

Diplomarbeit

Entwicklung und Test einer interaktiven Schnittstelle für
den Einsatz der digitalen Unterschrift im Entwicklungs-
und Fertigungsprozess eines modernen
Luftfahrtunternehmens

von Joachim Greiner

durchgeführt am
Institut für Flugzeugbau
Universität Stuttgart

betreut von Dipl.-Ing. Peter Schnauffer

Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
Abbildungsverzeichnis	3
Abkürzungen	5
Einleitung	6
1 SmarTeam als PDM	7
1.1 Darstellung in SmarTeam.....	8
1.2 Stadien im Lebenszyklus eines Dokuments	11
1.3 Handschrift Erkennungssystem HESY	13
1.3.1 Funktionen von HESY	14
1.4 Umsetzung am IFB.....	15
2 Umsetzung in SmarTeam	16
2.1 Benutzerdefinierte Skripte in SmarTeam.....	17
2.2 Das Dialogfeld zum Unterschreiben	19
2.2.1 Allgemeine Funktionen	20
2.2.2 Checkin / Release Regeln.....	20
2.2.3 Unterschriftenerkennung.....	21
2.2.4 Checkin ohne Unterschrift	22
2.3 Unterschriften in Word-Dokumenten.....	22
2.4 Unterschriften in Excel.....	24
2.5 Unterschriften in CATIA	26
2.6 Freigabe von Dateien ohne Unterschrift	28
3 Funktion des PDMs anhand eines Beispiels	30
3.1 Sichern und einspielen von komplexen Strukturen.....	30
3.2 Release von unterschriebenen Dokumenten	37
3.3 Freigeben von Dokumenten, die nicht unterschrieben werden können	39
3.4 Die Funktionen „New Release“ und „Obsolete“.....	41
Zusammenfassung und Ausblick	42
Literaturverzeichnis.....	43

Abbildungsverzeichnis

Abbildung 1.1 schematische Darstellung eines PDM Systems nach [1]	7
Abbildung 1.2 SmarTeam Ansicht der einzelnen Projekte. Auf der linken Seite die Baumstruktur der Projekte mit den jeweiligen Unterprojekten, rechts die zusätzlichen Informationen zum ausgewählten Projekt.....	9
Abbildung 1.3 Ansicht der Dokumente und Ordner auf ein bestimmtes Projekt bezogen. Gleiche Aufteilung wie bei der Projektansicht.	10
Abbildung 1.4 gleiches Dokument mit verschiedenen Revisionsnummern	11
Abbildung 1.5 Lebenszyklus eines Dokuments im PDM System	12
Abbildung 1.6 HESY Unterschriften-Pad [7]	13
Abbildung 1.7 Unterschrift aufgenommen von HESY	
Abbildung 1.8 Unterschrift um 90° gedreht	13
Abbildung 1.9 schematische Darstellung der Funktionen von HESY	14
Abbildung 1.10 schematische Darstellung der Vorgehensweise am IFB.	15
Abbildung 2.1 SmarTeam Menü in Word.....	16
Abbildung 2.2 SmarTeam Menü in CATIA.....	16
Abbildung 2.3 Darstellung der Skriptausführung bei einem bestimmten SmarTeam Ereignis. (Hier: Klick auf „Checkin“)	17
Abbildung 2.4 SmarTeam Skript Maintenance: Das bei einem bestimmten Ereignis auszuführende Skript wird hier in die entsprechende Spalte einfach eingetragen.....	18
Abbildung 2.5 Benutzeroberfläche des Unterschriften-Programms	19
Abbildung 2.6 schematische Darstellung des Unterschriftenvergleichs.....	21
Abbildung 2.7 schematische Darstellung des Wegs eines Word- bzw. Excel-Dokuments von der ersten Unterschrift bis zur Freigabe (release).	23
Abbildung 2.8 Worddokument mit drei Unterschriften, sowie den dazugehörigen Dateien, welche unter dem Dokument angehängt sind.	24
Abbildung 2.9 komplett unterschriebenes Excel Dokument. Die Dateien mit den Daten der Unterschrift werden am Ende eingefügt.....	25
Abbildung 2.10 schematische Darstellung des Weges einer CATIA-Zeichnung von der ersten Unterschrift bis zur Freigabe.....	26
Abbildung 2.11 Dateien mit den Daten der einzelnen Unterschriften. Bei CATIA Zeichnungen werden die Dateien nicht angehängt, sondern direkt in SmarTeam eingecheckt.....	27
Abbildung 2.12 DIN Normfeld einer CATIA Zeichnung mit erfolgten Unterschriften in den Feldern „bearbeitet“ und „geprüft“	28
Abbildung 2.13 Benutzeroberfläche für den Release Vorgang. Hier besteht nur noch die Möglichkeit, die Authentifizierung der Unterschrift abzuschalten. Ein Programm muss hier nicht ausgewählt werden.....	29
Abbildung 3.1 sichern von Dateien oder komplexeren Strukturen im Menü „SmarTeam“ in CATIA.....	30
Abbildung 3.2 beispielhafte Baumstruktur in CATIA	31
Abbildung 3.3 SmarTeam-Fenster, das nach dem klicken auf „sichern“ erscheint. Hier besteht die Möglichkeit den Ordner des zu sichernden Dokuments/Produkts zu wählen.....	32
Abbildung 3.4 Vorgehensweise bei verschiedenen CATIA-Formaten.....	33
Abbildung 3.5 durch Klicken auf „Einchecken“ im Untermenü „Lebenszyklus“ wird der Vorgang gestartet	34
Abbildung 3.6 Die kompletten Daten des gesamten Flugzeugs werden ohne Unterschrift ins SmarTeam eingecheckt.	35
Abbildung 3.7 Baumstruktur der einzelnen Dokumente des ausgewählten Projekts.	36

Abbildung 3.8 Einfügen der Unterschrift in eine CATIA Zeichnung.....	37
Abbildung 3.9 freigegebene CATIA Zeichnung mit den drei Unterschriften, als Dateien im gleichen Ordner wie die Zeichnung abgelegt.....	38
Abbildung 3.10 nach dem Klicken auf „freigeben“ erscheint die Benutzeroberfläche zur Eingabe der Unterschrift für die Freigabe.....	39
Abbildung 3.11 SmarTeam Baumstruktur mit freigegebenen Objekten und den dazugehörigen Unterschriften.....	40

Abkürzungen

CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Modeling
dll	Dynamik Link Library
HESY	Handschriften Erkennungssystem
IFB	Institut für Flugzeugbau
PDM	Produkt Daten Management

Einleitung

In einem modernen Unternehmen der Luftfahrtindustrie ist man heute darum bemüht, wie in jedem anderen Unternehmen natürlich auch, die Kosten in Produktion und Entwicklung so gering wie möglich zu halten. Vor Allem lange Entwicklungszeiten kosten viel Geld.

Die vorliegende Arbeit beschäftigt sich damit, die Zeiten von der Erstellung eines Bauteils am Computer bis zu dessen Produktion zu verringern. Auch soll dabei versucht werden, den Verbrauch von bedrucktem Papier zu senken. Dies geschieht durch die Einführung der elektronischen Unterschrift.

In einer so sicherheitsrelevanten Industrie wie der Luftfahrt, werden die entworfenen Bauteile von mehreren Leuten geprüft, bis das Bauteil gefertigt wird. Die bestandene Prüfung des Bauteils wird in der Regel durch Unterschrift der dazu befugten Personen gegeben. Dies kann unter Umständen einige Zeit dauern, denn das zu unterschreibende Dokument muss dann von einem Unterschreibenden zum anderen gebracht werden, bis derjenige zum Unterschreiben kommt, kann wiederum Zeit vergehen. Dann wird das Dokument zum Nächsten gebracht, usw.

Nach einem Gesetz des Deutschen Bundestages sind elektronische Unterschriften genauso rechtsverbindlich wie eine, auf einem Papier geleistete Unterschrift.

Ein von Baltus [7] entwickeltes Gerät, HESY genannt, ist in der Lage Unterschriften elektronisch aufzuzeichnen und zu verarbeiten. Das Ziel dieser Arbeit ist nun, HESY mit einem Anwendungsprogramm und einem Produktdatenmanagement-System (PDM) zu kombinieren. Das Ziel soll sein, Dokumente und Zeichnungen direkt nach dem Erstellen zu unterschreiben und sie dann über ein PDM den Mitarbeitern zur Verfügung zu stellen, die dann ggf. auch noch unterschreiben können. Damit stehen die Dokumente allen Mitarbeitern auf einmal zur Verfügung und müssen auch nicht mehr ausgedruckt werden.

Als PDM System wurde SmarTeam V4 R14 von Dassault Systèmes verwendet. HESY wirkt hierbei als Bindeglied zwischen SmarTeam und dem CAD-Programm CATIA in der Version V5 R14 und R15 von Dassault Systèmes sowie den Programmen Word und Excel von Microsoft.

Als Betriebssystem wird Windows 2000 oder XP benötigt.

1 SmarTeam als PDM

In einem Unternehmen der heutigen Zeit, fallen eine immer größere Anzahl an Informationen, Normen und Gesetzen an. Diese Informationen liegen im Allgemeinen in elektronischer Form vor. Ziel ist es, diese Flut an Informationen zu bündeln und zielgerichtet zur Verfügung zu stellen, so dass jeder Mitarbeiter auch die und nur die Informationen bekommt, die er benötigt.

Mit Hilfe eines Produktdatenmanagement Systems (PDM) können Informationen auf eine unkomplizierte und komfortable Art und Weise aufbereitet und zur Verfügung gestellt werden. Dabei darf das PDM nicht als einzelnes Programm gesehen werden, sondern muss eingebettet in die gesamte IT Strategie des Unternehmens betrachtet werden. Dies gilt speziell für den Bereich des Engineerings. Hier ist das PDM mit CAD-Systemen (Computer Aided Design), CAE-Systemen (Computer Aided Engineering) sowie CAM-Systemen (Computer Aided Modeling) verbunden. Auch sind Office Programmen wie z.B. Word oder Excel mit dem PDM kombiniert.

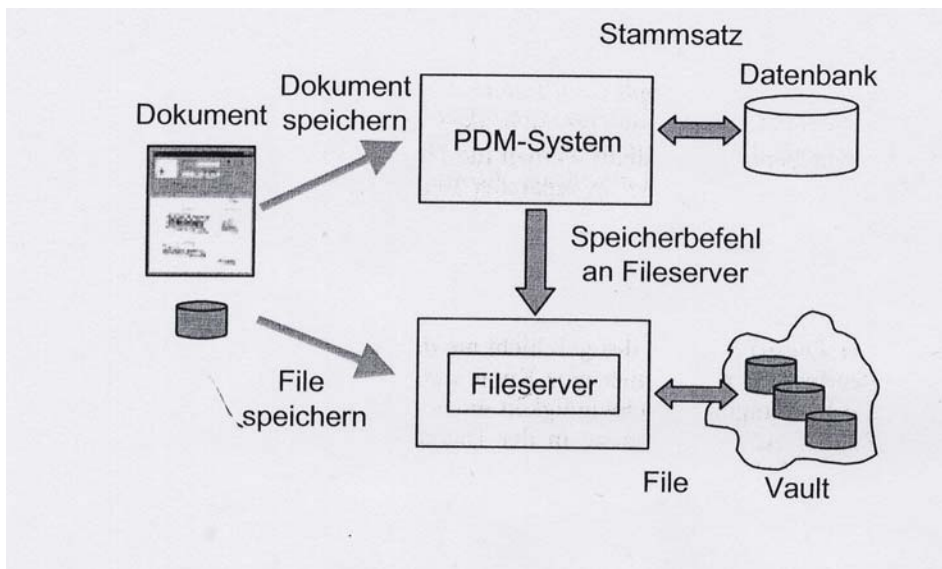


Abbildung 1.1 schematische Darstellung eines PDM Systems nach [1]

Vom Prinzip her ist das PDM vergleichbar mit einer Partition auf einem Server, auf die auch jeder Mitarbeiter zu greifen kann. Allerdings bietet das PDM deutlich mehr Funktionalität und Sicherheit. So kann man z.B. sehen, welcher Mitarbeiter gerade welches Dokument bearbeitet, oder wer es erstellt hat. Auch können einem Dokument mehrere verschiedene Stadien zugewiesen werden. Mehr dazu in Kapitel 1.2.

