

Zweck

Die Ideen-Inventur dient dazu, alle bisherigen Ideen zusammenzuschreiben, um am Samstag, dem 18.6.2005 mit Christian Schöbel darüber diskutieren zu können (noch mal brain storming).

Im letzten Kapitel wird ein Bericht über das Treffen niedergeschrieben.

Ideen

Drfc0001 (digits_v05.doc):

- 3D (2D) Abbild der Welt wird in einer verteilten Datenbank im Internet bereitgestellt, um es für verschiedene Anwendungen zu benützen
 - o Vorteile der Bereitstellung in einer verteilten Datenbank:
 - jeder Anbieter kann sich auf seinen Zuständigkeitsbereich spezialisieren
 - Netzlast nicht so groß, wie bei einem zentralen Server, da man in vielen Fällen 3D-Daten benötigen wird, „in deren Nähe man sich befindet“
 - o Vorteile der Bereitstellung im Internet:
 - Jedes Datum wird nur einmal gespeichert (keine Redundanz)
 - Aktualisierungen von Daten lassen sich schneller an die Benutzer weitergeben
 - Der Digits Request wird immer an den DIGITS Server mit der niedrigsten Hierarchiestufe gesendet
 - o Es wäre unmöglich, alle Digits Requests zum Root Server zu senden (an einem Punkt konzentrierte Netzlast ==> keine Skalierbarkeit des Systems)
 - o Cacheing
 - Der wichtigste Parameter (Primärschlüssel) im Digits Request ist die *Virtual Roaming Area* (VRA). Diese ist ein Polygon, innerhalb dessen der Benutzer seine virtuelle Präsenz bewegen möchte.
 - Während der Bearbeitung des Digits Requests setzt die Hierarchie der DIGITS Server die virtuelle Realität bezüglich der VRA zusammen. Dabei ist der Server mit der niedrigen Hierarchiestufe prinzipiell für die Details (kleinen Objekte) verantwortlich und der Server mit der höheren Hierarchiestufe für den Horizont und die großen Objekte.
 - o Die VRA wird beim Durchreichen durch die Server-Hierarchie nicht geändert
 - o Jeder Server muß aufgrund der VRA und der eigenen Hierarchiestufe erkennen, aus welchem Umkreis er seine Objekte beizusteuern hat
- Die Zusammensetzung der virtuellen Realität bezüglich einer VRA wird in Abbildung 1 dargestellt.
- Applikationen:
 - o Eine Gegend kennenlernen
 - Frage: Wie fügt ein Architekt sein Projekt in die DIGITS-Datenbank ein, sodaß es Bürgerbeteiligung bei der Bewertung des Projektes geben kann?
 - o Geographische Suchmaschine
 - Einstieg per default an dem Ort, an dem sich die abfragende Maschine zur Zeit befindet
 - Links in der geographischen Suchmaschine können „betreten“ werden. Alternativen: (Kurz-) Information wird direkt dargestellt, Link auf eine klassische Homepage.
 - Auch 2D-Karten in der geographischen Suchmaschine verwenden
 - o VR-Navigation (heißt mittlerweile auch 3D-Wegefindung)
 - „gelbe Linie“
 - Verteilung der Kosten: grundlegende DIGITS Infrastruktur: öffentliche Hand; Straßenzug: Geschäftsleute gemeinsam; Eingangsbereich des Geschäftes: jeder Geschäftsmann selber.
 - Gemeinsamer Navigationsserver für mehrere Geschäftsleute (Hotels, Geschäfte, ...)
 - Für die Hierarchie der DIGITS Server wird das DNS verwendet (wenn ich einen Request an *city.wien.digits-root.gv.at* schicke, werden der Reihe nach die Server *city.wien.digits-root.gv.at*, *wien.digits-root.gv.at* und *digits-root.gv.at* durchlaufen).
 - Es gibt für jeden Server einen Zuständigkeitsbereich. Der Zuständigkeitsbereich eines untergeordneten Servers ist immer ganz im Zuständigkeitsbereich eines übergeordneten Servers enthalten. Der Zuständigkeitsbereich ist nur wichtig für die Index-Abfrage.
 - Es gibt eine Index-Abfrage, bei der ein Digits Index Request an den root Server gerichtet wird, um für eine bestimmte *Area of Interest* herauszufinden, an welche(n) Server der Digits Request geschickt werden muß. Parameter des Digits Index Request: *Area of Interest*, *Level of Detail*, *Reality*.
 - Sollte auch ein Mobilfunkbetreiber DIGITS Server betreiben, dann könnten ein oder mehrere der folgenden Parameter für die Definition des Zuständigkeitsbereiches verwendet werden: Service Area, Roaming Area, Location Area,

