix zum herkömmlichen Objektbezeichner vorgesehen. Denkbar wäre aber auch, den Zusatzbezeichner als Suffix zum herkömmlichen Objektbezeichner vorzusehen.

In einer alternativen Ausbildung des Verfahrens gibt das Schema vor, dass wenn der herkömmliche Objektbezeichner vom Bezeichnertyp NUMERIC ist, der Zusatzbezeichner ebenfalls vom Bezeichnertyp NUMERIC und ein Vielfaches eines Basiswertes ist, wobei der Basiswert größer ist als der um 1 erhöhte größte numerische Wert unter allen herkömmlichen

Objektbezeichnern der Objekte des Adressraums des Servers S1, S2. Außerdem gibt das Schema vor, dass der alternative Objektbezeichner ebenfalls vom Bezeichnertyp NUMERIC ist und durch Addieren des Zusatzbezeichners und des herkömmlichen Objektbezeichners des jeweiligen Objekts gebildet wird (vgl. Figur 5).

Die Figuren 3 und 6 zeigen zwei Beispiele eines Verfahrens zum Validieren eines alternativen Objektbezeichners innerhalb einer OPC-UA-basierenden Kommunikationsumgebung K, wobei jedes Objekt eines Adressraums eines Servers S1, S2 durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist und zusätzlich durch wenigstens einen alternativen Objektbezeichner eindeutig bezeichnet ist. Im Allgemeinen umfasst das Verfahren die Schritte:

- Entnehmen eines alternativen Objektbezeichners PrefixedNodeID bzw. BiasedNodeID aus einer Anfrage eines Clients C1, C2,
- Teilen des alternativen Objektbezeichners nach einem bestimmten Schema unter Anwendung eines Teilungskriteriums, so dass anschließend ein herkömmlicher Objektbezeichner NodeID und ein Zusatzbezeichner Prefix bzw. Bias vorliegen,
- Beurteilen des herkömmlichen Objektbezeichners NodeID als valide, wenn er in einem Verzeichnis NodeIDArray der in diesem Adressraum des Servers S1, S2 vorhandenen herkömmlichen Objektbezeichner ermittelt werden kann (vgl. Node Validator),
- Beurteilen des Zusatzbezeichners als valide, wenn er in einem Verzeichnis Prefix[Y] bzw. Bias[Y] der für diesen Adressraum des Servers S1, S2 definierten Zusatzbezeichner ermittelt werden kann (vgl. Prefix Validator bzw. Bias Validator),
- Beurteilen des alternativen Objektbezeichners als valide, wenn zuvor sowohl der herkömmliche Objektbezeichner als auch der Zusatzbezeichner als valide beurteilt wurden (vgl. Logical AND),

- Beurteilen des alternativen Objektbezeichners als nicht valide, wenn zuvor der herkömmliche Objektbezeichner und/oder der Zusatzbezeichner als nicht valide beurteilt wurden.

Gemäß einer ersten Ausbildung des Verfahrens (vgl. Figur 3) gibt das Schema vor, dass wenn der alternative Objektbezeichner vom Bezeichnertyp STRING ist, als Teilungskriterium ein Separatorzeichen Literal verwendet wird und das Separatorzeichen innerhalb der gesamten Zeichenkette des alternativen Objektbezeichners PrefixedNodeID als zweiter Zusatzsbezeichnerteil identifiziert wird, wobei der Teil der Zeichenkette vor dem Separatorzeichen als erster Zusatzbezeichnerteil Prefix und der Teil der Zeichenkette nach dem Separatorzeichen als herkömmlicher Objektbezeichner NodeID identifiziert und weiterverarbeitet wird (vgl. Prefix Splitter).

In einem nicht in den Figuren gezeigten Fall könnte stattdessen vorgesehen sein, dass der Teil der Zeichenkette vor dem Separatorzeichen als herkömmlicher Objektbezeichner und der Teil der Zeichenkette nach dem Separatorzeichen als erster Zusatzbezeichnerteil identifiziert und weiterverarbeitet wird.

Gemäß einer anderen Ausbildung des Verfahrens (vgl. Figur 6) gibt das Schema vor, dass wenn der alternative Objektbezeichner vom Bezeichnertyp NUMERIC ist, als Teilungskriterium ein Basiswert Base verwendet wird und der alternative Objektbezeichner BiasedNodeID durch den Basiswert Base dividiert wird, wobei der Ganzzahlquotient dieser Division, insbesondere nach erneuter Multiplikation mit dem Basiswert Base, als Zusatzbezeichner Bias und der Rest dieser Division als herkömmlicher Objektbezeichner NodeID identifiziert und weiterverarbeitet wird (vgl. Base Modulo). In der Figur 6 ist die erneute Multiplikation des Ganzzahlquotienten mit dem Basiswert Base nicht gezeigt.

Der Ganzzahlquotient an sich, entspricht dem Index y der Dimension bzw. des Informationsmodells.

Aufgrund dieser erfindungsgemäßen stufenweisen Validierung, kann die Validierung alternativer Objektbezeichner sehr effizient erfolgen. Und da bei der erfindungsgemäßen Bereitstellung alternativer Objektbezeichner für ein zusätzliches Informationsmodell lediglich ein neuer Zusatzbezeichner in das Verzeichnis der Zusatzbezeichner eingetragen wird, jedoch keine Knoteninstanzen repliziert werden, erfordert die Validierung alternativer Objektbezeichner kaum zusätzlichen Bedarf an Speicherplatz und Rechenleistung bzw. Rechenzeit. Somit ermöglicht die Erfindung das Implementieren Y zusätzlicher Informationsmodelle bzw. Kontexte für einen Datenbestand, ohne dass dadurch der Bedarf an Speicherplatz und Rechenleistung bzw. Rechenzeit um den Faktor Y+1 steigen würde.

Die Figuren 4 und 7 zeigen zwei Beispiele eines Verfahrens zum Ermitteln von Referenzen eines alternativen Objektbezeichners innerhalb einer OPC-UA-basierenden Kommunikationsumgebung K, wobei jedes Objekt eines Adressraums eines Servers S1, S2 durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist und zusätzlich durch wenigstens einen alternativen Objektbezeichner eindeutig bezeichnet ist. Im Allgemeinen umfasst das Verfahren die Schritte:

- Entnehmen eines alternativen Objektbezeichners PrefixedNodeID bzw. BiasedNodeID aus einer Anfrage eines Clients C1, C2,
- Teilen des alternativen Objektbezeichners nach einem bestimmten Schema unter Anwendung eines Teilungskriteriums, so dass anschließend ein herkömmlicher Objektbezeichner NodeID und ein Zusatzbezeichner Prefix bzw. Bias vorliegen,
- Ermitteln der dem herkömmlichen Objektbezeichner NodeID

zugeordneten Referenzen in einem Verzeichnis ForwardReferenceMap der in diesem Adressraum des Servers S1, S2 vorhandenen Referenzen, wobei jede Referenz einen herkömmlichen Objektbezeichner eines zugeordneten anderes Objekts umfasst (vgl. NodeID Lookup),

- für jede ermittelte Referenz ForwardReferences Bilden eines alternativen Objektbezeichners PrefixedForwardReferences bzw. BiasedForwardReferences durch Kombinieren des Zusatzbezeichners Prefix bzw. Bias mit dem von der Referenz ForwardReferences umfassten herkömmlichen Objektbezeichner des zugeordneten anderen Objekts nach einem bestimmten Schema (vgl. Prefix Concatenator bzw. Bias Adder).