Ein PDM System besteht im Wesentlichen aus zwei großen Teilen. Der eine Teil ist eine Datenbank, in welcher alle Daten, die das Dokument betreffen gespeichert werden. Dazu gehören vor Allem der Dateiname, der Pfad der Datei, sowie noch Informationen zum Verfasser und zur Art des Dokumentes. Ebenfalls wichtig ist der Gesamtzusammenhang des Dokuments in einer bestehenden Struktur. Dies bedeutet, dass bestehende Verknüpfungen zu anderen Dateien ebenfalls berücksichtigt werden. Die Einträge in dieser Datenbank sind nur Verknüpfungen auf die eigentliche Datei. All diese Informationen sind im sogenannten Stammsatz zusammengefasst.

Der zweite Teil ist ein Fileserver, auf dem das Dokument dann abgespeichert wird. Der Fileserver hat einen abgesicherten Bereich, der mit einem elektronischen Tresor zu vergleichen ist. Dieser Bereich wird bei vielen Systemen „*Vault*“ (Tresor) genannt. In diesem Bereich sind die Dokumente sicher geschützt. In den *Vault* können nur Dokumente hineingelangen, aber keines mehr heraus. Das bedeutet, wenn einmal ein Dokument hineingestellt worden ist, bleibt dauerhaft im *Vault*. Wird das Dokument im PDM gelöscht, so wird einfach nur die Verknüpfung auf die Datei im *Vault* gelöscht, nicht aber die Datei selber. Dateien aus dem *Vault* löschen, kann nur ein Administrator der Datenbank, kein Administrator des PDMs.

Eine Unterteilung in diese zwei Bereiche ist aus mehreren Gründen sinnvoll:

Eine Datenbank, in welcher alle Informationen und zusätzlich noch die Dateien gespeichert sind, würde viel zu groß werden und wäre dadurch schlecht zu warten. Des Weiteren würde die Leistungsfähigkeit des Systems unter einer großen Datenbank leiden. Die Dokumente vom Fileserver abzurufen verkürzt die Zugriffszeiten erheblich. Sollte aus verschiedenen Gründen die Datenbank einmal ausfallen, so sind die Dokumente weiterhin noch zu erreichen. Viele Firmen legen hierauf besonderen Wert. Das ganze System ist in Abbildung 1.1 schematisch dargestellt.

1.1 Darstellung in SmarTeam

Die Verzeichnisstruktur von SmarTeam ist prinzipiell die selbe, wie die Verzeichnisstruktur in Windows. Das oberste Element in der Hierarchie ist hierbei das Projekt. Ebenso können bei einem Projekt auch noch Unterprojekte erstellt werden. Das Fenster ist in zwei Bereiche geteilt. Im linken Bereich befindet sich die Baumstruktur, im rechten Bereich werden die zum jeweils ausgewählten Projekt vorhandenen Informationen angezeigt. Der Informationsbereich ist in mehrere Karteikarten unterteilt, in denen dann die detaillierten Informationen abgerufen werden können. Abbildung 1.2 zeigt diese Ansicht in SmarTeam.

Von der Projektansicht kann dann einfach auf die zum jeweiligen Projekt gehörigen Dokumente und Ordner umgeschaltet werden. Die Ansicht hier ist identisch mit der, der Projektansicht. Auch hier befindet sich auf der linken Seite die Baumstruktur mit den Ordnern, und auf der rechten Seite die genaueren Informationen zum jeweils ausgewählten Dokument. Diese Ansicht ist in Abbildung 1.3 dargestellt.

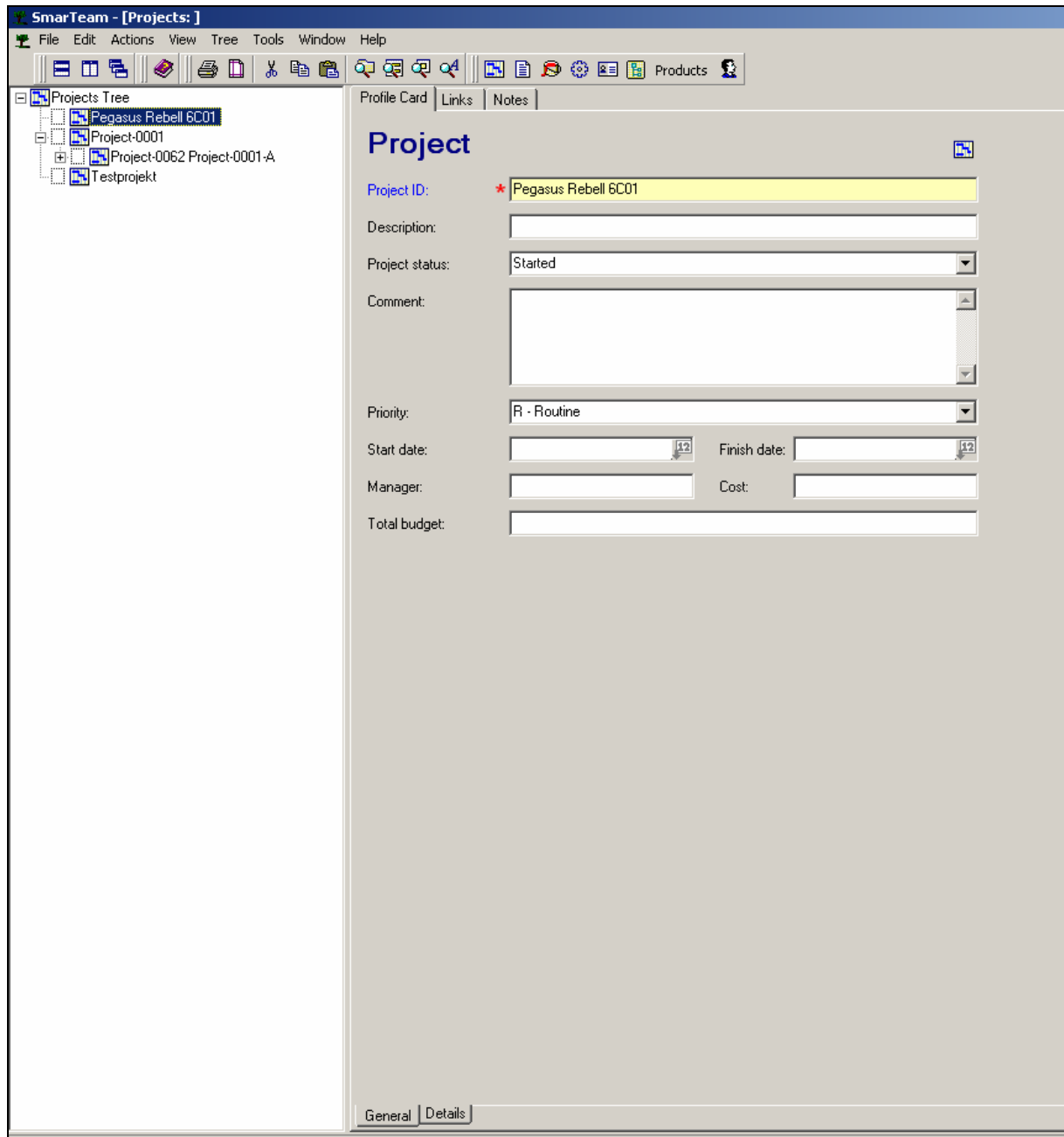


Abbildung 1.2 SmarTeam Ansicht der einzelnen Projekte. Auf der linken Seite die Baumstruktur der Projekte mit den jeweiligen Unterprojekten, rechts die zusätzlichen Informationen zum ausgewählten Projekt.

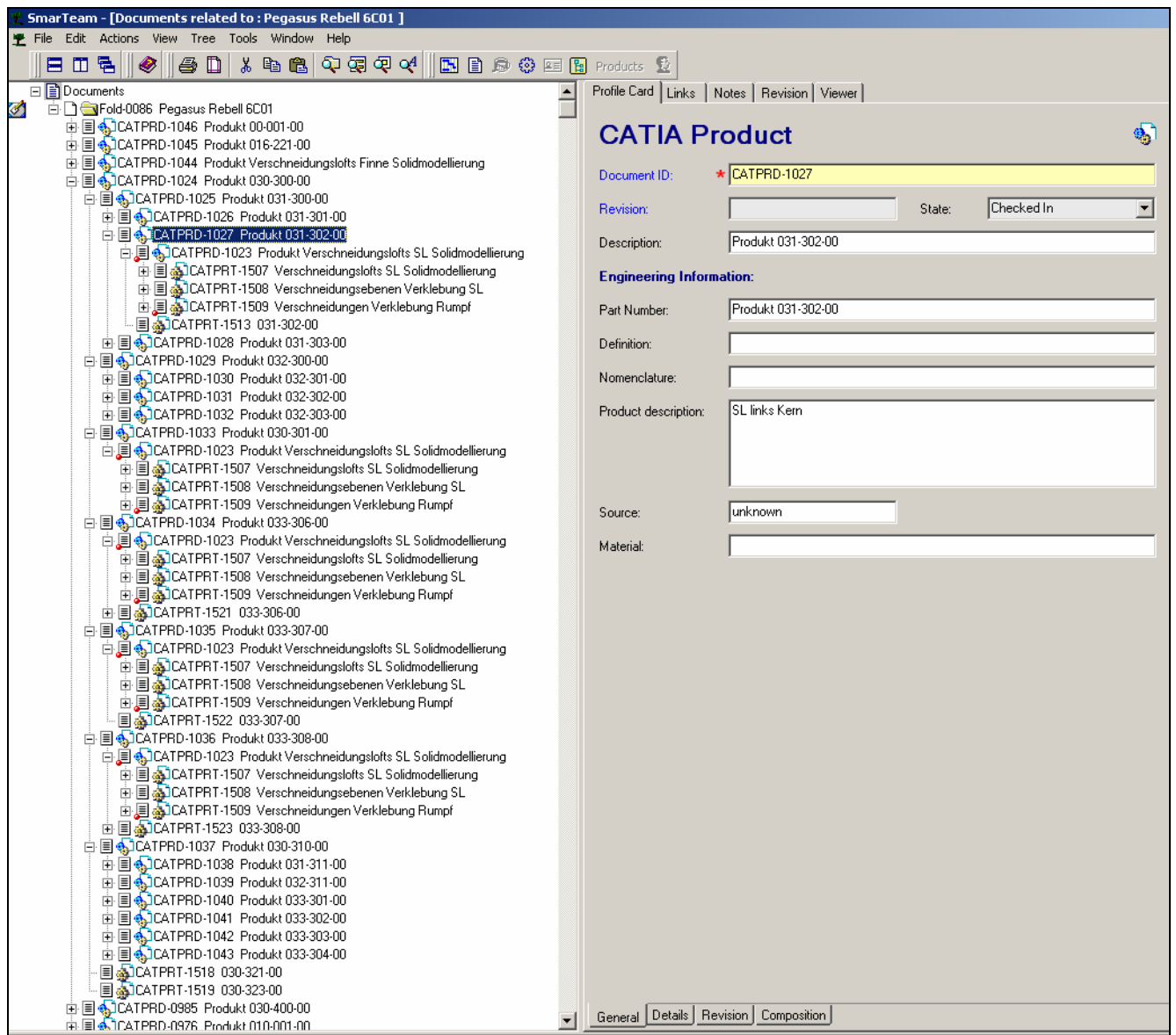


Abbildung 1.3 Ansicht der Dokumente und Ordner auf ein bestimmtes Projekt bezogen. Gleiche Aufteilung wie bei der Projektansicht.