- Koordinatensystem, Projektion, WGS84 muß unterstützt werden. Im Digits Request und im Digits Index Request muß auch angegeben werden, welches Koordinatensystem und welche Projektion verwendet werden sollen.
- Daten von zwei oder mehreren parallelen Servern werden benötigt ==> iterativer Ansatz und rekursiver Ansatz
- Authentifizierung / Authorisierung / Accounting: Identifikation des Users wird mitgeschickt. Daten werden so verschlüsselt, dass nur dieser User sie lesen kann. Symmetric Key: bei wenigen Usern mit häufigen Abfragen. Unsymmetrisches Verfahren mit öffentlichem Schlüssel.

Drfc0002 (erg_digits_v01.doc)

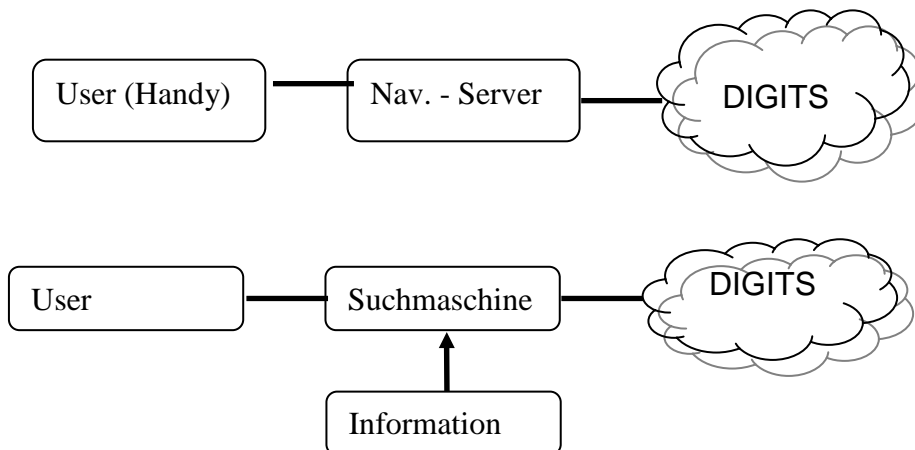
- Mobilfunk: Rendering könnte von externem Server übernommen werden (muß nicht im Endgerät vor sich gehen). „VR-Server“ nimmt Befehle entgegen (vorwärts/rückwärts/links/rechts/....) und sendet einen Video-Stream zum Endgerät.
- Es muß keine neue TLD eingeführt werden (Digits Root Server muß nicht *.digits* heißen).
- Definition von Nachbar-Servern: Keine überlappenden Objekte an den Grenzen. Eventuell kann die Index-Abfrage dadurch ersetzt werden (?).
- Überladen von Objekten. Dafür ist eine *Benennung* der Objekte nötig. Siehe Abbildung 2.

Drfc0003 (digits\$\$\$\$\$\$\$\$.doc)

Keine neuen technischen Ideen, bloß ein kurzes Brainstorming, wer bei welcher DIGITS Applikation an wen Geld zahlt, um die Applikation am Laufen zu halten.