Gemäß einer ersten Ausbildung des Verfahrens (vgl. Figur 4) gibt das Schema vor, dass wenn der alternative Objektbezeichner PrefixedNodeID, dessen Referenzen zu ermitteln sind, vom Bezeichnertyp STRING ist, als Teilungskriterium ein Separatorzeichen Literal verwendet wird und das Separatorzeichen innerhalb der Zeichenkette des alternativen Objektbezeichners als zweiter Zusatzsbezeichnerteil identifiziert wird,
und wobei der Teil der Zeichenkette vor dem Separatorzeichen als erster Zusatzbezeichnerteil Prefix und der Teil der Zeichenkette nach dem Separatorzeichen als herkömmlicher Objektbezeichner NodeID identifiziert und weiterverarbeitet wird.

In einem nicht in den Figuren gezeigten Fall könnte stattdessen vorgesehen sein, dass der Teil der Zeichenkette vor dem Separatorzeichen als herkömmlicher Objektbezeichner und der Teil der Zeichenkette nach dem Separatorzeichen als erster Zusatzbezeichnerteil identifiziert und weiterverarbeitet wird.

Weiterhin gibt das Schema vor, dass für jede ermittelte Referenz ForwardReferences der alternative Objektbezeichner

PrefixedForwardReferences ebenfalls vom Bezeichnertyp STRING ist und aus drei Teilen gebildet wird und zwar in der Reihenfolge erster Zusatzbezeichnerteil Prefix, zweiter Zusatzbezeichnerteil Separator Literal und herkömmlicher Objektbezeichner des jeweiligen zugeordneten anderen Objekts oder in der Reihenfolge herkömmlicher Objektbezeichner des jeweiligen zugeordneten anderen Objekts, zweiter Zusatzbezeichnerteil und erster Zusatzbezeichnerteil.

In dem Beispiel gemäß der Figur 4 tritt der Zusatzbezeichner als Prefix zum herkömmlichen Objektbezeichner auf. Denkbar wäre aber auch, dass der Zusatzbezeichner als Suffix zum herkömmlichen Objektbezeichner erscheint.

Gemäß einer anderen Ausbildung des Verfahrens (vgl. Figur 7) gibt das Schema vor, dass wenn der alternative Objektbezeichner BiasedNodeID, dessen Referenzen zu ermitteln sind, vom Bezeichnertyp NUMERIC ist, als Teilungskriterium ein Basiswert Base verwendet wird und der alternative Objektbezeichner durch den Basiswert dividiert wird, wobei der Ganzzahlquotient dieser Division nach erneuter Multiplikation mit dem Basiswert Base als Zusatzbezeichner Bias und der Rest dieser Division als herkömmlicher Objektbezeichner NodeID weiterverarbeitet wird. In der Figur 7 ist die erneute Multiplikation des Ganzzahlquotienten mit dem Basiswert Base nicht gezeigt. Der Ganzzahlquotient an sich, entspricht dem Index y der Dimension bzw. des Informationsmodells.

Weiterhin gibt das Schema vor, dass für jede ermittelte Referenz ForwardReferences der alternative Objektbezeichner BiasedForwardReferences ebenfalls vom Bezeichnertyp NUMERIC ist und durch Addieren des Zusatzbezeichners Bias und des

herkömmlichen Objektbezeichners des jeweiligen zugeordneten anderen Objekts gebildet wird. LU101410

Auch wenn das Verfahren gemäß der Figuren 4 und 7 für das Ermitteln von Vorwärtsreferenzen dargestellt und beschrieben ist, kann dieses ebenfalls für das Ermitteln von Rückwärtsreferenzen genutzt werden.

Bezugszeichenliste

K	OPC-UA-basierende Kommunikationsumgebung
S1, S2	Server, OPC-UA-Server
C1, C2	Client, OPC-UA-Client
N	Netzwerk

Patentansprüche

1. Verfahren zum Bereitstellen alternativer Objektbezeichner für Objekte eines Adressraums eines Servers (S1, S2) innerhalb einer OPC-UA-basierenden Kommunikationsumgebung (K),
wobei jedes Objekt des Adressraums des Servers (S1, S2) durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist,
wobei wenigstens ein Zusatzbezeichner definiert und in einem Verzeichnis der für diesen Adressraum des Servers (S1, S2) definierten Zusatzbezeichner gespeichert wird,
und wobei für jedes Objekt des Adressraums des Servers (S1, S2) ein das Objekt ebenfalls eindeutig bezeichnender alternativer Objektbezeichner gebildet wird, indem der Zusatzbezeichner mit dem herkömmlichen Objektbezeichner des jeweiligen Objekts nach einem bestimmten Schema kombiniert wird.

2. Verfahren nach Anspruch 1,
wobei das Schema vorgibt, dass wenn der herkömmliche Objektbezeichner vom Bezeichnertyp STRING ist, der Zusatzbezeichner ebenfalls vom Bezeichnertyp STRING ist und zwei Zusatzbezeichnerteile umfasst,
wobei der erste Zusatzbezeichnerteil eine frei definierbare Zeichenkette ist und der zweite Zusatzbezeichnerteil ein frei definierbares Separatorzeichen ist.

3. Verfahren nach Anspruch 2,
wobei das Schema vorgibt, dass der alternative Objektbezeichner ebenfalls vom Bezeichnertyp STRING ist und aus den drei Teilen erster Zusatzbezeichnerteil, zweiter Zusatzbezeichnerteil und herkömmlicher Objektbezeichner des jeweiligen Objekts

in dieser oder umgekehrter Kombinationsreihenfolge gebildet wird.

4. Verfahren nach Anspruch 1,
wobei das Schema vorgibt, dass wenn der herkömmliche Objektbezeichner vom Bezeichnertyp NUMERIC ist, der Zusatzbezeichner ebenfalls vom Bezeichnertyp NUMERIC und ein Vielfaches eines Basiswertes ist,
wobei der Basiswert größer ist als der um 1 erhöhte größte numerische Wert unter allen herkömmlichen Objektbezeichnern der Objekte des Adressraums des Servers (S1, S2).