1.2 Stadien im Lebenszyklus eines Dokuments

Ein Dokument durchläuft von seiner Einstellung in das PDM bis zum Zeitpunkt an dem es nicht mehr gebraucht wird, verschiedene Stadien. Dabei gibt es fünf verschiedene Stufen: new, checkin, checkout, release und obsolete.

Ein Dokument das im PDM angemeldet wird, bekommt den Status „new“. Hierbei wird nur ein Eintrag in die Datenbank vorgenommen und diese somit auf den späteren checkin Vorgang vorbereitet. Dabei wird auch dem Dokument eine Kennung zugewiesen, anhand derer es eindeutig identifiziert werden kann. Die Kennung setzt sich aus einem Kürzel, welches die Art des Dokuments beschreibt (DOC für Word Dokument, CATDRW für eine CATIA Zeichnung), und einer vierstelligen Nummer zusammen. Der Eintrag umfasst ebenfalls den Namen sowie den Pfad des Dokuments. Ebenfalls gespeichert werden Informationen, wie der Ersteller oder das Datum. Das Dokument befindet sich weiterhin auf einem lokalen Rechner.

Erst beim nächsten Schritt, dem checkin Vorgang, wird das Dokument auf den Fileserver verschoben und vom lokalen Rechner entfernt. Das Dokument ist nun im *Vault* gesichert und kann von allen Benutzern des PDMs genutzt werden. Sollten jetzt noch Änderungen am Dokument vorgenommen werden, so muss eine Kopie auf den Rechner des Benutzers gespielt werden (checkout). Das Original bleibt dabei im *Vault*. Wird das geänderte Dokument nun wieder eingespielt (checkin), so erhält es eine neue, so genannte Revisionsnummer. D.h. die Dokumente, die sich vom ursprünglichen Dokument unterscheiden, werden mit dem Zusatz a.0, a.1.... bezeichnet.

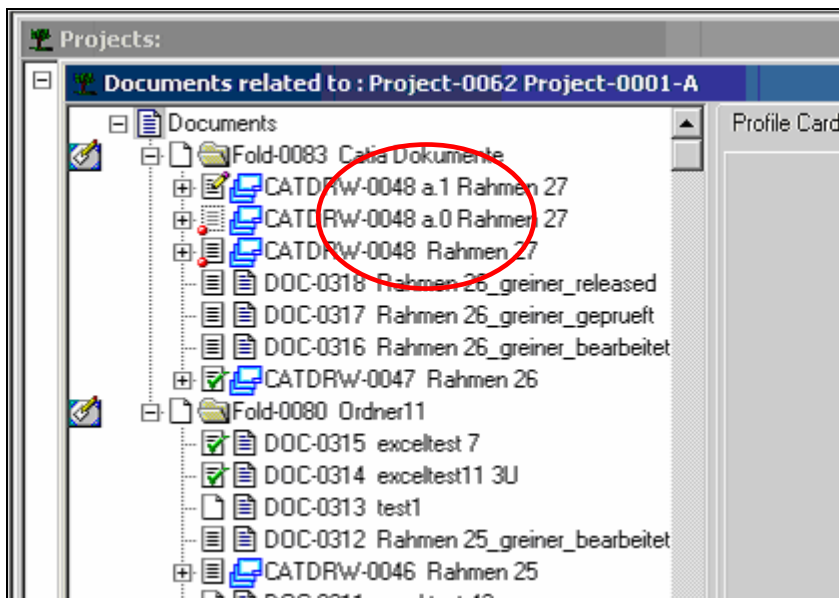


Abbildung 1.4 gleiches Dokument mit verschiedenen Revisionsnummern

Wie in Abbildung 1.4 zu sehen, bleibt die Kennung des Dokuments erhalten (CATDRW-0048). Jedoch wird bei jedem Checkout-Vorgang zwischen der Kennung und der Beschreibung des Dokuments eine neue Revisionsnummer eingefügt. Der rote Punkt am Dokumentensymbol bedeutet, dass es von diesem Dokument eine neuere Version mit einer höheren Revisionsnummer gibt. Ist auf dem Symbol noch ein Stift vermerkt, so bedeutet dies, dass dieses Dokument gerade ausgecheckt wurde. Das heißt, eine Kopie des Dokuments wurde auf den Rechner des jeweiligen Benutzers zur Bearbeitung gespielt.

In der nächsten Stufe (release) wird das Dokument dann für die Produktion freigegeben. In SmarTeam wird dies durch einen grünen Haken beim angezeigten Dokument verdeutlicht. Die letzte Stufe im Lebenszyklus eines Dokuments ist „obsolete“. Das bedeutet, das Dokument ist veraltet und wird für die Produktion nicht mehr benötigt. Der ganze Lebenszyklus ist in Abbildung 1.5 noch einmal schematisch dargestellt.

Lebenszyklus eines Dokuments

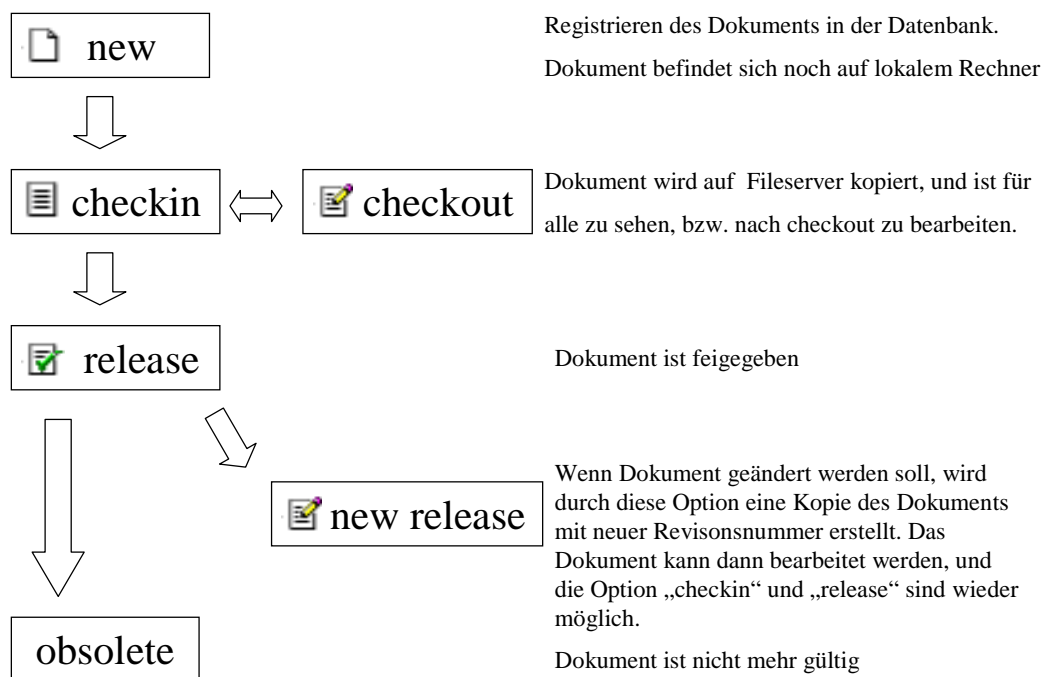


Abbildung 1.5 Lebenszyklus eines Dokuments im PDM System

1.3 Handschrift Erkennungssystem HESY

HESY ist eine Entwicklung von Baltus [7]. Das Gerät ähnelt einem Touchpad eines Laptops. Die Aufnahme der Unterschrift erfolgt über eine, an vier Dehnmessstreifen aufgehängten Platte. Durch die unterschiedliche Gewichtung der Platte geben die Messstreifen unterschiedliche Signale ab. Aufgrund dieser Werte können dann die Daten der Unterschrift aufgezeichnet werden.



Abbildung 1.6 HESY Unterschriften-Pad [7]

Das System ist nicht nur in der Lage eine Unterschrift zweidimensional aufzunehmen, wie dies auf einem Papier der Fall wäre. HESY zeichnet eine Unterschrift vierdimensional auf. Zu den üblichen zweidimensionalen Koordinaten wird zusätzlich die Drucktiefe registriert, so wie die Zeitdauer der Unterschrift. Diese Daten werden dann in einer Datei abgespeichert. Da es absolut unmöglich ist, zweimal exakt die selbe Unterschrift abzuliefern, unterscheiden sich auch die aufgezeichneten Daten einer Unterschrift geringfügig. Sollten zweimal die exakten Daten auftauchen, so handelt es sich hierbei um eine Fälschung. Dieses Verfahren ist besonders fälschungssicher, denn wenn das Schriftbild kopiert wird, fehlen die Daten der Unterschrift. Auch das „Nachfahren“ einer bestehenden Unterschrift würde sofort auffallen, da hier die Drucktiefe sowie die Geschwindigkeit nicht nachgemacht werden können. Somit ist die „elektronische Fälschung“ der Unterschrift noch schwieriger als eine Fälschung der „normalen“ Unterschrift. Beides erfüllt den Straftatbestand der Unterschriftenfälschung.

Die beiden nachfolgenden Bilder zeigen den dreidimensionalen Charakter, der von HESY aufgezeichneten Unterschrift. Abbildung 1.7 zeigt die Unterschrift von oben betrachtet, so wie sie auch auf einem Papier stehen würde. In Abbildung 1.8 ist die Unterschrift dann um 90° gedreht, also eine Ansicht von der Seite. Hierbei ist gut zu sehen, wie von HESY auch die Drucktiefen mit aufgezeichnet werden.

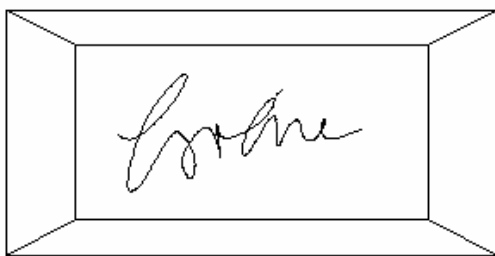


Abbildung 1.7 Unterschrift aufgenommen von HESY

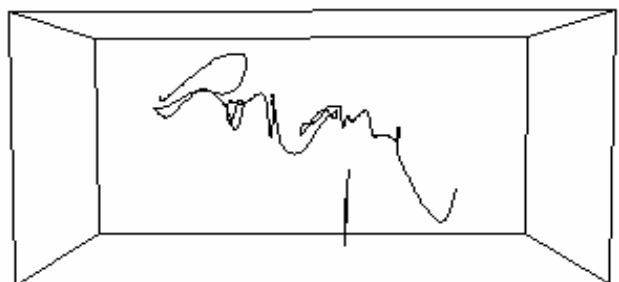


Abbildung 1.8 Unterschrift um 90° gedreht

1.3.1 Funktionen von HESY

HESY bietet neben den Treibern, noch einen erheblichen Funktionsumfang. Alle Funktionen liegen als Programmbibliotheken (Dynamik Link Library, dll) vor, und können so unabhängig von der jeweiligen Programmiersprache angesprochen und benutzt werden. Zum Umfang gehören Funktionen, die es ermöglichen, die Unterschrift einzulesen und als Bild darzustellen. Für die Darstellung als Bild, muss die verwendete Programmiersprache jedoch über ein Element verfügen, das in der Lage ist einzelne Bildpunkte darzustellen. Dabei liegen pro Bildpunkt die Koordinaten vor. Die Darstellung der Koordinaten ist wichtig, denn in dieser Form wird die Unterschrift aufgenommen, in einzelnen Punkten pro Zeitabschnitt.

