

# Erweiterungsmöglichkeiten von MS Word

Die folgende Liste stellt eine Übersicht über die Möglichkeiten zur Einflussnahme auf das Verhalten von Microsoft Word dar. Der Schwerpunkt liegt hierbei auf MS Word 2000, da diese Version vorliegt. In der Dokumentation finden sich zwar einige Hinweise auf neue Funktionen und Möglichkeiten, die in vorherigen nicht oder in beschränktem Umfang vorhanden waren. Daher sind entsprechende Annahmen bzgl. vorheriger Versionen unbedingt zu prüfen.

## Erweiterungsmöglichkeiten

### Makroaufzeichnungen

Der Aufgabenbereich von (Aufzeichnungs-) Makros sind einfache Befehlsfolgen, beispielsweise zur Formatierung. Diese aufgezeichneten Makros können dann durch den Benutzer bearbeitet werden - dies geschieht dann im Visual Basic Editor. Wird ein reines Aufzeichnungsmakro betrachtet, so ist klar, dass diese Methode nicht annähernd genug Ausdruckskraft besitzt, als dass sie weiterhin in Betracht gezogen werden könnte.

### Webskripte

Die Skriptsprache ähnelt den Makros, ist allerdings gedacht zum Einsatz in Worddokumenten, die als Website veröffentlicht werden. Diese Variante kann also evtl. zu einem späteren Zeitpunkt betrachtet werden, wenn eine Vorabfassung des Werks als Onlinefassung angezeigt werden soll.

### ActiveX - Controls

Sind nicht direkt in Word integrierbar. Laut Dokumentation sind in VBA 6 zusätzliche ActiveX-Komponenten integrierbar. ActiveX ist ein sprachunabhängiges Format, mit dem Komponenten verteilt werden können und basiert auf COM. ActiveX-Controls können in Web-Pages beispielsweise für Multimedia Effekte und interaktive Objekte eingebunden werden.

### Makros allgemein bzw. Visual Basic Module (VBA-Module)

Diese sog. VBA-Makros werden in dem von MS Word mitgelieferten Visual Basic Editor erstellt. Diese Methode zur Erweiterung von MS Word, aber auch allen anderen Office-Komponenten, ist der offensichtliche Favorit der Entwickler. Es ist aber zu klären, ob die Bezeichnung 'Visual Basic für Applikationen' tatsächlich etwas eigenständiges ist. Die Zugriffsmöglichkeiten auf Office-Komponenten sind sehr weitreichend und gut dokumentiert. Eigene Beispielprogramme waren ohne erheblichen Aufwand innerhalb kürzester Zeit erstellt. Die Abbildung 1 veranschaulicht, dass VBA-Module Elemente eines Dokuments oder Formulars sind. Hieraus ergeben sich eventuell Restriktionen bezüglich der Zugriffsmöglichkeiten auf den Dokumenteninhalte bzw. auf Ereignisse.

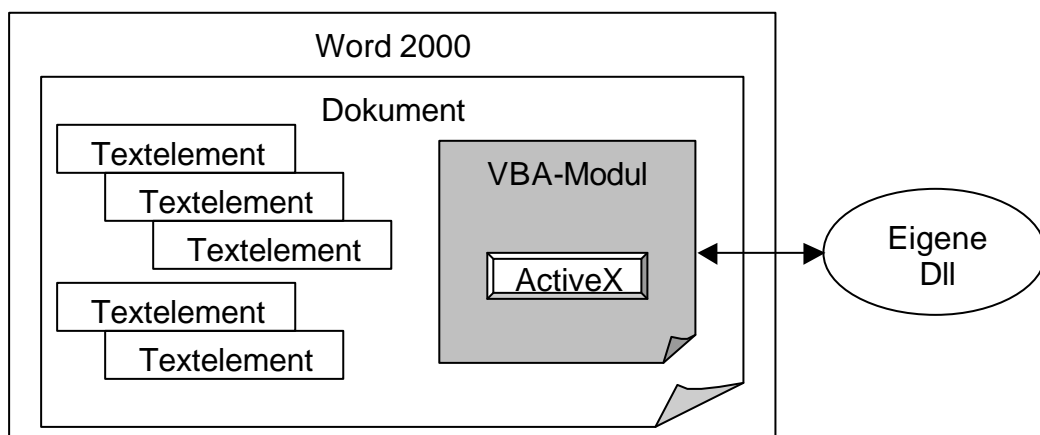


Abbildung 1

Beurteilung: Bietet viele Möglichkeiten, die auch recht umfangreich dokumentiert sind. Nachteil: An welche Ereignisse man anknüpfen kann und welche Aktivierungsmöglichkeiten bestehen,

muss noch geklärt werden. Eventuell ist diese Lösung nicht robust, da man eine Mixtur von Codierungstechniken und Philosophien einsetzt.

