

## Kurzbeschreibung

# taraVRoptimizer

**Autor:** DR. Jörg Hamel und Herbert Beesten, tarakos GmbH

Stand 6.2002

### Allgemeines

Der **taraVRoptimizer** wird eingesetzt, wenn aus 3D-CAD-Systemen bzw. aus 3D-Planungs- und Projektierungstools Projekte als VRML2-Dateien exportiert, und diese dann für die weitere Verwendung im taraVRbuilder optimiert und angepasst werden. Die so optimierten VRML2-Dateien können dann durch die VRML-Importfunktion in den taraVRbuilder importiert und dort weiterverwendet werden (mit Möglichkeit der Skalierung und Übernahme von Animationen).

So ist es möglich, innerhalb des taraVRbuilders aus verschiedenen Systemen VRML-Dateien zu übernehmen und gemeinsam in einer virtuellen Szene darzustellen. Diese Szenen können dann gestaltet und ergänzt werden, auf Wunsch auch wieder als VRML2-Datei exportiert werden.

Der **taraVRoptimizer** wird als Programmoption in den taraVRbuilder integriert und kann dann innerhalb der Importfunktion durch den Anwender wahlweise die Importfunktion ergänzen.

### Struktur der Anwendung

Der **taraVRoptimizer** ist in Java unter Visual J++ und Visual Basic realisiert. Als Bibliotheks-Datenbank wird MSAccess über ADO genutzt. Der **taraVRoptimizer** kann vordefinierte Operationen an VRML-Strukturen automatisiert im Batchmodus abarbeiten und im Bedarfsfall den Exportoperationen aus den Planungswerkzeugen direkt nachgeschaltet werden.

### Folgende Funktionen werden vom taraVRoptimizer ausgeführt:

- ?? Zerteilen der VRML-Datei durch Auslagern von Knoten mit externen Prototypen oder Inlines,
- ?? Normalendrehung,
- ?? Skalierung,
- ?? Kapselung von Geometrien, die in Weltkoordinaten plaziert sind, in Transformknoten,
- ?? Erkennung gleicher Geometrien, Referenzierung über DEF/USE,
- ?? Generierung von Level-Of-Detail-Knoten,
- ?? Einmischen der Zusatzinformationen der Planungswerkzeuge,

- ?? Ablegen neuer Objektdefinitionen in Dateien,
- ?? Entfernung dynamischer Daten

### Skriptbefehle

Der Java-Teil vom **taraVRoptimizer** optimiert eine in ihm geladene Szene auf Basis eines Skriptes.

Es können Skripte für die Optimierung von VrmI-Export-Dateien von Microstation 7.1, eM-Workplace und ergoPlan verwendet werden

Im Einzelnen gibt es folgende Möglichkeiten für die verschiedenen VRML-Exporte:

#### eM-Workplace (eM-WorkplaceNT & RobCAD)

Der VrmI-Export von eM-Workplace hat die Eigenschaft, daß fast keine Referenzierung, also der Gebrauch von USE zur Referenz auf mit DEF benannten Objekten, genutzt wird; einzige Ausnahme: Erscheinungs-(VRML: *Appearance*) Knoten.

Allerdings wird von DEF reichlich Gebrauch gemacht um Objekte zu benennen. Dieser Umstand wird von uns genutzt, um eM-Planner-IDs einzufügen und somit die Tür für weitere Zusatzinformationen zu öffnen, dazu dient der Befehl **mergeEmPlannerData**

Die Benennung der Objekte durch eM-Workplace endet auf der Ebene der Transformationen (VRML: *Transform*). Gestalt(VRML: *Shape*)-Knoten sind davon nicht betroffen und damit erstes Ziel für die strukturelle Optimierung mit dem Befehl: **findShapeDefs**.