Drfc0004.ppt: (DIGITS Sitzung 0001)

- Einige mögliche Quellen für die 3D-Daten werden aufgezählt (Folie 6)
- BTS Idee (3D-Scanner wird auf BTS montiert, um Umgebungsplan stets aktuell zu halten) (Folie 6)
- Definition einiger Begriffe: DIGITS, das Protokoll; DIGITS, die GeoIS (Geographische Infrastruktur); DIGITS, ein Service des Internet (Folie 8)
- VR-Navigation mit verschiedenen virtuellen Fortbewegungsmitteln (Fußgänger, Auto) (Folie 9)
- Geographische Suchmaschine mit verschiedenen virtuellen Fortbewegungsmitteln (Fußgänger, Auto, Hubschrauber) (Folie 11)
- Blockdiagramme:



- Idee für Anwendung: Mountainbike-Strecken 3-dimensional „erfahrbar“ (Folie 26)
- Es wird auf das Vereinheitlichungsproblem hingewiesen (Folie 26); Anmerkung: dem begegnen wir durch Standardisierung, z.B. im Web3D Konsortium (<http://www.web3d.org/>). Außerdem ist DIGITS ein hierarchisches System: Jeder untergeordnete DIGITS Server muß sich beim übergeordneten Server anmelden. Der Betreiber des übergeordneten Servers hat auch eine Verantwortung für die untergeordneten Server.

Drfc0005.txt: Einige Gedanken zum Thema „WAS wollen WIR WEM verkaufen?“

Drfc0006.txt: Aufgrund der Überlegung, dass man die DIGITS Inhalte auch für militärische / terroristische / verbrecherische Zwecke nutzen könnte (so wie jede Landkarte und jeden Lageplan), komme ich zu der Behauptung, dass es eine Instanz geben müssen wird, die die Inhalte der DIGITS Server kontrolliert.

Drfc0007.txt: Ich stelle mir die Frage, ob man die Daten innerhalb der DIGITS-Server und der im DIGITS-Protokoll übertragenen Daten auch mit Hilfe von XML (bzw. vielleicht sogar ASN.1) darstellen/definieren kann.

Diese Frage ist mittlerweile überholt, da ja die Weiterentwicklung von VRML, X3D, sich mit genau diesem Thema beschäftigt (siehe <http://www.web3d.org/>). Die Frage, ob sich X3D/VRML für die Verwendung in DIGITS eignet, bleibt allerdings noch offen.

Drfc0008.txt: Meinung: prinzipiell sollte jeder das Recht haben, einen Digits Root Server zu betreiben. Er ist dann aber auch verantwortlich für das „Weltbild“, das auf diesem und den untergeordneten Servern hinterlegt wird. Hinter dieser Meinung steht die – philosophische – Erkenntnis, dass auf einem Digits Root Server und seinen untergeordneten Servern genau ein „Weltbild“ gelagert ist (im wahrsten Sinne des Wortes). Da es aber so viele Weltbilder gibt, wie es Menschen gibt, sollte jeder das Recht haben, sein Weltbild im Internet darzustellen – dass sich das niemals rechnen wird, ist eine andere Geschichte. Ein abgeschwächter Ansatz dieses Postulats ist die Idee, in einem DIGITS System mehrere *Realities* zu definieren, sodaß man die Möglichkeit hat, sich in das DIGITS Weltbild „einzuhängen“ und bloß einige Objekte zu überladen/hinzuzufügen/(zu löschen).

Drfc0009.ppt (DIGITS Sitzung 0002): Es wird versucht, dahinter zu kommen, wie Siemens mit DIGITS Geld machen könnte. Einige potentielle Interessenten für 3D-Navigation werden aufgezählt.

- Reed Messe
- Flughafen
- Bahnhof
- Tiergarten / Prater

Drfc0010.txt: ist irrelevant

Drfc0011.doc: beschreibt ein „Projekt“, um DIGITS einem breiteren Messe-Publikum bekannt zu machen. Prototyp.