## COM-Add-Ins

Laut Dokumentation handelt es sich hierbei "...um Zusatzprogramme, die Office-Programme um benutzerfreundliche Befehle und spezialisierte Funktionen erweitert. COM-AddIns können in einer beliebigen Programmieranwendungen von Microsoft ...implementiert werden".

In Abbildung 2 zeigt die Positionierung: die AddIns beziehen sich direkt auf Word2000 und nicht auf einzelnen Dokument, Vorlagen oder Dokumentelemente.

Meine Versuche haben allerdings folgendes Bild gezeichnet: ein unter Visual C++ 6.0 erstelltes COM-AddIn liess sich überhaupt nicht einbinden, leider gab es bei diesem offensichtlich erfolglosen Versuch keine Fehlermeldung, so dass die Ursache im Dunkeln liegt. Ein unter Visual Basic 6.0 erstelltes AddIn ließ sich zwar einbinden, aber wie es dann aktiviert, oder gar benutzt wird, war nicht zu klären. Zu diesem Punkt habe ich bisher weder eine Anleitung noch ein Beispiel finden können.

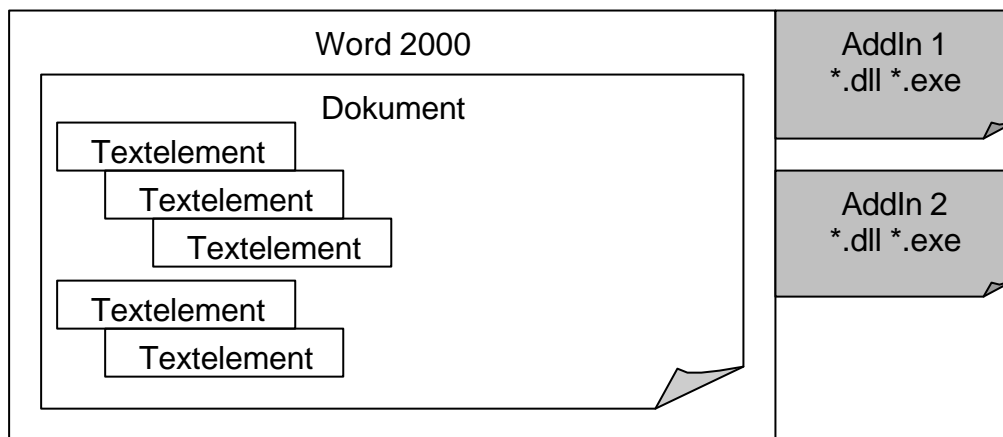


Abbildung 2

Beurteilung: Bietet den gesamten dokumentierten Sprachumfang (solange man Visual Basic für Applikationen einsetzt). Der sauberste Weg wäre hier wohl, ein Add-In zu schreiben, das einerseits die Interaktion mit dem Autor übernimmt und die anderen (relevanten) Änderungen weiterleitet. Hier liegt der Vorteil in einer saubereren Architektur und in der freien Wahl der Programmiersprache. Allerdings sind bisherige Tests nicht erfolgreich.

## OLE Automatisierung

Eine weitere Möglichkeit, die allerdings nirgendwo dokumentiert ist, ist das Einbinden der von Word bereitgestellten Typbibliothek in Visual C++ 6.0 mittels des Class-Wizards. In meinem Beispielprogramm (Container-Dokument) ist es mir bisher gelungen, ein Word-Dokument samt Menüs und Symbolleisten anzuzeigen, aber ich konnte nicht auf Elemente des Objektes zugreifen.

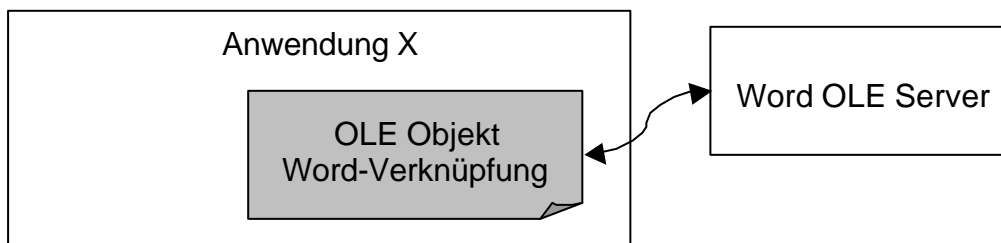


Abbildung 3

Beurteilung: Ist nicht dokumentiert. Zwar hat man die Typbibliothek, aber wo hier die Tücken liegen, ist nicht abzuschätzen. Diese Methode ausserdem zur Folge, dass man nicht Word startet, sondern eine andere Applikation. Dieses könnte zu Akzeptanzproblemen führen.