Automatisches Auflösen von Selbstschneidungen in zweidimensionalen Polygonen für ein CSG System

(Diplom-)Hausarbeit

an der

Verwaltungs- und Wirtschafts-Akademie in Leer e.V.

von

Heiko Alexander Weber

Leer, Groninger Str.

Betreuung:

Prof. Dr. Thorsten Spitta

2017

>

Inhalt

[1 Überschrift1 3](#_Toc491877620)

[1.1 Überschrift2 3](#_Toc491877621)

[1.1.1 Überschrift3 3](#_Toc491877622)

[Literatur <Typ Überschrift1; mit Löschtaste\_links Nummer entfernt> 4](#_Toc491877623)

# Abstrakt

Diese Arbeit findet im Rahmen des Studiengangs 16 and der VWA-Leer unter Betreuung von Prof. Dr. Spitta (Uni-Bielefeld) als Hausarbeit statt. Hierbei handelt es sich um eine Vorstudie zu dem „Automatisches Auflösen von Selbstschneidungen in zweidimensionalen Polygonen für ein CSG System“ .

# Einleitung und Zieldefinition

Ein bekanntes Problem bei sogenannten „CSG Systemen“, was für „Constructive Solid Geometry“ steht, ist die Verwendung eines sich selbst schneidenden planaren Polygonen (im *R2*Raum) für weitere Berechnungen oder kinematische Operationen (wie Extrusionen). Diesem Problem ist bereits eine Menge an Literatur gewidmen. In diesem Kontext stellen die genannten Polygone die originären Daten dar.

Diese Arbeit hat die Entwicklung eines Algorithmus bei dem die Integrität dieser originären Daten, unter dem Aspekt der Selbstschneidung, sichergestellt bzw. wieder hergestellt wird zum Ziel.

# Betriebswirtschaftlicher Aspekt

Die ORGADATA AG hat in der Vergangenheit bereits Vorkommnisse des in „Einleitung und Zieldefinition“ genannten Problems verzeichnet. Eine Lösung dieses Problems würde an der Stelle der Verwendung der originären Daten, zum Erzeugen abgeleiteter Daten (Gitternetzgeometrien im *R3* Raum) einen zu hohen Aufwand erfordern. Zusätzlich wäre dies nur eine Korrektur eines Symptoms und somit eine Verlagerung bzw. Verschleierung des Problems. Da dieses System, wie die meisten anderen, nach dem „*GiGo*[[1]](#footnote-1)“ Prinzip arbeitet, führt eine Injizierung falscher Eingabeparameter zu einem falschen Ergebnis, im schlimmsten Fall sogar zu einem nicht definierten Ergebnis.

Momentan wird dieses Problem von der Abteilung, welche die entsprechenden originären Daten erfasst, gelöst in dem die Zeichnungen manuell auf Integrität überprüft und ggf. korrigiert werden. Dies ist aufgrund der massiven Datenmenge (mehrere zig tausend Zeichnungen) und der Tatsache, dass diese Überprüfung und Korrektur nur im bereits aufgetretenen Fehlerfall erfolgt, nicht sinnvoll. Ein weiterer Aspekt hierbei sind die Kosten, die das manuelle überprüfen bzw. korrigieren verursacht. Es sei folgende Beispielrechnung gegeben:

Ein Mitarbeiter *e*, verantwortlich für die Erfassung originärer Daten, überprüft und korrigiert eine Menge an Zeichnungen *d* pro Monat, bei welchen durchschnittlich ein Zeitaufwand *h* pro Einheit notwendig ist. Genannter Mitarbeiter verursacht der ORGADATA AG monatlich Kosten *k*, bei einer Gesamtarbeitszeit (*H*) pro Monat.

Folgende Kosten fallen pro Mitarbeiter monatlich zur Bearbeitung dieser Sache an:

Unter der Annahme *beispielhaft angelehnter Werte*[[2]](#footnote-2) würden somit die vorhergehende Formel folgendermaßen materialisiert:

Es wurden hierbei verursachte Gesamtkosten pro Monat in Höhe von 2000€, 160 Stunden Arbeit pro Monat, 20 Zeichnungen pro Monat sowie ein Zeitaufwand von 0.5 Stunden pro Zeichnung veranschlagt.

Es sei gesagt, dass die *kaum schätzbaren Kosten, welche aus den auftretenden Fehlerfällen resultieren*[[3]](#footnote-3) in dieser Rechnung noch nicht berücksichtigt sind. Diese sind, grob geschätzt, weitaus höher.

Dieses Projekt wurde, unter dem Aspekt der wirtschaftlichkeit, mit der Intention gewählt, die genannten Kosten zu minimieren bzw. zu eliminieren.

# Literatur <Typ Überschrift1; mit Löschtaste\_links Nummer entfernt>

**[Codd82]** Codd, E. F.: Relational Database: A Practical Foundation for Productivity. Communications of the ACM 25(1982) 2, 109-117.

**[ElNa09]** Elmasri, R.; Navathe, S.B.: Grundlagen von Datenbanksystemen – Ausgabe Grundstudium. 3.Aufl. Pearson, München – Boston et al. 2009 (Übers. US-Ausgabe 2000).

**[SpBi08]** Spitta, T.; Bick, M.: Informationswirtschaft – Eine Einführung. 2. Aufl., Springer, Berlin – Heidelberg – NewYork et al 2008.

oder:

Codd, E. F. (1982): Relational Database: A Practical Foundation for Productivity. Communications of the ACM 25(2), 109-117.

Elmasri, R.; Navathe, S.B. (2009): Grundlagen von Datenbanksystemen – Ausgabe Grundstudium. 3.Aufl. Pearson, München – Boston et al. (Übers. US-Ausgabe 2000).

Spitta, T.; Bick, M. (2008): Informationswirtschaft – Eine Einführung. 2. Aufl., Springer, Berlin – Heidelberg – NewYork et al.

Wenn Sie die Dokumentvorlage abspeichern, wird hinter 'Stand' das Tagesdatum   
und in der Mitte der Fußzeile der Dateiname eingefügt. Das ist für die Erstellung machmal praktisch.

Vor der Abgabe können Sie die Fußzeile einfach löschen, wenn Sie das wollen.

Die erste Seite hat in OO den **Seitentyp** *Erste Seite*, alle weiteren den Typ *Standard*.

In Word gibt es m.W. das Format „Seitentyp“ nicht.

1. „GiGo“ steht für „Garbage in, Garbage out“ und bezeichnet eine Systemart bei der die Eingabe falscher Parameter zur Ausgabe eines falschen bzw. nicht definierten Ergebnis führt. [↑](#footnote-ref-1)
2. „Beispielhafte angelehnte Werte“, da die Offenlegung von Gehältern o.Ä. gegen die Geheimhaltungsvorschriften vertraulicher Daten verstöße. [↑](#footnote-ref-2)
3. Damit sind die durch genannte Fehlerfälle (falsche Ansichten oder im schlimmsten Fall Abstürze) auftretenden Kosten beim Kunden gemeint, welche indirekt der ORGADATA AG entstehen. Dazu gehört auch der Aufwand des Debuggens bzw. der Kategorisierung gemeldeter Fehler. [↑](#footnote-ref-3)