Vor dieser Operation ist im Skript noch **simplifyIndexedFaceSets** positioniert, um den Vergleich der Geometrien zu erleichtern, denn auch wenn Geometrien gleich sind, so muß ihre Repräsentation in der Datei nicht identisch sein. Deshalb wird mit diesem Befehl eine Sortierung auf die Felder der IndexedFaceSets angewendet. Nebenbei werden auch mehrfach auftretende Koordinaten entfernt.

#### Microstation

In den von uns analysierten Microstation-Export-Dateien sind *Shapes* grundsätzlich in Weltkoordinaten definiert. Zellen in Microstation sind als Gruppen in VrmI definiert und werden mit USE referenziert. Ausnahme sind Gruppenknoten, deren Namen mit '\_MIRRORED' enden. Vermutlich sind die zugrundeliegenden Zellen durch Spiegelung anderer Zellen entstanden. Diese werden nicht mit USE referenziert, sondern mehrfach neu definiert, und tauchen in der taraVRoptimizer-Statistik als 'multidefs' auf.

#### ergoPlan

Es wurde nur eine ergoPlan-Datei untersucht. Diese zeigte keinen Bedarf für strukturelle Optimierung. Allerdings waren alle Koordinaten in Millimetern angegeben, was eine Skalierung erforderlich macht.

#### Level-Of-Detail (LOD) Erzeugung

Level-Of-Detail-Knoten (VRML: LOD) ermöglichen es, ein Modell abhängig von der Entfernung in verschiedenen Detailierungsgraden anzuzeigen. Das bringt vor allem Geschwindigkeitsvorteile beim Anzeigen. Der **taraVRoptimizer** ermöglicht das Erstellen von LOD-Knoten mit der Anweisung **createLOD:**.

## Entfernen von Knoten

**taraVROptimizer** bietet derzeit drei Anweisungen, um Knoten zu löschen, deren Reihenfolge im Skript ist egal. Die Anweisung **removeDynamicNodes** ist für die Optimierung für statische Zielsysteme gedacht und entfernt alle dynamischen Komponenten wie Sensoren, Skripte etc. **RemoveBindableNodes** entfernt dagegen Views, Background, Fog und Lichter. Dies ist nützlich, wenn mehrere Szenen zusammengefügt werden sollen.

Mit Hilfe der Anweisung **removeNode**: kann noch genauer spezifiziert werden, welche Knoten gelöscht werden sollen, die Selektion ist dabei über Knotenname (Wildcards wie \* und ? sind erlaubt) und -typ möglich.

## Import von Zusatzinformationen der Planungswerkzeuge eM-Workplace & RobCAD

### Grundlagen: Dateistrukturen

Für RobCAD gibt es, wie auch für eM-Workplace, die Möglichkeit, eM-Planner-Update-Dateien zu schreiben.

Über den Skript-Befehl "mergeEmPlannerData:" können diese Dateien in die aktuelle Szene eingemischt werden, wobei die ID als Kommentar in der VRML-Datei erscheint.

In der eM-Planner-Datei werden nur Datensätze mit den "PMToolInstance" und "Workplace\*Res" Kennungen betrachtet

### Implementierung

Die eM-Planner-Update-Datei wird in der Klasse `tara.vrml.opt.UserDataMergerEmplanner` gelesen, die Namensinformation und die eM-Planner-Id wird extrahiert und in einer Tabelle abgespeichert. Anschließend wird die Szene zweimal durchlaufen, um die Tabelleneinträge mit den Objektnamen abzugleichen.

Objekte in der Szene, deren Namen mit einem Namen aus der eM-Planner-Datei gematcht werden kann, bekommen die eM-Planner-Id als VrmL-Kommentar hinzugefügt, der Eintrag wird dabei aus der Tabelle gelöscht. Im zweiten Durchlauf werden auch Objekte in das Matching einbezogen, die mit '\_TRAFO' enden (scheinbar leere, d.h. kinder- und geometrieloze Objekte in der Szene, die aber in der eM-Planner-Datei referenziert werden).

tarakos GmbH, im Juni 2002