5. Verfahren nach Anspruch 4,
wobei das Schema vorgibt, dass der alternative Objektbezeichner ebenfalls vom Bezeichnertyp NUMERIC ist und durch Addieren des Zusatzbezeichners und des herkömmlichen Objektbezeichners des jeweiligen Objekts gebildet wird.

6. Verfahren zum Validieren eines alternativen Objektbezeichners innerhalb einer OPC-UA-basierenden Kommunikationsumgebung (K),
wobei jedes Objekt eines Adressraums eines Servers (S1, S2) durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist und zusätzlich durch wenigstens einen alternativen Objektbezeichner eindeutig bezeichnet ist,
umfassend die Schritte:
- Entnehmen eines alternativen Objektbezeichners aus einer Anfrage eines Clients (C1, C2),
- Teilen des alternativen Objektbezeichners nach einem bestimmten Schema unter Anwendung eines Teilungskriteriums, so dass anschließend ein herkömmlicher Objektbezeichner und ein Zusatzbezeichner vorliegen,
- Beurteilen des herkömmlichen Objektbezeichner**s** als

valide, wenn er in einem Verzeichnis der in diesem Adressraum des Servers (S1, S2) vorhandenen herkömmlichen Objektbezeichner ermittelt werden kann,

- Beurteilen des Zusatzbezeichners als valide, wenn er in einem Verzeichnis der für diesen Adressraum des Servers (S1, S2) definierten Zusatzbezeichner ermittelt werden kann,

- Beurteilen des alternativen Objektbezeichners als valide, wenn zuvor sowohl der herkömmliche Objektbezeichner als auch der Zusatzbezeichner als valide beurteilt wurden,

- Beurteilen des alternativen Objektbezeichners als nicht valide, wenn zuvor der herkömmliche Objektbezeichner und/oder der Zusatzbezeichner als nicht valide beurteilt wurden.

7. Verfahren nach Anspruch 6,
wobei das Schema vorgibt, dass wenn der alternative Objektbezeichner vom Bezeichnertyp STRING ist, als Teilungskriterium ein Separatorzeichen verwendet wird und das Separatorzeichen innerhalb der gesamten Zeichenkette des alternativen Objektbezeichners als zweiter Zusatzsbezeichnerteil identifiziert wird, und wobei der Teil der Zeichenkette vor dem Separatorzeichen als erster Zusatzbezeichnerteil und der Teil der Zeichenkette nach dem Separatorzeichen als herkömmlicher Objektbezeichner identifiziert und weiterverarbeitet wird oder der Teil der Zeichenkette vor dem Separatorzeichen als herkömmlicher Objektbezeichner und der Teil der Zeichenkette nach dem Separatorzeichen als erster Zusatzbezeichnerteil identifiziert und weiterverarbeite wird.

8. Verfahren nach Anspruch 6,
wobei das Schema vorgibt, dass wenn der alternative Objektbezeichner vom Bezeichnertyp NUMERIC ist, als

Teilungskriterium ein Basiswert verwendet wird und der alternative Objektbezeichner durch den Basiswert dividiert wird, wobei der Ganzzahlquotient dieser Division, insbesondere nach erneuter Multiplikation mit dem Basiswert Base, als Zusatzbezeichner und der Rest dieser Division als herkömmlicher Objektbezeichner identifiziert und weiterverarbeitet wird. LU101410

9. Verfahren zum Ermitteln von Referenzen eines alternativen Objektbezeichners innerhalb einer OPC-UA-basierenden Kommunikationsumgebung (K),
wobei jedes Objekt eines Adressraums eines Servers (S1, S2) durch einen herkömmlichen Objektbezeichner eindeutig bezeichnet ist und zusätzlich durch wenigstens einen alternativen Objektbezeichner eindeutig bezeichnet ist,
umfassend die Schritte:
- Entnehmen eines alternativen Objektbezeichners aus einer Anfrage eines Clients (C1, C2),
- Teilen des alternativen Objektbezeichners nach einem bestimmten Schema unter Anwendung eines Teilungskriteriums, so dass anschließend ein herkömmlicher Objektbezeichner und ein Zusatzbezeichner vorliegen,
- Ermitteln der dem herkömmlichen Objektbezeichner zugeordneten Referenzen in einem Verzeichnis der in diesem Adressraum des Servers (S1, S2) vorhandenen Referenzen, wobei jede Referenz einen herkömmlichen Objektbezeichner eines zugeordneten anderes Objekts umfasst,
- für jede ermittelte Referenz Bilden eines alternativen Objektbezeichners durch Kombinieren des Zusatzbezeichners mit dem von der Referenz umfassten herkömmlichen Objektbezeichner des zugeordneten anderen Objekts nach einem bestimmten Schema.

10. Verfahren nach Anspruch 9,
wobei das Schema vorgibt, dass wenn der alternative Objektbezeichner, dessen Referenzen zu ermitteln sind, vom Bezeichnertyp STRING ist, als Teilungskriterium ein Separatorzeichen verwendet wird und das Separatorzeichen innerhalb der Zeichenkette des alternativen Objektbezeichners als zweiter Zusatzsbezeichnerteil identifiziert wird,
und wobei der Teil der Zeichenkette vor dem Separatorzeichen als erster Zusatzbezeichnerteil und der Teil der Zeichenkette nach dem Separatorzeichen als herkömmlicher Objektbezeichner identifiziert und weiterverarbeitet wird
oder der Teil der Zeichenkette vor dem Separatorzeichen als herkömmlicher Objektbezeichner und der Teil der Zeichenkette nach dem Separatorzeichen als erster Zusatzbezeichnerteil identifiziert und weiterverarbeitet wird.

11. Verfahren nach Anspruch 10,
wobei das Schema vorgibt, dass für jede ermittelte Referenz der alternative Objektbezeichner ebenfalls vom Bezeichnertyp STRING ist und aus drei Teilen gebildet wird
und zwar in der Reihenfolge erster Zusatzbezeichnerteil, zweiter Zusatzbezeichnerteil und herkömmlicher Objektbezeichner des jeweiligen zugeordneten anderen Objekts
oder in der Reihenfolge herkömmlicher Objektbezeichner des jeweiligen zugeordneten anderen Objekts, zweiter Zusatzbezeichnerteil und erster Zusatzbezeichnerteil.

