

Sechs Fragen zu GML

Sven Havemann

Institut für ComputerGraphik und WissensVisualisierung (CGV), TU Graz

9. August 2009

1 Findet GML bereits Anwendung in professionellen Programmen oder ist das für die absehbare Zeit vorgesehen?

GML ist zur Zeit im wesentlichen in unserer Gruppe im CGV an der TU Graz im Einsatz, wo wir es als Forschungsplattform nutzen und erweitern. Die industrielle Anwendung ist aber durchaus unser Ziel, dazu haben wir neben einigen Industrie- und Forschungsprojekten eine enge Kooperation mit dem Fraunhofer IGD in Darmstadt.

Grundsätzlich ist der Einsatz von GML immer dann sinnvoll, wenn es um komplexere 3D-Konstruktionen geht, die durch relativ wenige Parameter beschrieben werden können. Aus einer GML-Beschreibung kann sehr schnell ein 3D-Mesh erzeugt werden, für die Bildschirmausgabe oder um es als Dreiecksnetz zu exportieren. Der generative Ansatz von GML befördert das Zerlegen komplexerer Konstruktionen in wiederverwendbare Teile ("Modellbibliotheken"). Selbst eine einzelne Modellinstanz läßt sich meist rasch durch Einfügen von Parametern zu einer Modellfamilie verallgemeinern.

Die GML-Software auf generative-modeling.org kann und soll für nichtkommerzielle Zwecke frei verwendet werden. Es gibt dort zur Zeit die sehr kompakte Anwendung **TestGML** (nur 5 MB), die etwas umfangreichere Entwicklungsumgebung **GML-Studio** basierend auf **ActiveGML**, unserem ActiveX-Control, und das Browser-Plugin für IE und Firefox. Weiterhin bieten wir Windows-Entwicklern das **GMLApplet** Projekt, das ist eine DLL mit VisualStudio-Projekt, sodass man GML in andere Applikationen integrieren kann. GML läuft übrigens ebenso auf Linux und MacOS.

2 Ist grundsätzlich ein Ausbau der Sprache/Anwendung geplant und wenn ja, für welche Spezialgebiete?

Es gibt eine ganze Reihe unterschiedlicher Nutzungs-Szenarien. Einige haben wir auch schon prototypisch realisiert oder sind dabei, sie in Projekten umzusetzen.

Erweiterung bestehender Anwendungen um 3D-Modellierfunktionalität. Viele IT-Projekte haben im Grunde mit 3D zu tun, auch wenn es auf den ersten Blick nicht so scheinen mag. Anders als reine Modellbetrachter bietet GML eine volle Modellierfunktionalität, aber ohne daß man ein großes Modellierpaket wie 3DStudioMax oder eine komplexe CAD-Bibliothek wie ACIS benötigt. GML läßt sich höchst einfach in bestehende Anwendungen integrieren, dies erfordert nur ein OpenGL-Ausgabefenster und das Hinzulinken einer DLL. Auch das API ist sehr simpel: Ist eine GML-Modellbibliothek geladen, kann die Host-Applikation kurze GML-Codestücke an die GML schicken, um hochkomplexe Geometrien sehr schnell erzeugen und anzeigen zu lassen. Modelle können auch interaktiv verändert werden, und der veränderte Parameter-Zustand kann auch abgefragt und als GML-Code gespeichert werden.

3D-Modellierung für Spezialanwendungen. Es gibt viele Anwendungsfälle, wo 3D-Modelle ganz spezifischen Anforderungen genügen müssen. Dann macht es Sinn, die Konstruktion sorgfältig zu parametrisieren so daß nur noch eine kleine Menge von Parametern frei änderbar ist, aus denen die übrigen Modellparameter dann berechnet werden. Überraschend viele komplizierte Dinge können mit erstaunlich wenigen Parametern beschrieben werden, darunter Brücken, Gebäude und Fassaden, Straßen, Rohrsysteme, Autofelgen, Möbel und gotische Masswerkfenster.

Schnelles Instanzieren von 3D-Modellen. Der Variantenreichtum selbst sehr großer Modellsammlungen (z.B. Google Warehouse) stößt schnell an die Grenzen, wenn man sehr viele ähnliche Modelle benötigt. Dazu sind parametrische Modelle ideal. Wir arbeiten zur Zeit an einem GML-Plugin für Maya, mit dem man sehr schnell unterschiedliche Modell-Instanzen erzeugen kann, die dann ganz normal mit Maya weiterbearbeitet werden können.

Schrittweises Verändern von 3D-Modellen. Um 3D-Modelle zu optimieren verfährt man schrittweise, indem man eine Simulation der Anforderungen durchführt, das Modell ein wenig verändert, und wieder simuliert usf. Will man diesen Prozess automatisieren, ist es offensichtlich von Vorteil, wenn man Modell-„Populationen“ systematisch erzeugen lassen kann, statt zur Konstruktionsänderung jedesmal wieder das CAD-Zeichenbrett zu bemühen.