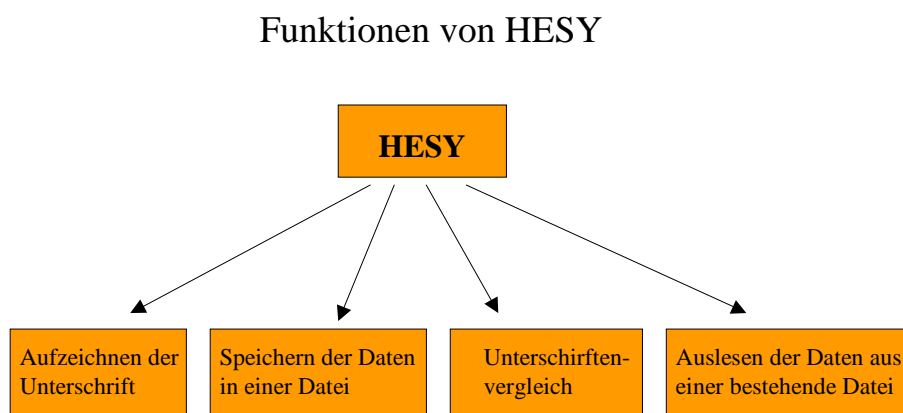


Abbildung 1.9 schematische Darstellung der Funktionen von HESY

Die so erhaltenen Werte können natürlich auch abgespeichert werden. Sie liegen dann als Textdatei vor und können dann z.B. in ein Dokument eingefügt werden. Des Weiteren ist es auch möglich, die so gespeicherten Daten einer Unterschrift wieder aus der Datei einzulesen und daraus das Bild der Unterschrift zu erstellen. Diese Funktion ist dann wichtig, wenn einmal Zweifel an der Echtheit einer Unterschrift bestehen. So können dann die Daten ausgelesen werden und das Bild kann dann mit dem Bild der zweifelhaften Unterschrift verglichen werden.

Eine weitere wichtige Funktion ist die Vergleichsfunktion. Mit Hilfe dieser Funktion ist es möglich, zwei Unterschriften zu vergleichen und festzustellen, ob sie von der gleichen Person stammen. Dabei wird berücksichtigt, dass ein Mensch niemals zweimal die exakt gleiche Unterschrift abliefert, sondern dass es immer Abweichungen gibt. Der Grad der Genauigkeit der Erkennung kann hier eingestellt werden. D.h. es wird ein gewisser Toleranzbereich angegeben, wie stark die beiden Unterschriften voneinander abweichen dürfen. Dieser Bereich ist frei einstellbar.

1.4 Umsetzung am IFB

Am IFB werden als Anwendungsprogramme, in denen zu unterschreibende Dokumente erstellt werden, hauptsächlich die Office-Anwendungen wie Word und Excel verwendet. Ein weiteres wichtiges Anwendungsprogramm ist das CAD-Programm CATIA. Diese Programme sind alle mit dem PDM-System SmarTeam gekoppelt. In die Verbindung zwischen den Anwendungsprogrammen und SmarTeam wird nun noch eine Software eingebaut, die es mit Hilfe des HESY ermöglicht, eine Unterschrift in das Dokument einzufügen und dieses dann direkt in das PDM zu verschieben (einzuchecken). In Abbildung 1.10 ist diese Vorgehensweise schematisch dargestellt.

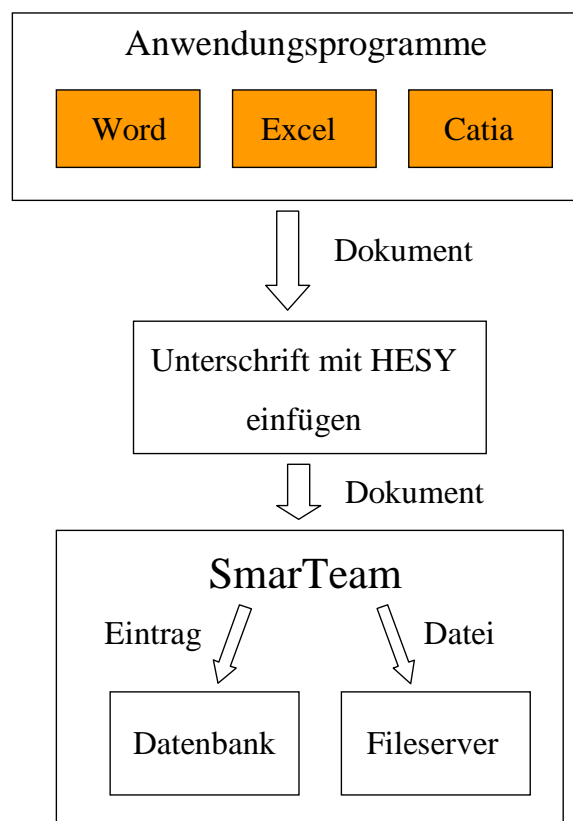


Abbildung 1.10 schematische Darstellung der Vorgehensweise am IFB.

2 Umsetzung in SmarTeam

Bei installiertem SmarTeam sind in den Programmen CATIA, Word und Excel wichtige SmarTeam Funktionen wie z.B. „checkin“, „checkout“ oder „release“ verfügbar, so dass diese Operationen direkt aus dem jeweiligen Anwendungsprogramm heraus aufgerufen werden können. Voraussetzung hierfür ist, dass SmarTeam gestartet wurde und dass auch zwischen dem Anwendungsprogramm und SmarTeam eine Verbindung hergestellt wurde. Dies geschieht über den Befehl „Activate SmarTeam“ in Word und Excel, oder über „Verbinden“ in CATIA. Die Befehle befinden sich jeweils im Menü „SmarTeam“ des Programms. In diesen Komplex wird jetzt noch zusätzlich HESY eingebaut.

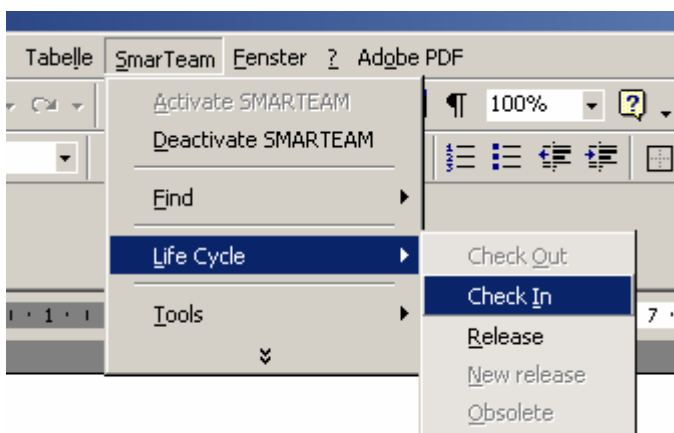


Abbildung 2.1 SmarTeam Menü in Word

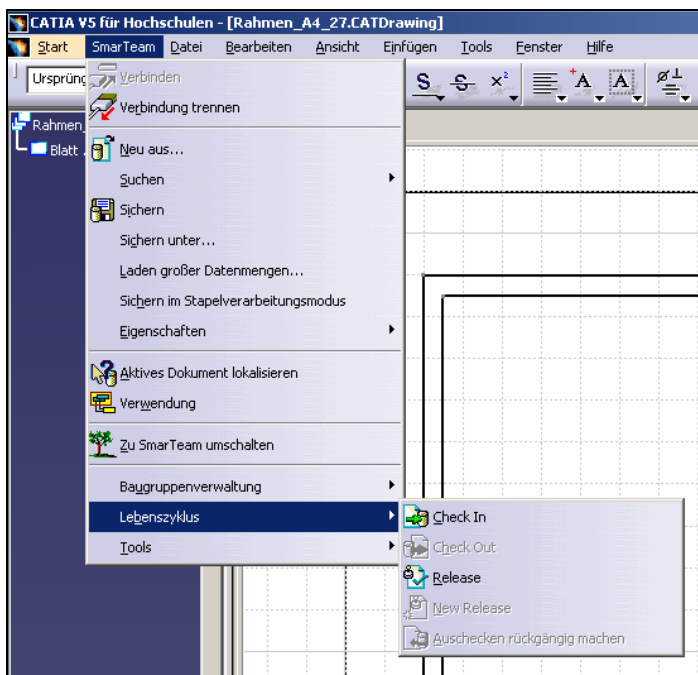


Abbildung 2.2 SmarTeam Menü in CATIA

2.1 Benutzerdefinierte Skripte in SmarTeam

SmarTeam bietet mit einem Tool das „SmarTeam Script Maintenance“ heißt, die Möglichkeit bei gewissen SmarTeam Ereignissen ein vom Administrator selbstgeschriebenes Skript einzubauen. Es besteht die Möglichkeit ein Skript vor, nach oder anstelle des ausgewählten Ereignisses zu starten.

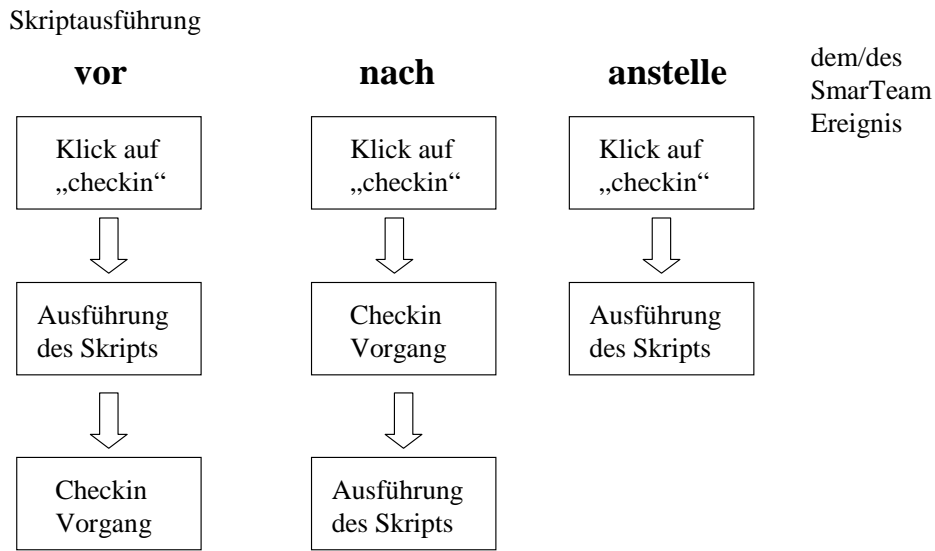


Abbildung 2.3 Darstellung der Skriptausführung bei einem bestimmten SmarTeam Ereignis. (Hier: Klick auf „Checkin“)

Als Beispiel wird das Ereignis „checkin“ betrachtet: Ist die Option gewählt, dass das Skript vor dem Ereignis ausgeführt wird, so startet das Skript sobald der Benutzer auf „checkin“ klickt. Das Skript wird nun komplett abgearbeitet, und erst dann beginnt der Checkin-Vorgang. Sind die Einstellungen so gewählt, dass das Skript nach dem Ereignis gestartet wird, so wird zuerst der Checkin-Vorgang durchgeführt, und dann sofort im Anschluss daran das Skript aufgerufen. Ist die Einstellung so gewählt, dass das Skript anstelle eines SmarTeam Ereignisses aufgerufen wird, so startet das Skript direkt nach klicken auf „checkin“. Nachdem das Skript dann durchgelaufen ist, wird der Checkin-Vorgang nicht mehr durchgeführt.

Neben dem Checkin-Ereignis gibt es noch eine Reihe anderer Ereignisse bei denen Skripte aufgerufen werden können. Nicht bei allen Ereignissen stehen alle drei Optionen (vor, nach, anstelle) zur Verfügung.