Drfc0012.ppt: VR-Navigation (3D-Wegefindung) wird kurz vorgestellt als mögliche Mobilfunk-Applikation, da es bei 4G angeblich um VR gehen könnte (wie man im Dezember 2004 behauptet hat). Diese Folien sind mehr oder weniger aus den bisherigen Folien „zusammenkopiert“. Allerdings wird bei der „Verwendung von GPS Koordinaten“ umformuliert auf „Verwendung von GPS/Galileo-Koordinaten“.

Drfc0013.doc:

- DIGITS berechnet selbsttätig eine Landscape-Textur, die für die 3D-Darstellung verwendet werden könnte.
- Session basierter Ansatz: Ein Digits Request entspricht nicht dem Laden eines Levels in einem 3D-Game, sondern es wird eine Session mit dem DIGITS System aufgebaut und während sich der Betrachter durch die virtuelle Wirklichkeit bewegt, werden immer wieder Daten nachgeladen, je nach Standort, Blickrichtung (und zukünftiger Bewegung ?).
- DNS studieren, um DIGITS besser entwerfen zu können

Drfc0014-0016: Machbarkeitsstudie Institut für Eisenbahnwesen/ Wiener Linien ????? Eventuell bekommen wir im September einen Termin, um die Idee vorzustellen. Es geht um

- 3D Wegefindung in (und um) U-Bahn Haltestellen.

Ideen ohne DRFC:

- Beim Digits Index Request gibt es einen dritten Parameter, die *Reality*. Unter einem Digits Root Server können mehrere Wirklichkeiten gelagert sein (parallele Server mit überlappenden Zuständigkeitsbereichen). Diese „Paralleluniversen“ werden durch einen String (Namen), eben die *Reality*, voneinander unterschieden, der vom Betreiber des Root Servers vergeben werden sollte.