Web-basierte 3D-Anwendungen (Client-Server). Schnelles „Formulieren“ von Anfragemodellen für die 3D-Suche und die web-basierte Produktkonfiguration sind zwei Projekte, an denen wir zur Zeit arbeiten. Dies basiert auf unserem GML-Plugin für Internet-Browser (IE/Firefox) und dem grossen Vorteil prozeduraler Modelle, sehr kompakt und damit schnell und einfach versendbar zu sein. Wir können dadurch zum Beispiel 3D-Modelle während sie editiert werden auf Client und Server synchron halten.

Austauschformat für „intelligente“ 3D-Modelle. Ein hochkomplexes, aber wissenschaftlich wie praktisch höchst spannendes Thema ist der Austausch parametrischer Modelle zwischen verschiedenen 3D-Softwarepaketen. Dies ist zur Zeit praktisch nicht möglich: Ein parametrisch veränderbares Getriebemodell, das in CATIA erstellt wurde, kann in Maya nur mehr als statische Geometrie importiert werden. GML wurde aber als erweiterbare Verallgemeinerung bestehender 3D-Modellbeschreibungs-Formate ursprünglich genau zu diesem Zweck entworfen. Selbst wenn CATIA-spezifische Modellier-Operationen, die dort als GML-Erweiterungen verfügbar sind, in Maya nicht vorhanden sein mögen, so kann vermutlich ein Großteil der Parametrik, die auf Standard-Operationen beruht, herüberge- rettet werden. – Dies sehen wir als ein sehr wichtiges langfristiges Ziel für GML an.

3 Unter GML lassen sich weitere, thematisch ähnliche Erklärungen finden. Wird es bei der gewählten Abkürzung bzw. Benennung bleiben?

Ja. Der Name „Generative Modellersprache“ bringt es einfach auf den Punkt. Ausserdem hat GML sich schon einen gewissen Namen gemacht. Die einzige thematische Überschneidung gibt es zur „Geography Markup Language“, wir sprechen zur Unterscheidung von Geo-GML und Shape-GML. Die „Graph Modeling Language“ scheint im Grafik-Bereich nicht so relevant zu sein, die „Game Modeling Language“ ist ebenfalls in einer anderen Nische zuhause, so dass es insgesamt wenige Missverständnisse gibt. Wir wissen meist, worüber wir reden.

4 Gibt es ausführliche Informationen/Tutorials in deutsch?

Nein, fast überhaupt nicht. Das ist bisher auch nicht geplant. Alles ist eigentlich konsequent auf Englisch. Da haben wir die Dokumente auf der GML-Homepage und das GML-Wiki aufgesetzt. Die Dokumentation ist aber mit Sicherheit noch recht verbesserungswürdig, unsere Kapazitäten sind da leider beschränkt.

5 Gestatten Sie eine prinzipielle Erwähnung von GML in Büchern?

Sicher, gerne! Wir sind an der Verbreitung interessiert und kooperieren auch gerne mit Forschung und Industrie, wenn es um spezielle Fragen geht. In gewisser Weise sind wir mit GML ja in einer Situation *Bolution looking for problem*".

6 Können Sie kurz in Stichpunkten den Ansatz schildern, der GML den Vorteil gegenüber klassischer Modellierung verschafft?

Generative Modellierung bedeutet, dass Listen von Objekten durch Sequenzen von Operationen ersetzt werden. In GML lassen sich so sehr einfach Verarbeitungsketten beschreiben, die flexibel auf Objekte und Objekt-Teile angewendet werden können. Dadurch gewinnt man automatisch eine Konstruktionsbeschreibung, die sich parametrisieren läßt.

Erzeugen von GML-Code im Hintergrund. GML ist als – sehr einfache – Programmiersprache mächtiger als reine Datenbeschreibungsformate. GML wurde aber, anders als z.B. JavaScript, weniger als eine benutzerfreundliche Programmiersprache entworfen. Stattdessen kann GML-Code sehr einfach automatisch erzeugt, ungeordnet, übertragen und verarbeitet werden. An Methoden zur „unsichtbaren“ GML-Erzeugung forschen wir derzeit recht aktiv.

Schnelle Freiformflächen. GML enthält viele Operatoren zur konventionellen Polyedermodellierung (bis auf Halbkanten-Niveau). Die gleichen Polyeder können durch Umschalten einzelner Kanten-Flags aber auch als Kontrollnetze für Freiformflächen dienen (Catmull-Clark Subdivision Surfaces). Die Modell-Auswertung wurde sehr stark optimiert, sodaß Modelle mit einigen tausend Basisflächen mehrfach in der Sekunde neu aufgebaut werden können.