Weiterhin ist es möglich mit dem SmarTeam Skript Editor Skripte selbst zu erstellen. Die Sprache lehnt sich sehr stark an Visual Basic an, ist aber im Umfang bei weitem nicht so leistungsfähig. Sollen zum Beispiel Skripte mit einer Benuteroberfläche und Benutzerinteraktion geschrieben werden, so ist dies im SmarTeam Editor nicht möglich. Hier muss auf eine andere Programmiersprache ausgewichen werden. Das so geschriebene Programm kann wiederum über das SmarTeam Skript gestartet werden, so dass am Ende das gewünschte Ergebnis entsteht. Der Vorteil des SmarTeam Editor ist, dass alle Objekte des SmarTeam Modells zur Verfügung stehen und nicht wie in einer „externen“ Sprache noch Verweise auf die entsprechenden Objekte erstellt werden müssen.

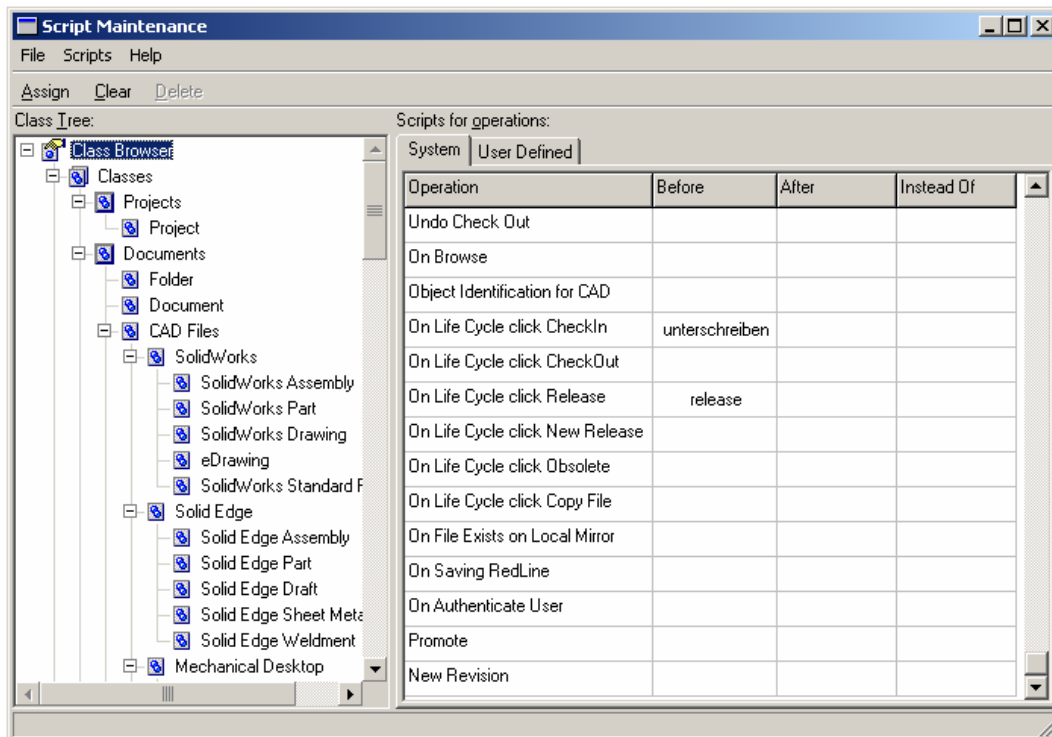


Abbildung 2.4 SmarTeam Skript Maintenance: Das bei einem bestimmten Ereignis auszuführende Skript wird hier in die entsprechende Spalte einfach eingetragen

2.2 Das Dialogfeld zum Unterschreiben

Um nun die von HESY aufgenommene Unterschrift in das Dokument einzufügen und dieses dann sofort in SmarTeam einzuspielen, wurde ein SmarTeam Skript mit dem Ereignis „on Life Cycle click checkin“ gekoppelt. Hier ist es nur möglich ein Skript vor dem eigentlichen Ereignis zu starten. Dies bedeutet, dass das Skript aufgerufen wird, sobald der Benutzer auf „checkin“ klickt. Dabei spielt es keine Rolle, ob dies in SmarTeam selbst geschieht, oder aus dem SmarTeam Menü von Word, Excel oder CATIA heraus.

Für ein ansprechendes Dialogfeld wurde aus den oben beschriebenen Gründen ein Programm in Visual Basic 6.0 geschrieben. Dieses Dialogfeld enthält alle Funktionen von HESY, sowie alle Funktionen um die Unterschrift in die jeweiligen Programme einzufügen. Dieses Programm liegt als so genannte Programmbibliothek vor und wird vom SmarTeam Skript aus gestartet. Die Datei heißt „smunterschreiben.dll“ und muss in Windows als Programmbibliothek registriert sein. Dateien im dll-Format sind hartcodiert, das bedeutet sie liegen im Binärkode vor und können nicht mehr in den ursprünglichen Quellcode zurück geführt werden. Somit sind die darin enthaltenen Informationen geschützt. Dies spielt im weiteren noch eine wichtige Rolle, wenn es um die Manipulationssicherheit des Systems geht.

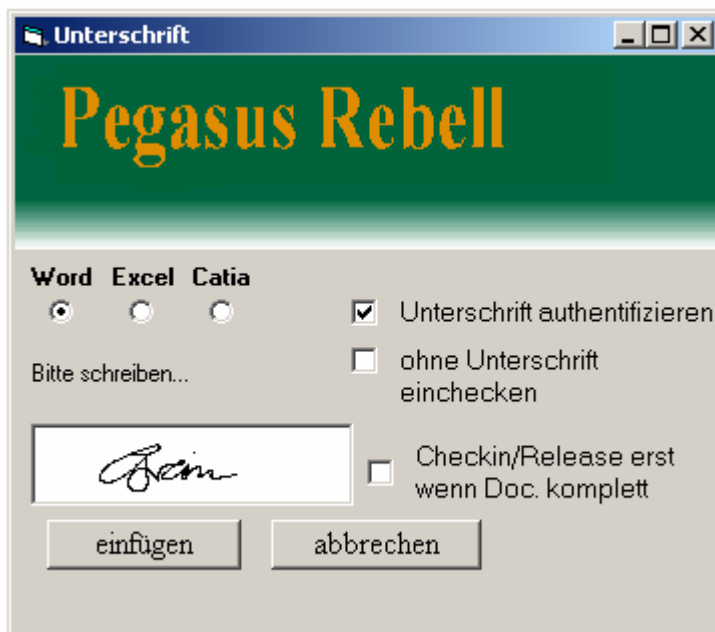


Abbildung 2.5 Benutzeroberfläche des Unterschriften-Programms

Im SmarTeam Skript selbst sind nur Funktionen enthalten, welche für den Checkin- bzw. Release-Vorgang ins SmarTeam verantwortlich sind.

Die Funktionen des HESY liegen ebenfalls in separaten Dateien vor und werden zusätzlich benötigt. Es handelt sich hierbei um die Dateien:

- biowr50g.dll
- biowr50v.dll
- biowr51r.dll

Diese Dateien müssen ebenfalls als Programmbibliothek in Windows registriert sein, um einen fehlerfrei Funktion des Programms zu gewährleisten. Es handelt sich hierbei um Funktionen zum Einlesen und Speicher oder Vergleichen von Unterschriften sowie auch Funktionen die eine Kommunikation des HESY über die COM Schnittstelle des Rechners ermöglichen.

Dateiname	HESY Funktionen in der Datei
biowr50g.dll	Laden der Unterschrift aus einer bestehenden Datei
biowr50v.dll	Vergleichen von Unterschriften
biowr51r.dll	Aufzeichnen der Unterschrift, Speichern als Datei, Kommunikation über COM Schnittstelle

Tabelle 2-1 HESY Funktionen in den entsprechenden Dateien.

2.2.1 Allgemeine Funktionen

Das Programm enthält einige Funktionen, die unabhängig von dem Programm sind, in das die Unterschrift eingefügt werden soll.

Um die Unterschrift korrekt einzufügen, ist es erforderlich vor dem Drücken des „einfügen“-Buttons das entsprechende Programm auszuwählen. Wird dies nicht getan, erscheint eine Fehlermeldung, die eine Auswahl fordert. Sollte eine falsche Auswahl getroffen werden, so bricht das Programm ab.

2.2.2 Checkin / Release Regeln

Je nach Wunsch des Benutzers gibt es zwei Möglichkeiten, wie mit Dokumenten verfahren wird, die mehrere Unterschriften benötigen. Es besteht die Möglichkeit nach jeder erfolgten Unterschrift das Dokument in SmarTeam einzuchecken und zu sichern. Sobald alle erforderlichen Unterschriften vorhanden sind, wird das Dokument freigegeben („release“). Wie dies im Einzelnen funktioniert wird im Folgenden noch genauer beschrieben. Der Vorteil von diesem Verfahren ist, dass das Dokument gleich nach der Unterschrift in den gesicherten Bereich gestellt wird und dort von den anderen Unterschreibenden gesehen werden kann. Jedoch muss jeder, der noch unterschreiben muss, das Dokument jedes Mal durch die „checkout“ Funktion aus dem gesicherten Bereich zu kopieren und dann wieder einzuspielen. Da aber hierbei dann eine neue Revisionsnummer an das Dokument vergeben wird, befinden sich dann bis zum Freigeben in SmarTeam mehrere Exemplare des Dokuments.

Um dies zu umgehen gibt es die Option, das Dokument erst in den gesicherten Bereich zu stellen, wenn alle Unterschriften vorhanden sind. Das Dokument befindet sich dann solange auf einem lokalen Rechner oder Server, bis die Unterschriften komplett sind. Sobald die erste Unterschrift in das Dokument eingefügt worden ist, ist dieses automatisch schreibgeschützt und kann nicht mehr verändert werden. Sind alle Unterschriften vorhanden, wird das Dokument direkt freigegeben. Um diese Funktion zu aktivieren, muss die Option „Checkin/Release wenn Dok. komplett“ gewählt werden.

Wird bei einem Dokument keine Unterschrift benötigt, so kann es auch direkt ohne Unterschrift eingechekkt werden. Dazu muss die entsprechende Option aktiviert sein. Freigeben ohne Unterschrift ist nicht möglich.

2.2.3 Unterschriftenerkennung

Die Erkennung der Unterschrift ist wichtige Funktion. Hierbei wird anhand der Unterschrift überprüft ob der Unterschreibende auch derjenige ist, der er vorgibt zu sein. Jeder SmarTeam Benutzer wird beim ersten Benutzen des Unterschriften-Programms dazu aufgefordert, sieben Unterschriftenproben abzugeben, die später als Vergleich genutzt werden. Diese Unterschriften werden im Profil des jeweiligen Benutzers gespeichert. Möchte nun dieser Benutzer ein Dokument unterschreiben, so wird vor dem Einfügen in das Dokument die Unterschrift mit den Proben verglichen bis eine Übereinstimmung erkannt wird. Sollte es zu keiner Übereinstimmung kommen, so wird der Benutzer noch einmal zu einer erneuten Unterschrift aufgefordert. Abbildung 2.6 verdeutlicht diesen Vorgang noch einmal.