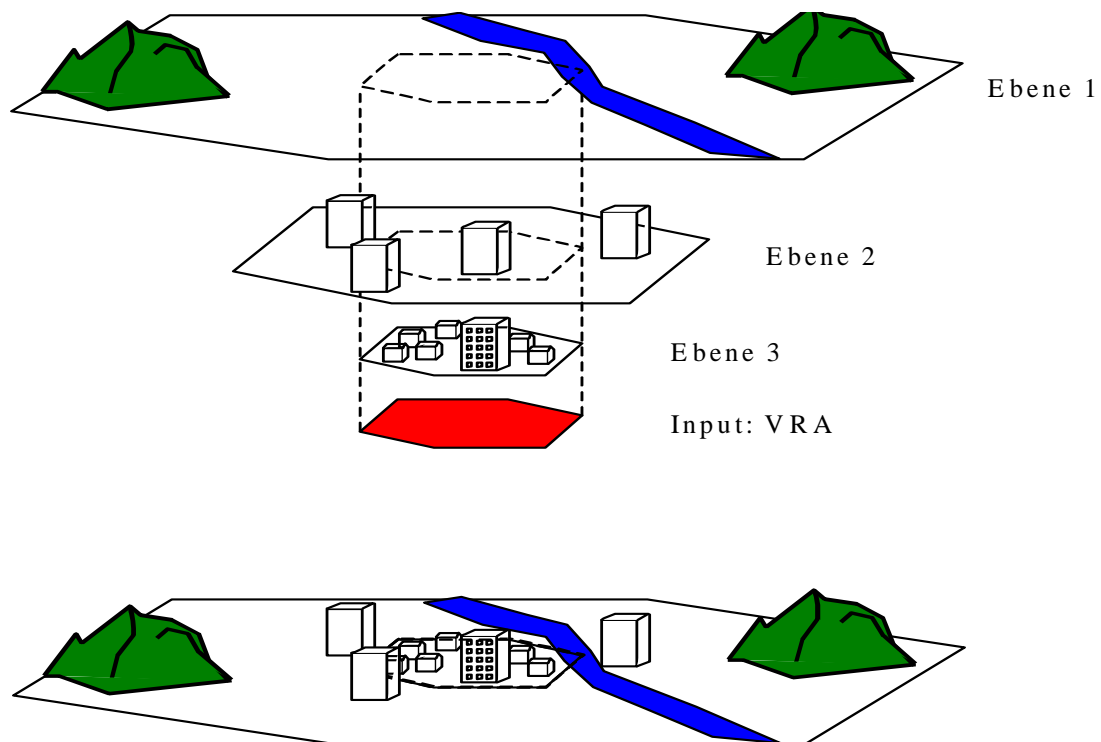


Abbildung 2: Zusammensetzung des Ergebnisses mit Überladen von Objekten

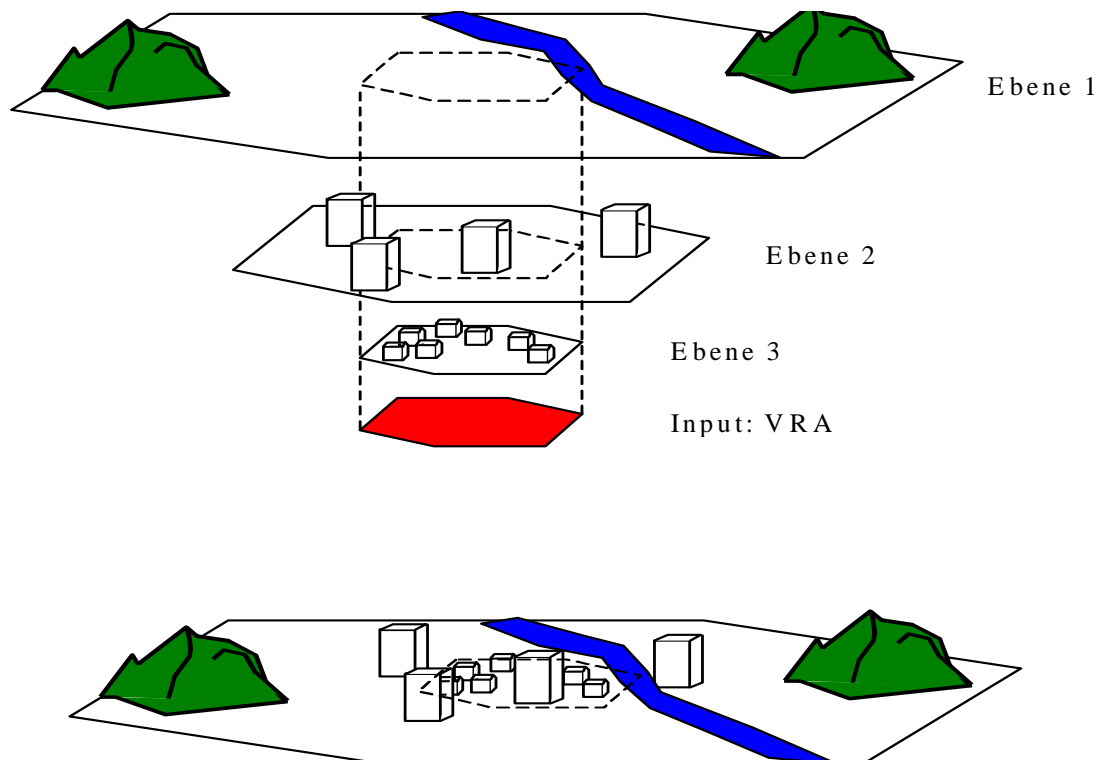


Abbildung1: Zusammensetzung des Ergebnisses ohne Überladen von Objekten

Bericht von der DIGITS Sitzung 0003

Ort: Salettl, Zeit: Samstag, 18.6.2005, 10:00-11:55, Anwesend: Christian Schöbel, Christoph Valentin

Es wurden einige neue Ideen hinzugefügt.