12. Verfahren nach Anspruch 9,
wobei das Schema vorgibt, dass wenn der alternative Objektbezeichner, dessen Referenzen zu ermitteln sind, vom Bezeichnertyp NUMERIC ist, als Teilungskriterium

ein Basiswert verwendet wird und der alternative Objektbezeichner durch den Basiswert dividiert wird, wobei der Ganzzahlquotient dieser Division, insbesondere nach erneuter Multiplikation mit dem Basiswert Base, als Zusatzbezeichner und der Rest dieser Division als herkömmlicher Objektbezeichner weiterverarbeitet wird.

13 Verfahren nach Anspruch 12,
wobei das Schema vorgibt, dass für jede ermittelte Referenz der alternative Objektbezeichner ebenfalls vom Bezeichnertyp NUMERIC ist und durch Addieren des Zusatzbezeichners und des herkömmlichen Objektbezeichners des jeweiligen zugeordneten anderen Objekts gebildet wird.

14. Server (S1, S1) ausgebildet zur Durchführung des Verfahrens nach einem der Ansprüche 1 bis 13.

15. System umfassend eine OPC-UA-basierende Kommunikationsumgebung (K) mit wenigstens einem Server (S1, S1) nach Anspruch 14 und wenigstens einem Client (C1, C2), die über ein Netzwerk (N) miteinander verbunden sind.

16. Computerprogrammprodukt, umfassend Anweisungen, die bei Ausführung durch einen Computer, diesen veranlassen, das Verfahren nach einem der Ansprüche 1 bis 13 auszuführen.