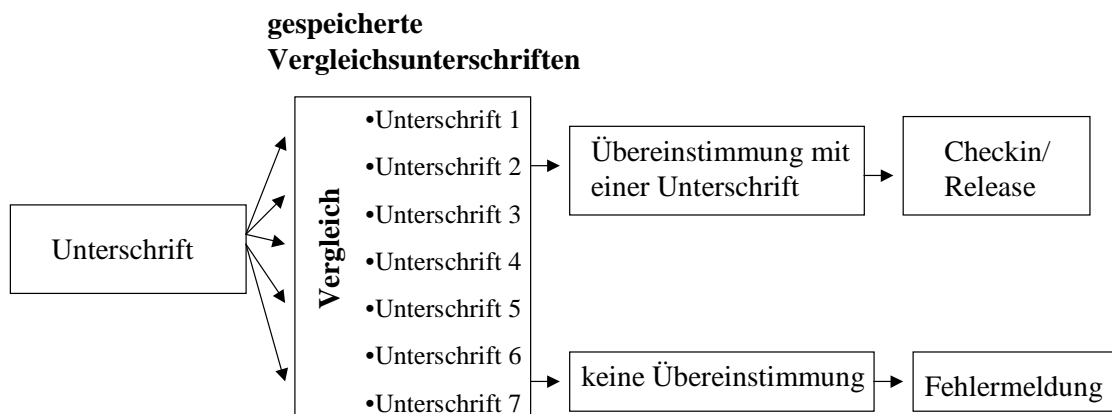


Abbildung 2.6 schematische Darstellung des Unterschriftenvergleichs

2.2.4 Checkin ohne Unterschrift

Sollte es erforderlich sein, ein Dokument ohne Unterschrift in SmarTeam einzuspielen, so ist dies durch aktivieren der entsprechenden Option möglich. Diese Option ist vor Allem für CATIA Dateien interessant, da hier außer Zeichnungen (CATIA Drawing) kein anderes Format unterschrieben werden kann. Alle anderen bestehenden CATIA-Formate müssen ohne Unterschrift ins SmarTeam eingecheckt werden.

2.3 Unterschriften in Word-Dokumenten

Das Unterschriften-Programm ist so geschrieben, dass Word-Dokumente unterschrieben werden können, bei denen bis zu neun Unterschriften erforderlich sind. Wenn die letzte Unterschrift eingefügt ist, wird das Dokument in SmarTeam auf „release“ gesetzt. Die Unterschrift wird in Word an der Stelle als Bild eingefügt, an welcher der Cursor zuletzt stand. Zusätzlich wird noch eine Datei mit den Unterschriftendaten am Ende des Word-Dokuments eingefügt. Der Name dieser Datei ist eine Kombination aus dem Namen des Dokuments sowie dem Benutzernamen des SmarTeam-Benutzers.

Die Anzahl der gesamten, sowie die Zahl der bereits eingefügten Unterschriften wird vom Programm direkt aus dem Word-Dokument ausgelesen. Dieser Mechanismus stellt natürlich auch gewisse Voraussetzungen an das Dokument, die erfüllen sein müssen. Die Zahl der gesamten Unterschriften ist in einer Textmarke gespeichert. Bei der Textmarke handelt es sich um eine in Word eingebaute Funktion, die es erlaubt, in einem Dokument, ähnlich einem Lesezeichen, Markierungen anzubringen. Die Textmarken sind für den Betrachter des Textes unsichtbar. Weiterhin ist es möglich zum schnelleren Blättern durch große Dokumente schnell von einer Textmarke zur nächsten zu springen.

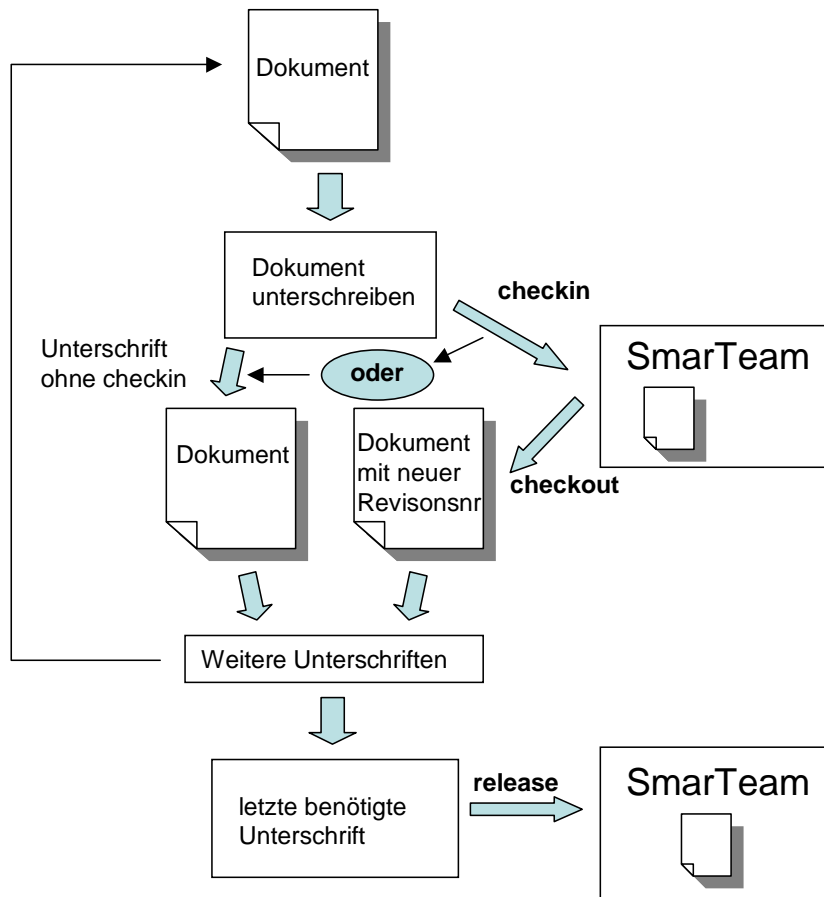


Abbildung 2.7 schematische Darstellung des Wegs eines Word- bzw. Excel-Dokuments von der ersten Unterschrift bis zur Freigabe (release).

Die Textmarke muss „smunterschriftX“ heißen, wobei X die Zahl der Unterschriften darstellt (z.B. smunterschrift3). Bei jeder eingefügten Unterschrift wird zusätzlich eine weitere Textmarke eingefügt, welche die schon gemachten Unterschriften zählt. Diese Textmarke hat die Form „smunterschriftenX“, wobei X hier die Zahl der bereits gemachten Unterschriften darstellt. Bei jeder neuen Unterschrift wird die alte Marke mit der aktuellen Zahl überschrieben. Die Textmarke „smunterschriftX“ muss in dem Dokument schon vorhanden sein. An welcher Stelle im Dokument die Textmarke angebracht wird ist unerheblich.

Sobald die Unterschrift eingefügt ist, wird das Dokument durch den Word-eigenen Schreibschutz mit Passwort geschützt. Das Passwort ist im hartcodierten Teil hinterlegt und somit, wenn überhaupt, nur den Programmierern bekannt. Somit wird verhindert, dass nach der Unterschrift noch Veränderungen vorgenommen werden können. Das Hinzufügen von weiteren Unterschriften ist aber nach wie vor möglich, vorausgesetzt das Dokument ist noch nicht komplett unterschrieben.

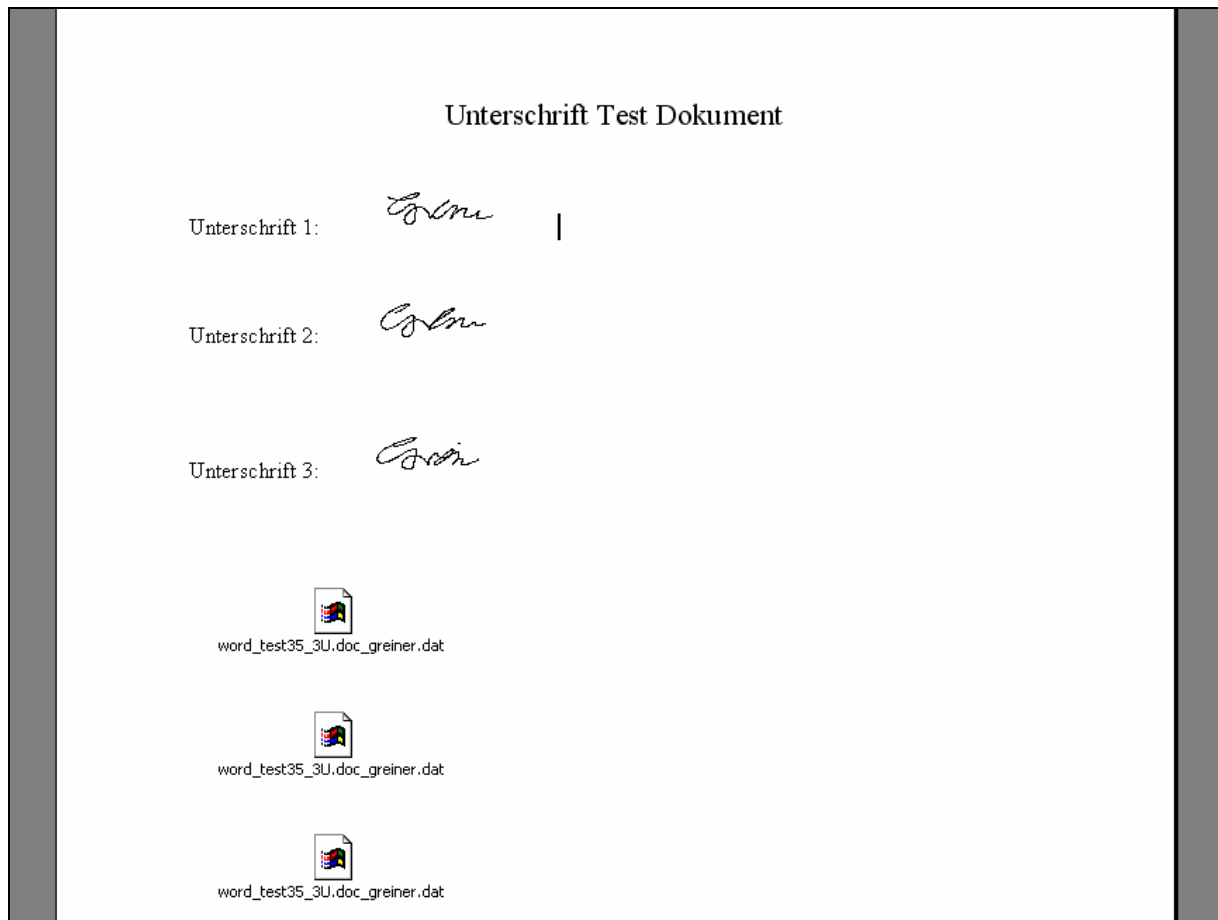


Abbildung 2.8 Worddokument mit drei Unterschriften, sowie den dazugehörigen Dateien, welche unter dem Dokument angehängt sind.

Abbildung 2.8 zeigt ein vollständig unterschriebenes Word-Dokument. Zu sehen sind die Bilder der einzelnen Unterschriften, sowie am Ende des Dokuments die Dateien mit den Daten der Unterschriften.

2.4 Unterschriften in Excel

Bei Excel verhält es sich ähnlich wie bei Word. Auch hier ist es möglich, bis zu neun Unterschriften in die Dokumente einzufügen. Nach der letzte Unterschrift wird hier ebenfalls das Dokument in SmarTeam auf „release“ gesetzt. Auch werden die Daten der Unterschrift als Datei am Ende des Dokuments eingefügt. Im Gegensatz zu Word, funktioniert in Excel die Erkennung der Zahl der Unterschriften etwas anders. Da es in Excel keine Textmarken im Sinne von Word gibt, wurde hier die entsprechende Excel-Funktion gewählt, die „Name“ heißt. Es muss auch hier vor dem ersten Unterschreiben eine Name der Form „smunterschriftX“ mit der Zahl der gesamten Unterschriften vorhanden sein. Im Weiteren wird hier ein zweiter Name „smunterschriftenX“ eingefügt, der die Zahl der bereits vorhandenen Unterschriften enthält.