- prinzipiell sind wir uns einig darüber, dass das wesentliche Problem von DIGITS darin besteht, dass die Verschaffung der 3D-Daten teuer ist (können die Applikationen wirklich die Bereitstellung von DIGITS finanzieren?) → Die DIGITS-Hierarchie wird, wenn überhaupt, so ähnlich entstehen, wie das DNS: zuerst gab es etc/hosts, solange, bis die Wartung und Übertragung dieses Files nicht mehr tragbar war → Leidensdruck → Investment in das DNS.
- AbN: Folglich: Der Ansatz, vorerst einmal bei EINEM Anwender (z.B. Wiener Linien) abzuklopfen, ob 3D Grafik überhaupt wirtschaftlich tragfähig sein kann (und im Mobilfunk realisierbar ist), erscheint goldrichtig. Dazu werden wir vorerst wahrscheinlich kein DIGITS benötigen (Wiener U-Bahn-Stationen alle auf einem Server).
- Neue Idee für eine Anwendung: Game-Konsole (z.B. Play-Station) mit irgendeinem Ego-Shooter, den ich aber in meiner eigenen Umgebung spielen kann – Gegend beliebig runtergeladen vom Netz. AbN: Vorteil: Der Hersteller des Spiels braucht nicht viele verschiedene Varianten des Spiels vertreiben, sondern kann die 3D-Daten im Internet warten, hat hierbei eventuell Synergien mit Fremdenverkehrsvereinen o.ä (Werbung einblenden!).
- Bezüglich des Parameters *Reality* gab es eine Verbesserung: Bei zwei verschiedenen *Realities* gibt es viele Objekte, die in beiden Wirklichkeiten identisch sein werden, bloß einige Objekte (z.B. zweisprachige Ortstafeln ;-)) werden sich unterscheiden. Deshalb erscheint der Ansatz, dafür zwei getrennte Server mit identischem Zuständigkeitsbereich aufzusetzen, suboptimal zu sein (doppelte HW-Kosten, Redundanz der Daten). Besser ist es, bloß einige Objekte je nach *Reality* innerhalb eines Servers zu überladen (AbN: wenn nicht beide *Realities* von unterschiedlichen Personen verwaltet werden).
- Filter-Eigenschaften der Server: Man könnte auch einen Server eine Hierarchiestufe niedriger daranhängen, der einige Objekte überlädt (z.B. japanische Botschaft richtet einen Server ein, in dem alle Anschriften auf japanisch dargestellt werden). Wieder einmal sieht man, dass die eindeutige Namensgebung der Objekte wesentlich ist. AbN: Vorschlag für die Namensgebung: Der Name eines Objektes besteht aus zwei Teilen: a) DNS-Name des höchsten Servers, auf dem dieses Objekt vorhanden ist (Objekte können ja auf unterschiedlich hohen DIGITS-Servern gleichzeitig gespeichert werden, je nachdem mit mehr oder weniger Details; „oben“ heißt dabei „näher“ beim Root-Server); b) ein Name, der innerhalb dieses Servers eindeutig ist. Beispiel: Ein Objekt existiert auf den Servern *salettl.doebling.cis-danubien.digits-root.wien.gv.at* und *doebling.cis-danubien.digits-root.wien.gv.at*, einmal mit mehr Details und einmal mit weniger Details (für den Server *cis-danubien.digits-root.wien.gv.at* ist es jedoch zu klein), dann könnte das Objekt den Namen *doebling.cis-danubien.digits-root.wien.gv.at/oktogen* tragen. Dieser Name muß auch in allen untergeordneten DIGITS-Servern für dieses Objekt verwendet werden. Immer, wenn ein Betreiber eines DIGITS-Servers ein Objekt neu hinzufügt oder eines löscht, muß er die Betreiber der untergeordneten DIGITS-Server davon in Kenntnis setzen (das wird noch ein kniffliges Problem, sobald es an die tatsächliche Realisierung geht).
- Die Idee, dass jeder DIGITS Request durch die gesamte Hierarchie bis zum Root Server „durchschlägt“, ist untragbar, da dann das System nicht skalierbar wäre.
 - o Der erste Ansatz war ein Cacheing, wobei jeder DIGITS-Server auch Objekte der übergeordneten Server in einem Cache bereithält. Die Objekte von den höheren Servern müssten dabei wohl eine längere Lebenszeit im Cache bekommen. Das würde aber bedeuten, dass gerade große Objekte länger bräuchten, bis sie aktualisiert werden.
 - o Ein weiterer Ansatz wäre, mehrere DIGITS Root Server zu verwenden, die alle dieselben Daten halten, aber die Digits Requests und DIGITS Index Requests auf diese Server zu verteilen. Eine Skalierung des Systems könnte dann durch das Aufstellen neuer Digits Root Server erfolgen. Allerdings sollten wir aus den Fehlern des DNS lernen: Jeder DNS name Server muß ein File enthalten, welches zentral gewartet wird und vom Administrator runtergeladen und installiert werden muß. Dieses File ist quasi der „Nachfolger des etc/hosts“ und enthält die IP Adressen aller DNS root server.
 - o Ein weiterer Ansatz wäre der einer „replizierten Datenbank“. Das heißt, der DIGITS Request wird prinzipiell von dem DIGITS Server abgearbeitet, an den er vom DIGITS User gesendet worden ist. Es gibt kein rekursives Durchlaufen der Hierarchie. Immer wenn ein Objekt in einem übergeordneten DIGITS-Server modifiziert worden ist (angelegt/geändert/gelöscht), werden alle betroffenen untergeordneten DIGITS-Server auch mit diesem Objekt upgedatet.