Fig. 1

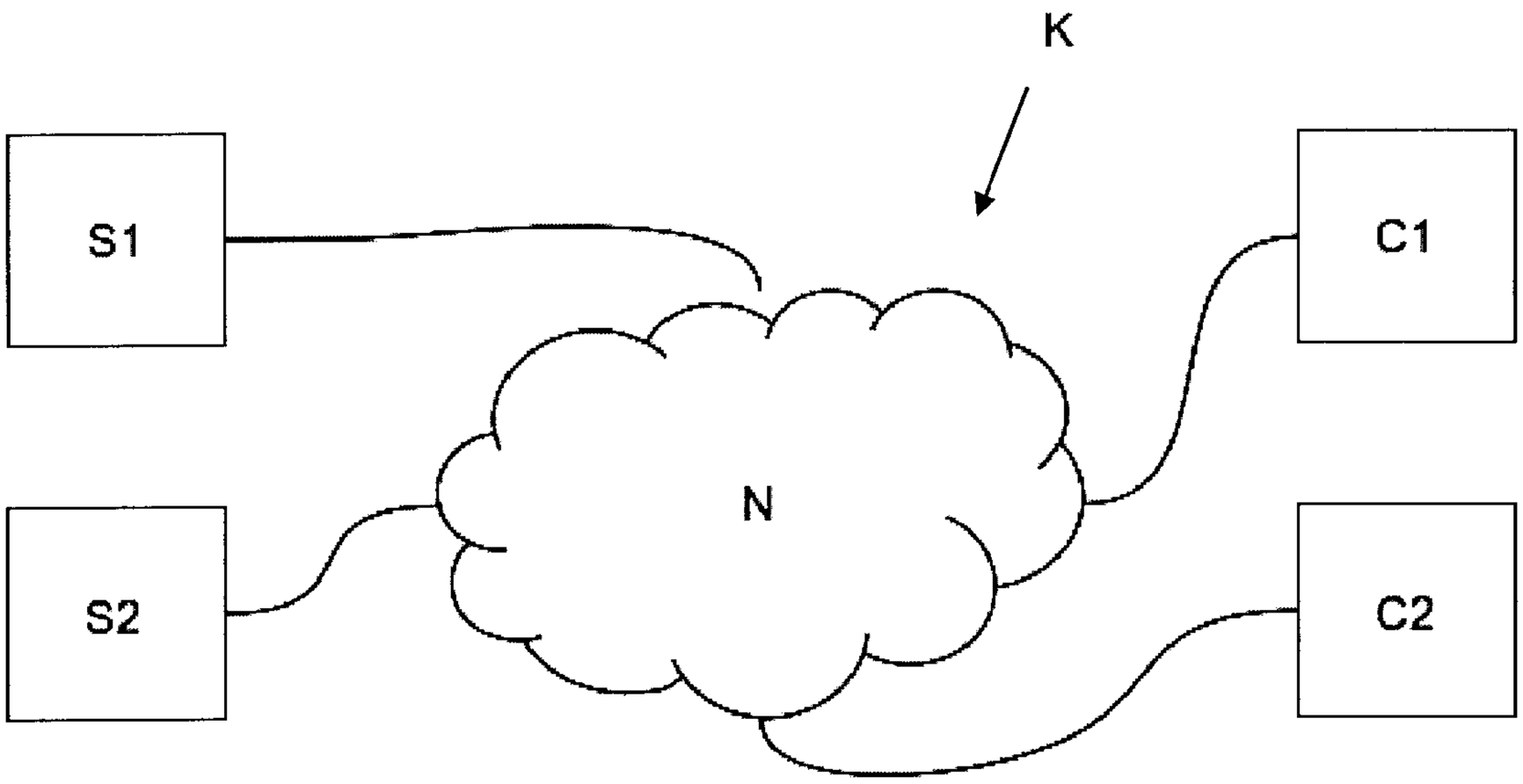

K
S1
S2
N
C1
C2

Fig. 2

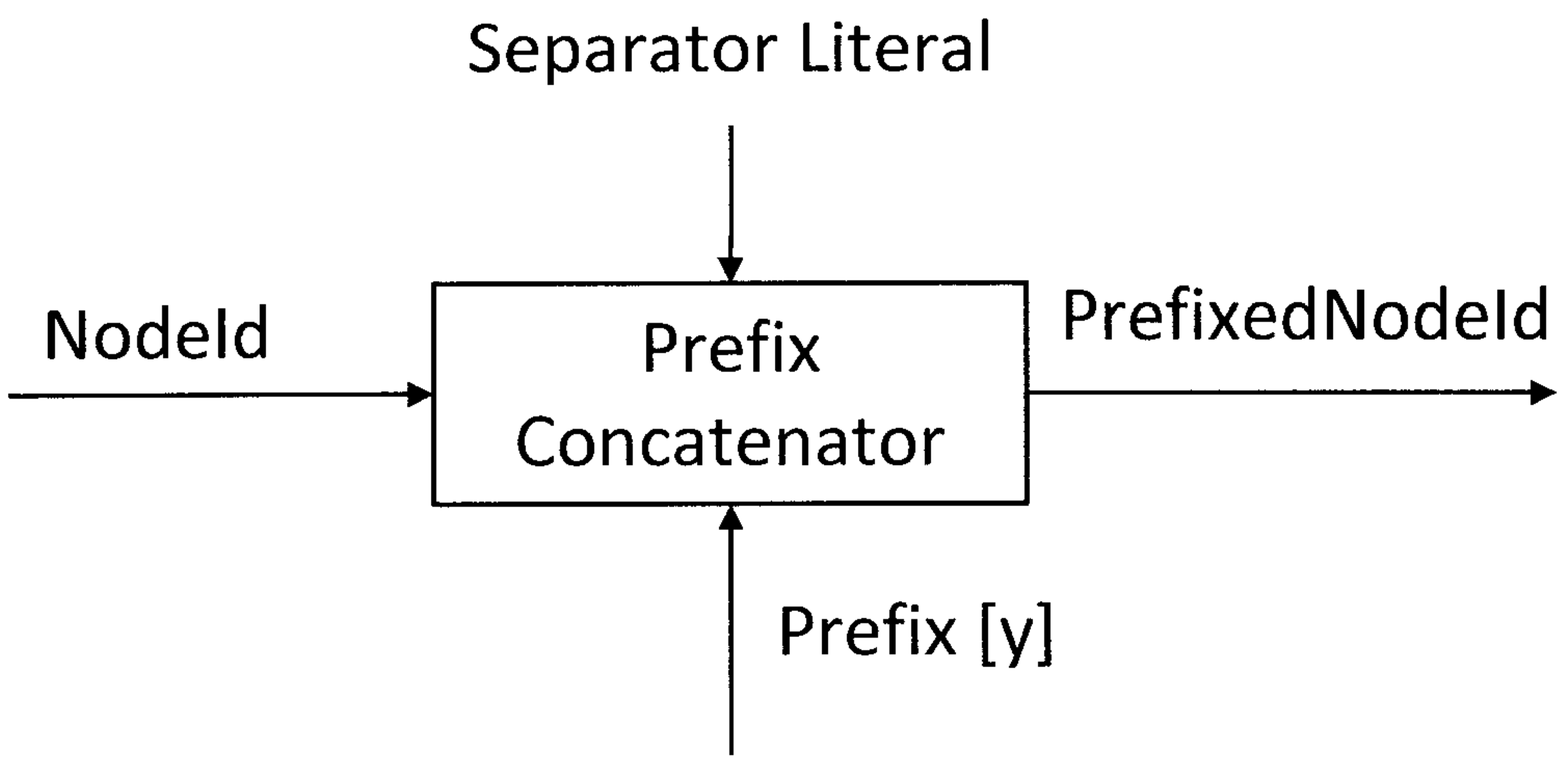

Separator Literal
NodeId