Nach erfolgter Unterschrift wird auch das Excel-Dokument mit Passwort schreibgeschützt, und kann nicht mehr geändert werden.

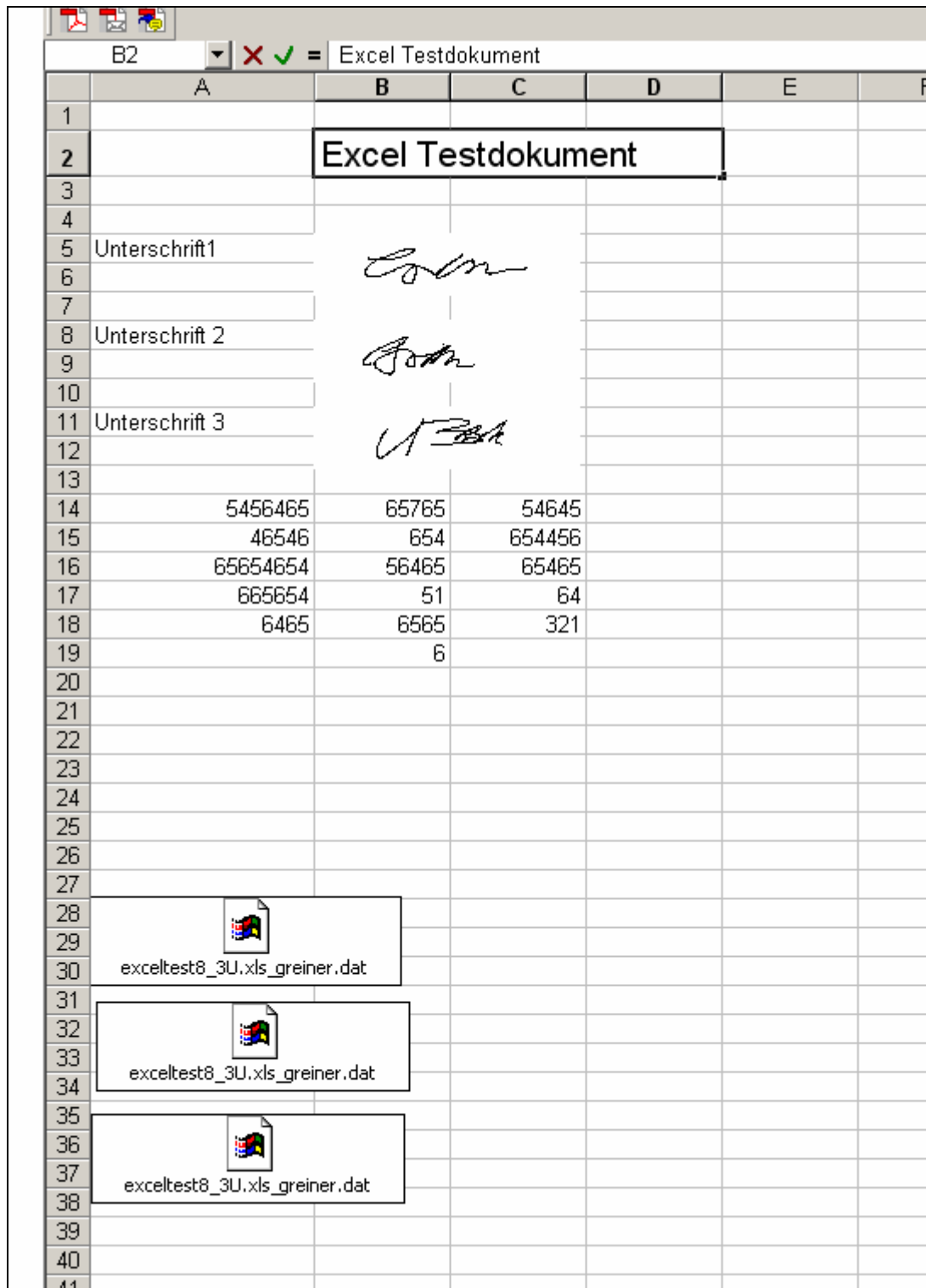


Abbildung 2.9 komplett unterschriebenes Excel Dokument. Die Dateien mit den Daten der Unterschrift werden am Ende eingefügt.

2.5 Unterschriften in CATIA

In CATIA können nur Zeichnungen (CATIA Drawings) unterschrieben werden. Alle übrigen CATIA-Formate müssen ohne Unterschrift in SmarTeam eingespielt werden. In einem DIN genormten Schriftfeld sind nur zwei Unterschriftsmöglichkeiten vorhanden. Eine Unterschrift für den Bearbeiter der Zeichnung, und eine zweite der Prüfung. Die erste Unterschrift wird automatisch in das Feld „bearbeitet“ im Schriftfeld eingefügt, die zweite im Feld „geprüft“. Um die Zeichnung freizugeben, ist eine weitere Unterschrift nötig, die jedoch nicht auf der Zeichnung abgebildet wird. Leider ist es bis zu der CATIA Version V5 R15 noch nicht möglich, automatisiert eine Datei in eine Zeichnung einzufügen. Deshalb wurde hier, im Gegensatz zu Word/Excel ein anderer Weg gewählt: Dieser Weg ist in Abbildung 2.10 graphisch verdeutlicht worden.

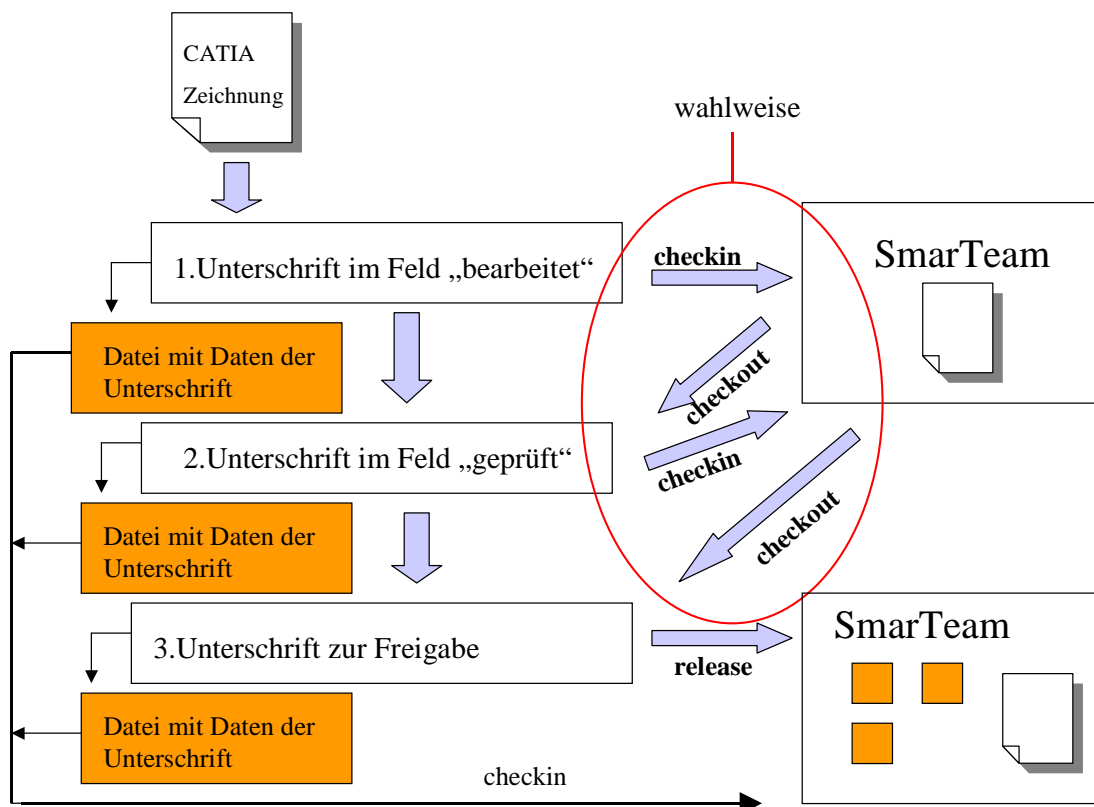


Abbildung 2.10 schematische Darstellung des Weges einer CATIA-Zeichnung von der ersten Unterschrift bis zur Freigabe.

Die Daten der Unterschrift werden bei CATIA als eigene Datei, im selben Verzeichnis wie die Zeichnung, ins SmarTeam eingefügt. Der Dateiname setzt sich aus dem Namen der CATIA-Zeichnung und dem Namen des Unterschreibenden zusammen. Bei der ersten Unterschrift wird noch der Zusatz „bearbeitet“ dem Dateinamen zugefügt. Bei der zweiten Unterschrift der Zusatz „geprüft“.

Um die Zeichnung nun freizugeben, ist noch eine weitere Unterschrift nötig. Diese Unterschrift wird ebenfalls als Datei mit dem Zusatz „release“ im selben Verzeichnis der Zeichnung abgespeichert. Ein Bild dieser Unterschrift wird aber nicht in die Zeichnung eingefügt, da hier nur zwei Unterschriftenfelder vorhanden sind.

Alle Dateien, die Daten von Unterschriften enthalten, werden sofort in SmarTeam eingespielt und erhalten somit den Status „checkin“. Damit ist auch eine Manipulation der Daten ausgeschlossen. Bei der zu unterschreibenden Zeichnung verhält es sich etwas anders: Sie wird nach den ersten beiden Unterschriften nicht zwangsläufig in SmarTeam eingecheckt. Dies hängt davon ab, ob der Benutzer die Option „checkin erst wenn Dokument komplett“ aktiviert hat. Ist dies der Fall, so verbleibt die Zeichnung auf dem lokalen Rechner des Benutzers. Spätestens bei der dritten Unterschrift, welche die Zeichnung freigibt, ist diese Option nicht mehr möglich. Dann wird die Zeichnung in SmarTeam eingecheckt und sofort auf „release“ gesetzt.

Leider fehlt bei CATIA auch in der aktuellen Version die Möglichkeit, Zeichnungen oder andere CATIA Produkte schreibzuschützen. Deshalb sind Zeichnungen nur durch den internen Schutz von SmarTeam gegen unerlaubtes Verändern gesichert.

Um nach erfolgter Unterschrift auch die Dateien mit den Unterschriften im Editor zu sehen, muss noch zusätzlich die F5-Taste gedrückt werden. Alternativ kann auch der Editor neu gestartet werden.

Abbildung 2.11 zeigt einen Ausschnitt aus SmarTeam, bei welchem eine CATIA-Zeichnung und die Dateien der eingefügten Unterschriften zu sehen sind. Das Dokument ist vollständig unterschrieben und somit freigegeben.

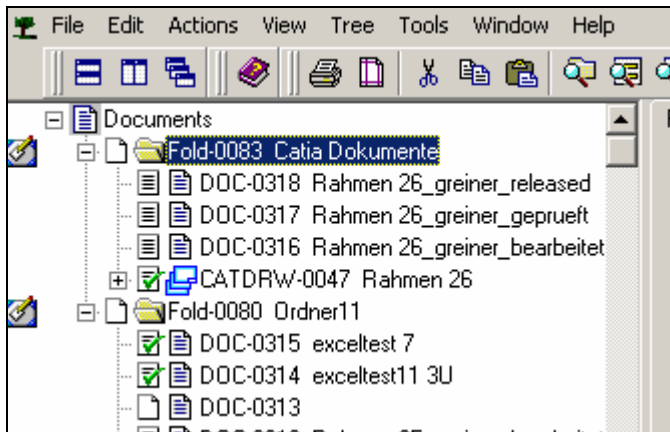


Abbildung 2.11 Dateien mit den Daten der einzelnen Unterschriften. Bei CATIA Zeichnungen werden die Dateien nicht angehängt, sondern direkt in SmarTeam eingecheckt.