AbN: Das heißt wohl, dass jeder untergeordnete DIGITS-Server alle Objekte erhält, die in seinem Zuständigkeitsbereich+ liegen. (Zuständigkeitsbereich+ ist der Zuständigkeitsbereich plus ein Sicherheitsabstand, der für große Objekte größer, und für kleine Objekte kleiner ist).

AbN:

Mein persönlicher Gewinn aus diesem Gespräch: ich glaube wieder an die technische (!) Realisierbarkeit von DIGITS (auch wenn noch viele Fragen zu beantworten sein werden, sobald die Zeit dafür gekommen ist). Eine Frage, die mich prinzipiell quält: Ist der hierarchische Ansatz prinzipiell nötig, oder würde es nicht reichen, Server anzuordnen, die einfach „nebeneinanderliegende“ Zuständigkeitsbereiche haben plus einem Directory Server, der mir hilft, den zuständigen Server zu finden? Ein Argument, das für die Hierarchie spricht, ist die Abwägung von „Level of Detail“ und „Größe der VRA“: Wenn man davon ausgeht, dass ein User eine konstante Zeit mit der Erforschung der Daten aus einem Digits Request verbringt, dann kann man folgende Aussage treffen: „Je schneller das virtuelle Fortbewegungsmittel ist, mit dessen Hilfe der User sich durch die virtuelle Welt bewegen möchte, desto größer wird die VRA sein, die er benötigt. Im Gegenzug wird der Level of Detail geringer sein. Das heißt, dass der User immer mit einem einzigen Digits Request auskommen sollte, um eine bestimmte Zeit mit den Daten zu verbringen. Wenn also das virtuelle Fortbewegungsmittel schneller ist, dann sollte der Digits Request an einen Digits Server einer höheren Hierarchiestufe gesendet werden, der ja einen größeren Zuständigkeitsbereich, dafür aber einen geringeren Level of Detail innehat. Bei einem langsameren Fortbewegungsmittel ist es genau umgekehrt“.

AbN: Beim nochmaligen Kontrolllesen dieses Berichtes ist mir ein wesentliches Problem der bisherigen Lösung aufgefallen: Angenommen, DIGITS beginnt „bottom up“, also in Wien gibt es jemanden, der einen Digits Root Server installiert (*digits-root.wien.gv.at*), in New York gibt es jemanden, der dasselbe tut (*root.digits.nyc.gov*), etc. etc.