Prefix
Concatenator
PrefixedNodeId
Prefix [y]

Fig. 3

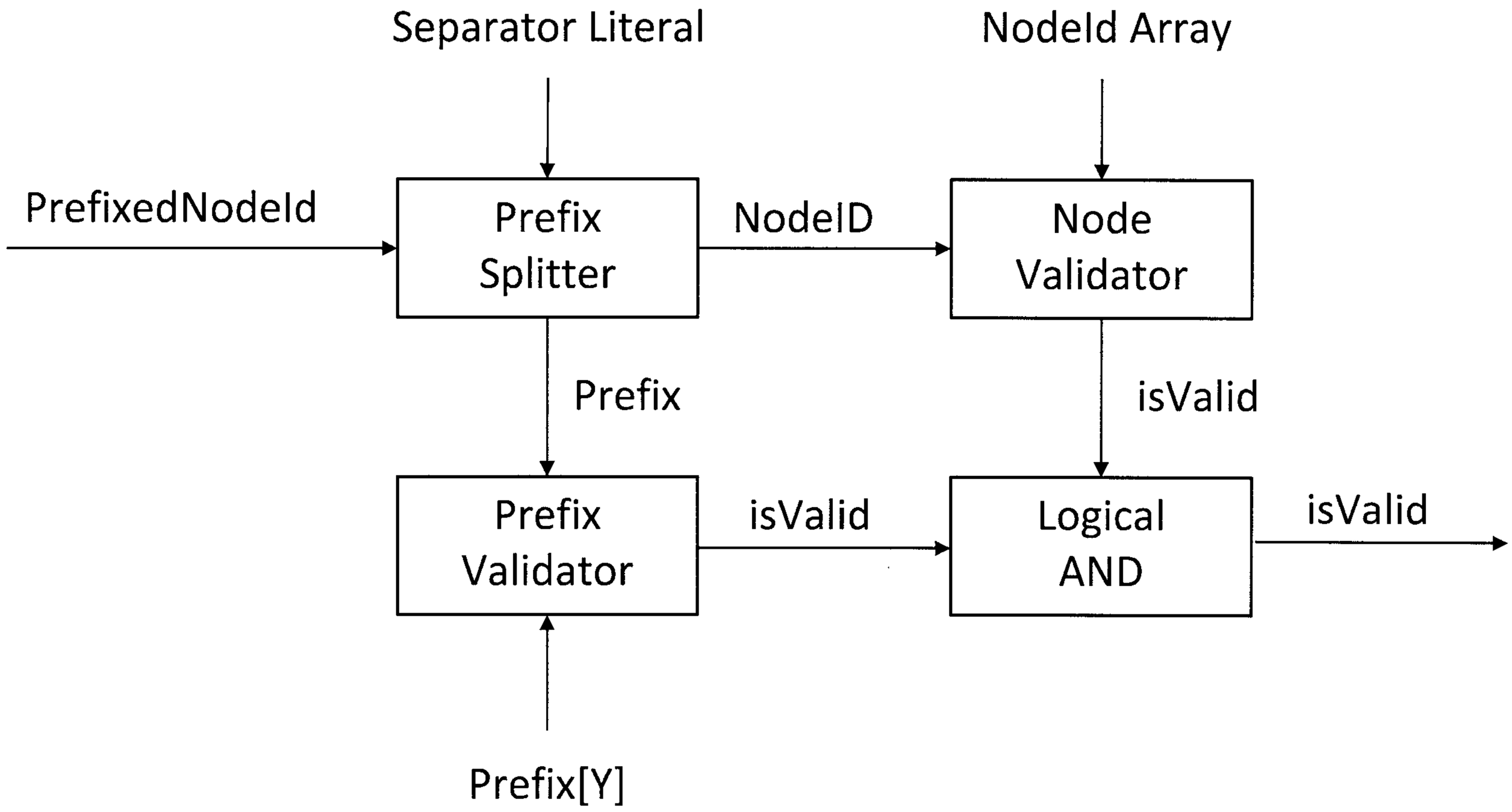
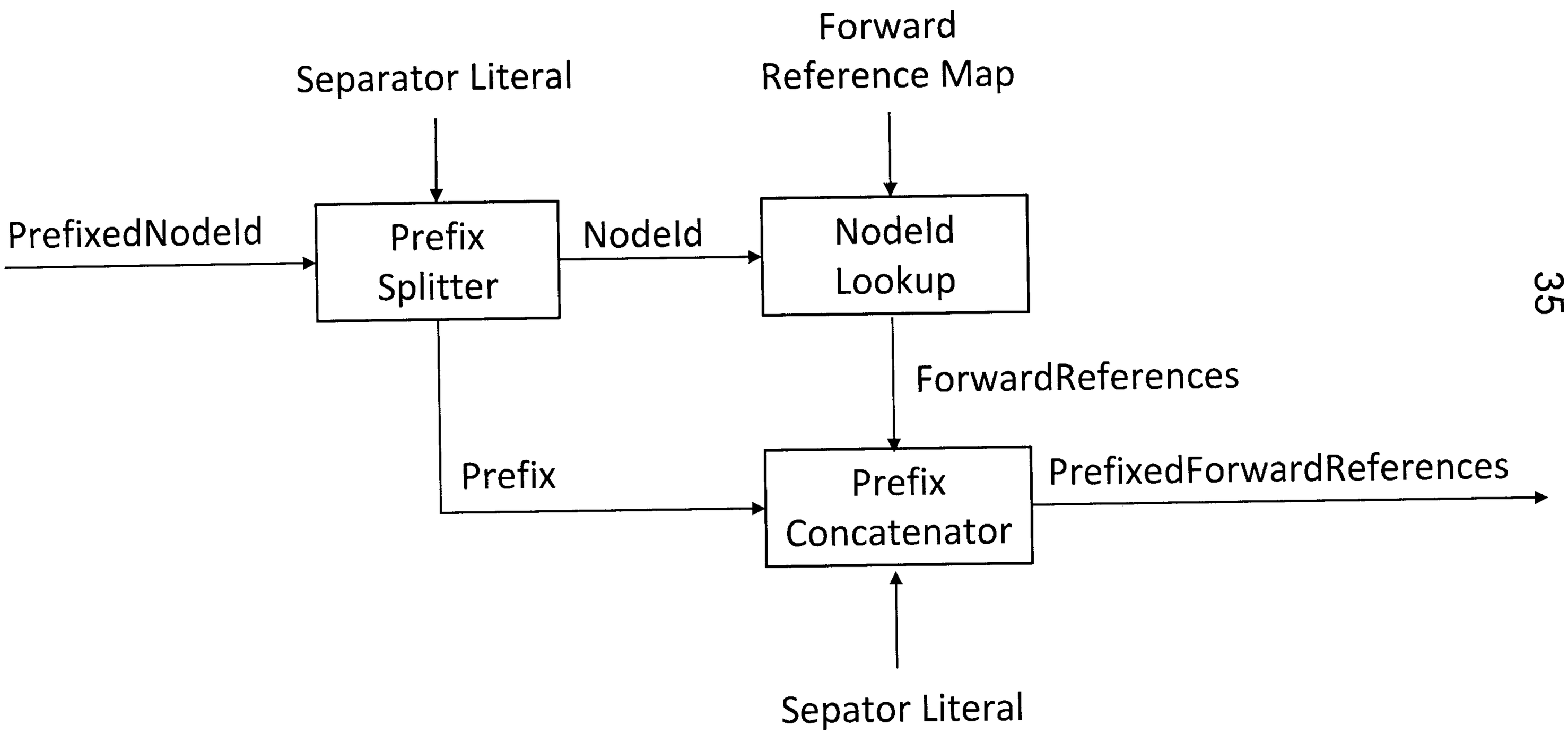

