Prof. Dr. Wolfgang Taube

Master-Thesis von Dominik Steffen: Analyse des Game Authoring Tool Entwicklungsprozesses und Konzeption eines Authoring Tool Frameworks. Juni 2015

Tools (Werkzeuge) unterstützen Arbeitsprozesse, die zur Erstellung eines Produktes oder einer Dienstleistung erforderlich sind. In diesen Arbeitsprozessen werden Objekte bearbeitet, die unterschiedliche Eigenschaften haben und unterschiedliche Bearbeitungsfunktionen benötigen. Unterschiedliche Entwicklergruppen arbeiten mit unterschiedlichen Objekten. Welche Objekte werden von welcher Benutzergruppen wie bearbeitet und wie hängen diese Objekte untereinander zusammen? Und wie entwickeln sich die Abhängigkeiten im Zeitverlauf – also bei Änderungen von Objekten?

Ganz am Ende der Arbeit erwähnen sie "das Konzept der nach Entwicklergruppen getrennten Authoring Tools…." (S. 95) – dieses Konzept wird aber in der Entwicklung der Thesis nicht deutlich. Sie beginnen mit einer ausführlichen Beschreibung der Stakeholder (Engineer, Artist, Game-Designer, Producer – S. 36). Die funktionalen Anforderungen bleiben aber auf einer recht allgemeinen Ebenen (X erstellen, bearbeiten, speichern). Die Frage ist: wie wirken sich denn die unterschiedlichen Sichtweisen der verschiedenen Gruppen auf das konkrete Design des FuseeAT aus? Bei der Darstellung des FuseeAT bleiben dann auf einmal nur noch der Engineer und der Artist übrig (S. 64) und die Auswirkungen auf die Entwurfsentscheidungen der FuseeAT werden nicht recht deutlich.

Aus meiner Erfahrung sind beispielsweise die Versionierung sowohl von Kode als auch von Assets und die verlustfreie bzw. verlustbehaftete Umwandlung einer Repräsentation in eine andere Repräsentation zentrale Probleme bei der Entwicklung komplexer interaktiver Medien-Projekte. Ist das auch bei der FuseeAT der Fall und wie werden die Probleme gelöst? Bei der Versionierung von Assets verweisen Sie auf Git (S. 71) – aber das bedeutet ja wieder eine völlig andere Oberfläche mit anderen, gar nicht so einfach zu verstehenden Konzepten.

Erstaunlich ist es, dass in der ganzen Arbeit kein einziger Screenshot aus dem Zielsystem verwendet wird. Die Unterstützung von Arbeitsprozessen macht sich doch sicherlich auch in Screen-Designs für die unterschiedlichen beteiligten Entwicklergruppen bemerkbar.

Zum Fusee Szenegraph schreiben sie: "Eine Node ist ein Knotenpunkt ohne spezielle Daten" (S. 83) Bei den C4D-Nodes sind "bereits Nutzdaten gespeichert" (S. 85). Weiterhin schreiben Sie: "Jegliche zu konvertierende Objekte einer Cinema 4D Szene können auf Fusee Seite zuerst einmal durch eine neue Node repräsentiert werden." – Die Frage ist: wo bleiben da die zusätzlichen C4D-Daten? Vermutlich werden sie als Components gespeichert, oder? Die eigentliche Frage ist aber: was ist denn der große Vorteil vom Fusee-Szenegraphen? Es muss ja einen Sinn haben, die Nodes nur als Container zu fassen.

Auf S. 54 machen Sie dann eine recht starke Aussage:

"Der Szenegraph der UE4 gleicht dem Fusee und Unity 3D Szenengraphen stark. …. und darum kann das System von Cinema 4D übernommen werden ohne voraussichtlich dem Arbeitsfluß der Designer und Artists zu schaden." - Das ist ja die zentrale Fragestellung der Arbeit, die sie hier einfach als gesetzt voraussetzen und dies bedarf dann doch wohl einer Erläuterung.