Wie soll man dann jemals die verschiedenen Digits Root Server durch einen einzigen Digits Root Server ersetzen (sobald man sich entschlossen hat, die Systeme zusammenzuführen), wenn man das bisherige Namens-Konzept hat? (alle untergeordneten Server aller verschiedenen Root Server müssen dann die Domain wechseln oder zumindest einen zusätzlichen DNS-Namen annehmen. Die Applikationen, die bisher bei „ihrem“ Digits Root Server angefragt haben, müssten alle umkonfiguriert werden, sodaß sie nun den neuen einzigen Root Server fragen).

Man müsste also während der Realisierung von DIGITS (also bereits während der Standardisierung?) die DNS-Hierarchie der DIGITS Server „von oben nach unten“ definieren – wer dann den Digits Root Server tatsächlich betreibt, ist eine andere Frage. Der DNS-Name „des“ DIGITS Root Servers müsste in der Standardisierung klar und deutlich festgelegt werden – oder zumindest muß sich eine passende NIC-Autorität finden, die diesem Server eine „passende“ Domain zur Verfügung stellt.

Was ist eigentlich der Vorteil der Verwendung des DNS für die DIGITS Hierarchie?: Vereinfachung der Administration: Jeder DIGITS Server findet „ganz von selbst“ nur aufgrund des eigenen DNS-Namens seinen übergeordneten Digits Server und kann sich bei ihm anmelden (unter Bekanntgabe des eigenen Zuständigkeitsbereiches und aller möglichen *Realities*, die in und unterhalb von ihm selbst existieren).

Ergebnis dieser „Diskussion mit mir selbst“: ich denke, wir sollten bei dem Konzept bleiben, DNS-Namen für die Hierarchie der Digits-Server zu verwenden, DNS ist ein gut bekanntes und gut verstandenes Feature des Internet und die Admins können gut damit umgehen. Wenn wir eine neue Digits-Hierarchie einführen würden, müssten wir zusätzliche Beziehungen zwischen den Servern administrieren (DNS-Namen müssten sie ohnedies bekommen) und die Admins hätten mehr zu lernen. Der erste Betreiber eines DIGITS Servers müsste sich allerdings Gedanken über die Hierarchie machen. Er würde wohl zuerst nur einen Server betreiben (was die HW betrifft), dieser würde aber mehrere DNS Namen bekommen: z.B.: <root>, .europe.<root> und vienna.europe.<root> und alle Objekte enthalten. (<root> ist ein „geeigneter“ DNS-Name). Aber vielleicht sollte man auch mit einem DIGITS Server beginnen, der nur den Namen <root> trägt und alle Objekte enthält. Man müsste dann bloß bei den Tools für die Erstellung der 3D Daten darauf achten, dass Objekte leicht zwischen verschiedenen Servern und Hierarchien hin- und hergeschoben werden können.

Dieser Betreiber des Digits Root Servers müsste Vorkehrungen treffen, dass sich alle folgenden Digits Betreiber seinem root Server „unterordnen“. Das scheint nach wie vor ein „atmosphärisches Problem“ zu sein: wir sollten von Anfang an trachten, jemanden zu überzeugen, den Digits Root Server zu betreiben, der „von allen anerkannt“ und „unparteiisch“ ist. Auch ein „neutraler“ DNS-Name für <root> scheint notwendig zu sein. Oder wir verwerfen das Ziel, dass die Systeme zusammenwachsen können und leben von Anfang an mit der Tatsache, dass es getrennte DIGITS-Welten geben wird, die in Konkurrenz miteinander sind und jeder für sich auf eigene Faust die 3D-Daten besorgen wird. Eine weitere Möglichkeit: Es zulassen, dass mehrere Digits Root Server nebeneinander entstehen, und dann „bei Bedarf“ ein Zusatzsystem entwickeln, das ein gemeinsames Directory darstellt. Es stellt sich die Frage, ob es nötig ist, dass jede Applikation auf 3D-Daten aus aller Welt zugreifen kann.