Separator Literal
NodeId Array
PrefixedNodeId
Prefix Splitter
NodeID
Node Validator
Prefix
isValid
Prefix Validator
isValid
Logical AND
isValid
Prefix[Y]

Fig. 4



LU101410

Fig. 5

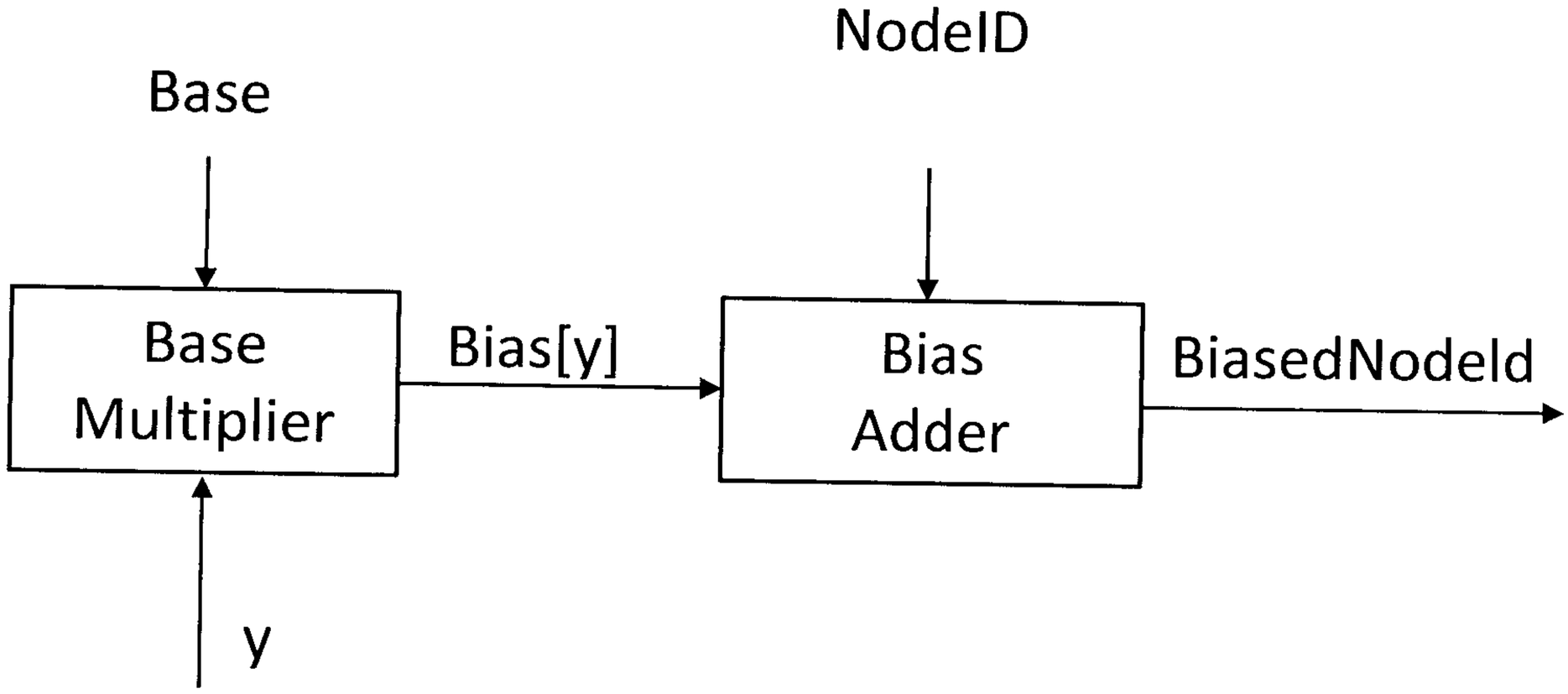

Base
NodeID
Base Multiplier
Bias[y]
Bias Adder
BiasedNodeId
y

Fig. 6

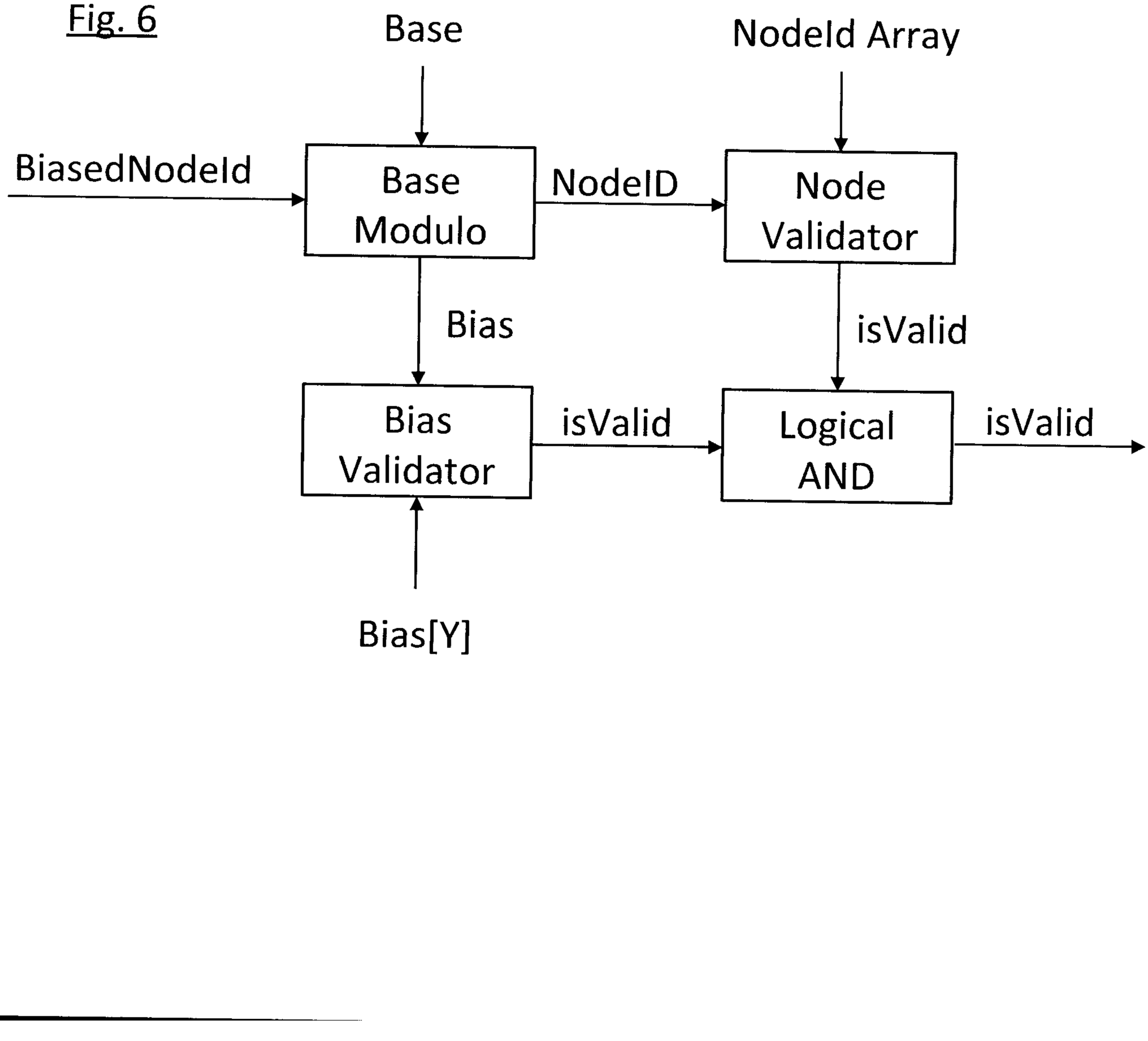

Base
NodeId Array
BiasedNodeId
Base Modulo
NodeID
Node Validator
Bias
isValid
Bias Validator
isValid
Logical AND
isValid
Bias[Y]

Fig. 7

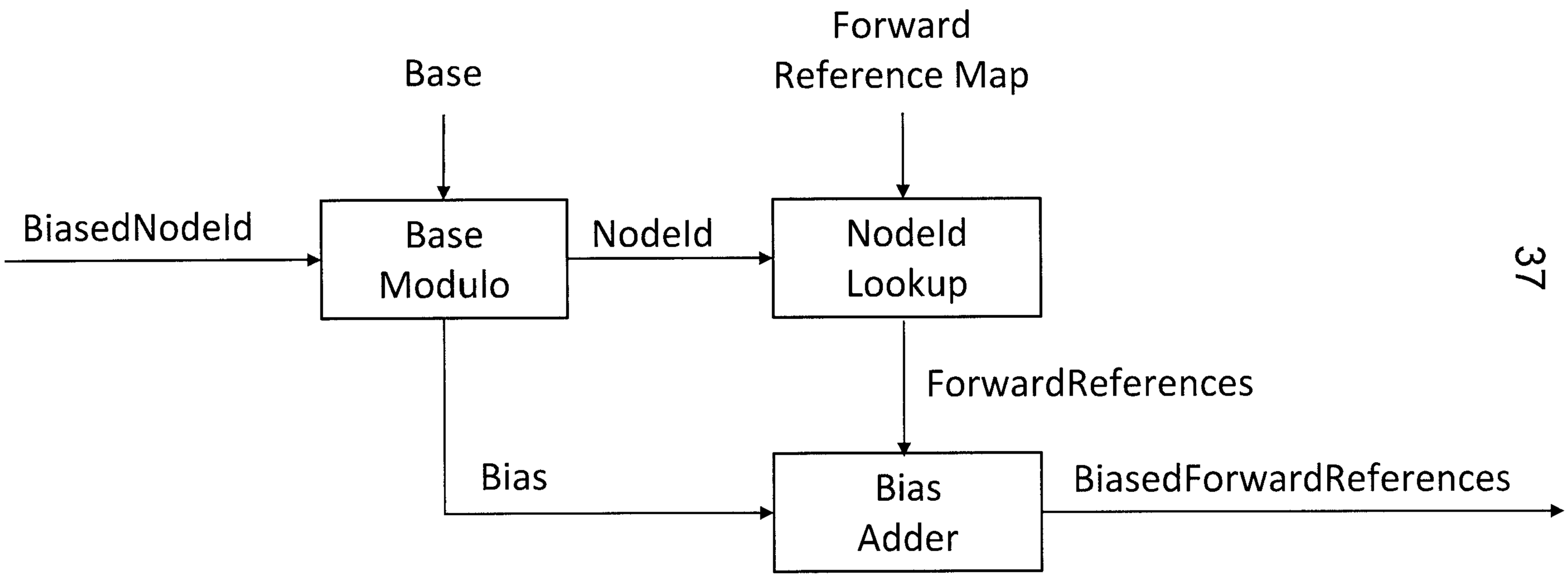


Fig. 8

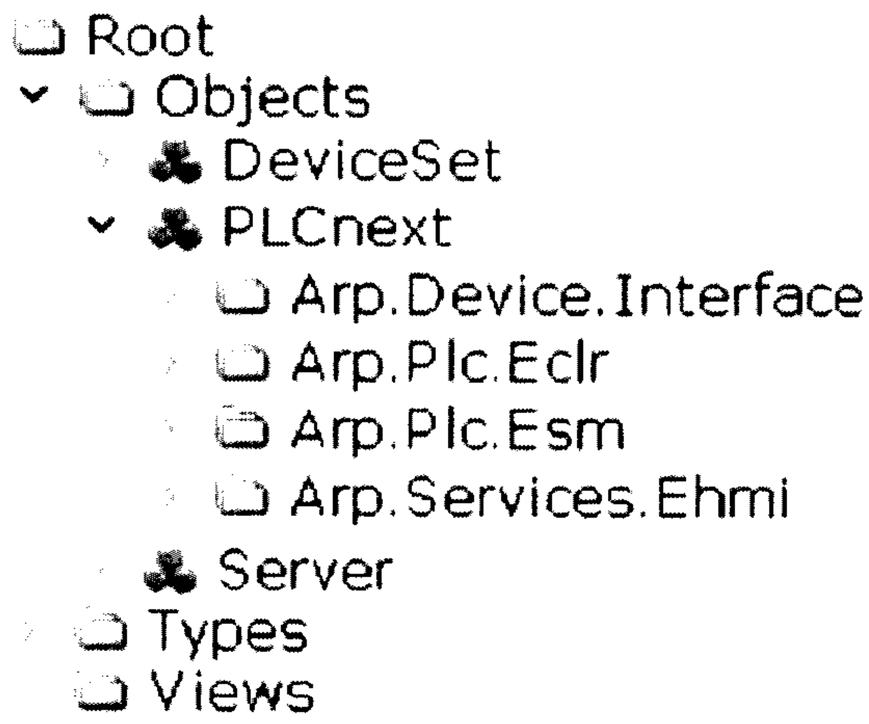

Root
Objects
DeviceSet
PLCnext
Arp.Device.Interface
Arp.Plc.Eclr
Arp.Plc.Esm
Arp.Services.Ehmi
Server
Types
Views

Fig. 9

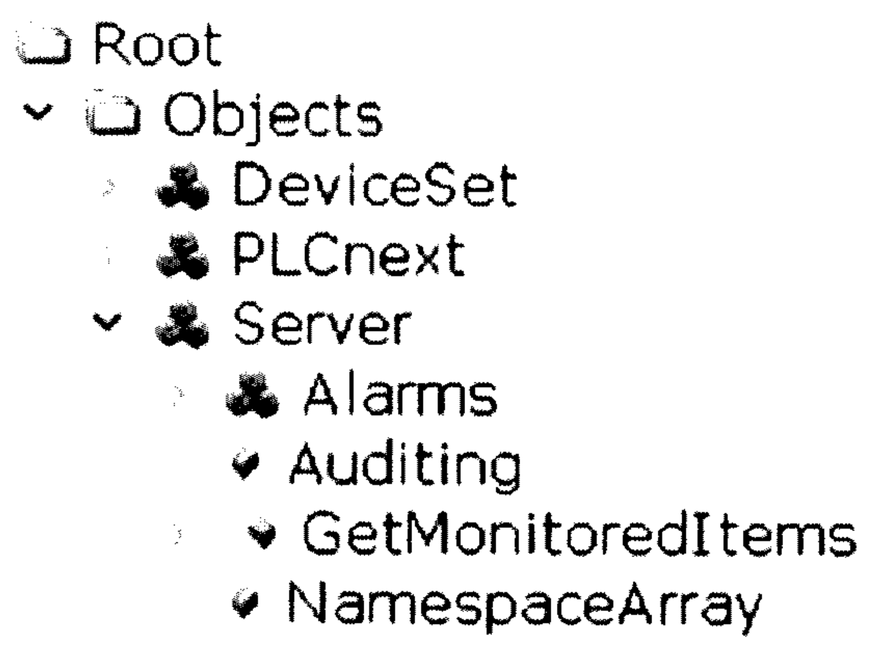

Root
Objects
DeviceSet
PLCnext
Server
Alarms
Auditing
GetMonitoredItems
NamespaceArray

Fig. 10

	PlcOpen.Programs	PlcOpen.GlobalVars
Arp.Plc.Eclr	PlcOpen.Programs:Arp.Plc.Eclr	PlcOpen.GlobaVars:Arp.Plc.Eclr
Arp.Plc.Eclr/Prog1	PlcOpen.Programs:Arp.Plc.Eclr/Prog1	PlcOpen.GlobaVars:Arp.Plc.Eclr/Prog1
Arp.Plc.Eclr/Prog1.Var1	PlcOpen.Programs:Arp.Plc.Eclr/Prog1.Var1	PlcOpen.GlobaVars:Arp.Plc.Eclr/Prog1.Var1

Fig. 11

	History.SessionA	**History.SessionB**
Arp.Plc.Eclr	History.SessionA:Arp.Plc.Eclr	History.SessionB:Arp.Plc.Eclr
Arp.Plc.Eclr/Prog1	History.SessionA:Arp.Plc.Eclr/Prog1	History.SessionB:Arp.Plc.Eclr/Prog1
Arp.Plc.Eclr/Prog1.Var1	History.SessionA:Arp.Plc.Eclr/Prog1.Var1	History.SessionB:Arp.Plc.Eclr/Prog1.Var1

LU101410