In Abbildung 2.12 ist ein Ausschnitt aus einem DIN genormten Schriftfeld abgebildet. Zu sehen sind die beiden Unterschriftenfelder „bearbeitet“ und „geprüft“. Beide sind jeweils mit Unterschrift versehen.



		Datum	Name
	Bearb		
	Gepr.		
	Norm		

Abbildung 2.12 DIN Normfeld einer CATIA Zeichnung mit erfolgten Unterschriften in den Feldern „bearbeitet“ und „geprüft“

2.6 Freigabe von Dateien ohne Unterschrift

Bei den meisten CATIA-Formaten, außer Zeichnungen (CATIA Drawing), ist es nicht möglich eine Unterschrift direkt in die Konstruktion einzufügen. Der Großteil der Dateien sind jedoch gerade solche Formate (CATIA-Product, CATIA-Part). Diese werden über die Funktion „Checkin ohne Unterschrift“ in SmarTeam eingecheckt. Wenn aber diese Dokumente freigegeben werden sollen, so ist auch hier eine Unterschrift erforderlich. Die Funktion ist hier ähnlich wie beim Freigeben von CATIA-Zeichnungen. Der Benutzer unterschreibt für die Freigabe, und eine Datei mit den Daten seiner Unterschrift wird in das Verzeichnis der freigegebenen Datei hineinkopiert und auf den Status Checkin gesetzt.

Wird in SmarTeam, oder in CATIA, die Funktion „release“ angeklickt, so öffnet sich eine Benutzeroberfläche, welche derer beim Checkin-Vorgang sehr ähnelt. Jedoch sind hier einige Funktionen weniger vorhanden. Die einzige Auswahl die noch vorhanden ist, ist die Möglichkeit zur Authentifizierung der Unterschrift. Der Algorithmus ist der gleiche wie beim Checkin-Vorgang. Wird die Unterschrift nicht erkannt, wird der Benutzer gebeten seine Eingabe zu wiederholen.

Über die Benutzeroberfläche wird die Unterschrift aufgezeichnet und dann als Datei abgespeichert. Das SmarTeam Skript übernimmt dann wie beim Checkin-Vorgang das Einspielen der Unterschriftendatei, sowie das Freigeben der ausgewählten Dateien. Der Dateiname der Dialogfläche ist „smfreigeben.dll“.

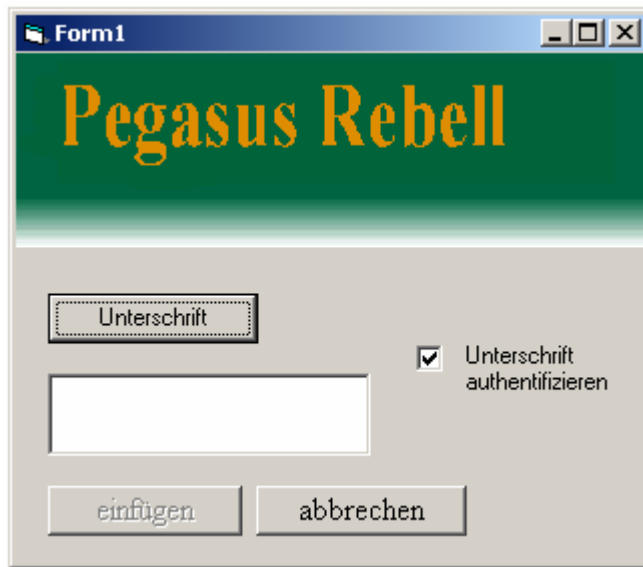


Abbildung 2.13 Benutzeroberfläche für den Release Vorgang. Hier besteht nur noch die Möglichkeit, die Authentifizierung der Unterschrift abzuschalten. Ein Programm muss hier nicht ausgewählt werden.

3 Funktion des PDMs anhand eines Beispiels

Im folgenden Kapitel wird die Funktionsweise, sowie die Einbindung von HESY anhand eines Beispiels verdeutlicht. Dabei wird ein komplettes Flugzeug, das in CATIA erstellt wurde, ins SmarTeam eingespielt bzw. eingecheckt. Auch wird auf die Besonderheiten von SmarTeam im Zusammenspiel mit CATIA näher eingegangen.

3.1 Sichern und einspielen von komplexen Strukturen

Bevor Dateien in SmarTeam eingecheckt werden können, müssen diese erst in SmarTeam gesichert werden. Dies bedeutet, dass die Dateien sozusagen vorgemerkt werden. Dabei wird in der SmarTeam Datenbank ein Eintrag mit den wichtigsten Informationen wie Dateiname und Pfad bezüglich der Datei vorgenommen. Die Datei bleibt aber noch an ihrem ursprünglichen Ort. Dies geschieht über die Funktion „sichern“ im Menü „SmarTeam“ in CATIA.

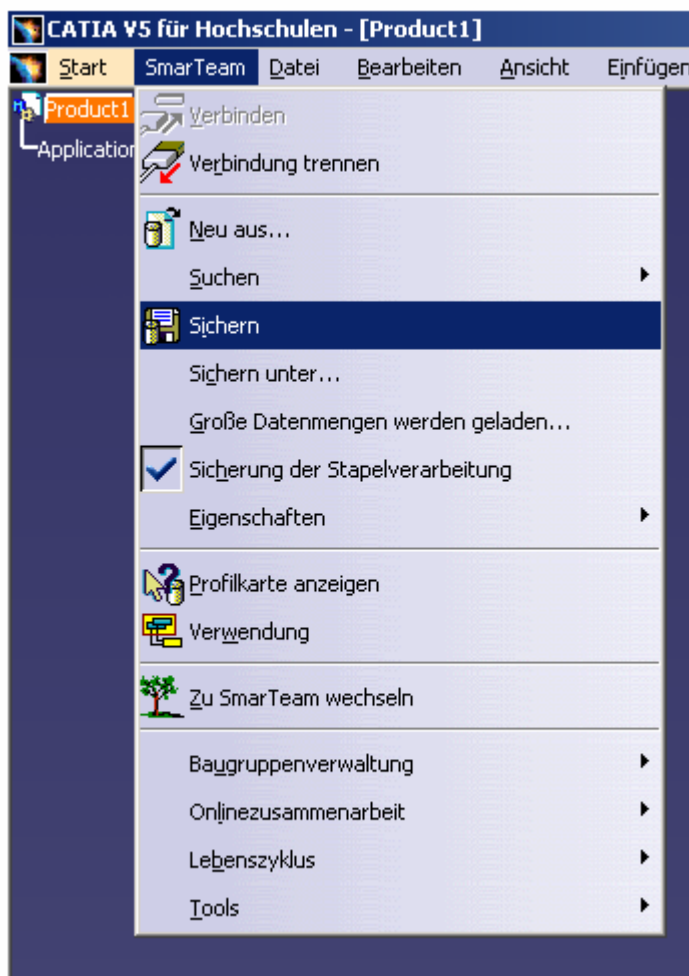


Abbildung 3.1 sichern von Dateien oder komplexeren Strukturen im Menü „SmarTeam“ in CATIA

Jede, etwas größere Konstruktion umfasst zwangsläufig mehrere Komponenten. Dabei kann es sich um die unterschiedlichsten Formate handeln. Eine Konstruktion muss nicht zwangsläufig nur aus CATIA-Formaten wie Produkte oder Teile bestehen, auch andere Formate wie z.B. Excel-Tabellen oder Bilder können in die Konstruktion mit einbezogen werden. Dabei entsteht unter den einzelnen Komponenten Abhängigkeiten. Es entsteht dann ein komplexe Baumstruktur, wie sie auch schon aus Windows bekannt ist. Eine solche Struktur ist in Abbildung 3.2 exemplarisch dargestellt.

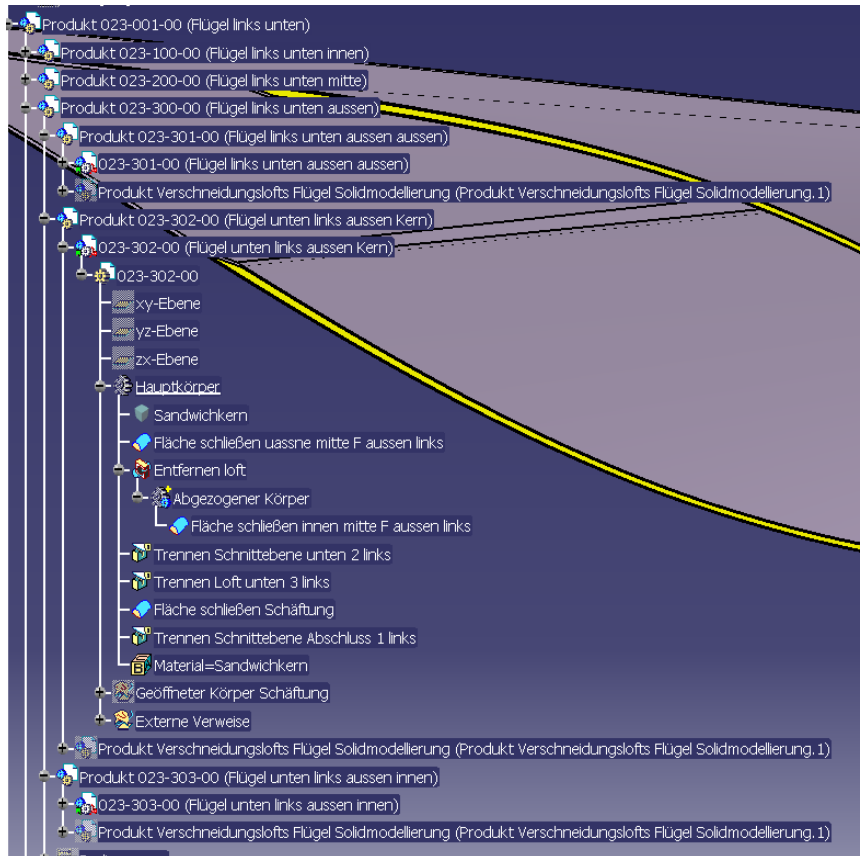


Abbildung 3.2 beispielhafte Baumstruktur in CATIA

Ziel beim Sichern und Einchecken ist es nun auch in SmarTeam nachher diese Struktur beizubehalten. Praktischerweise muss man nun in SmarTeam diese Struktur nicht von Hand anlegen, was bei einem großen Projekt ein enormer Arbeitsaufwand wäre, sondern diese Struktur von CATIA wird beim Sichern automatisch mit übernommen. Es ist ausreichend, wenn das in der Hierarchie am höchsten stehende CATIA-Produkt geladen wird und dann dieses in SmarTeam gesichert wird. Alle zu diesem Produkt gehörenden Komponenten werden automatisch erkannt und auch in SmarTeam gesichert. Dabei bleiben die Struktur und die Verknüpfungen der Teile untereinander erhalten.

Um diese Struktur der Teile und Produkte auch exakt zu übernehmen, legt SmarTeam in jedem Verzeichnis, aus welchem Dateien gesichert werden, einen weiteren Ordner an, in dem Informationen über den strukturellen Zusammenhang gespeichert sind. Diese Ordner sind sehr wichtig, damit CATIA auch Dateien eines Dokuments wiederfindet, die bereits schon in SmarTeam eingechekkt wurden. Somit kann auch ein komplettes CATIA-Projekt am Rechner angeschaut und bearbeitet werden, auch wenn schon Teile davon eingespielt wurden. Wenn ein bereits eingespielte Komponente noch geändert werden soll, so ist dies nur möglich, wenn sie durch den Auscheck-Vorgang aus dem gesicherten Bereich herauskopiert wird.

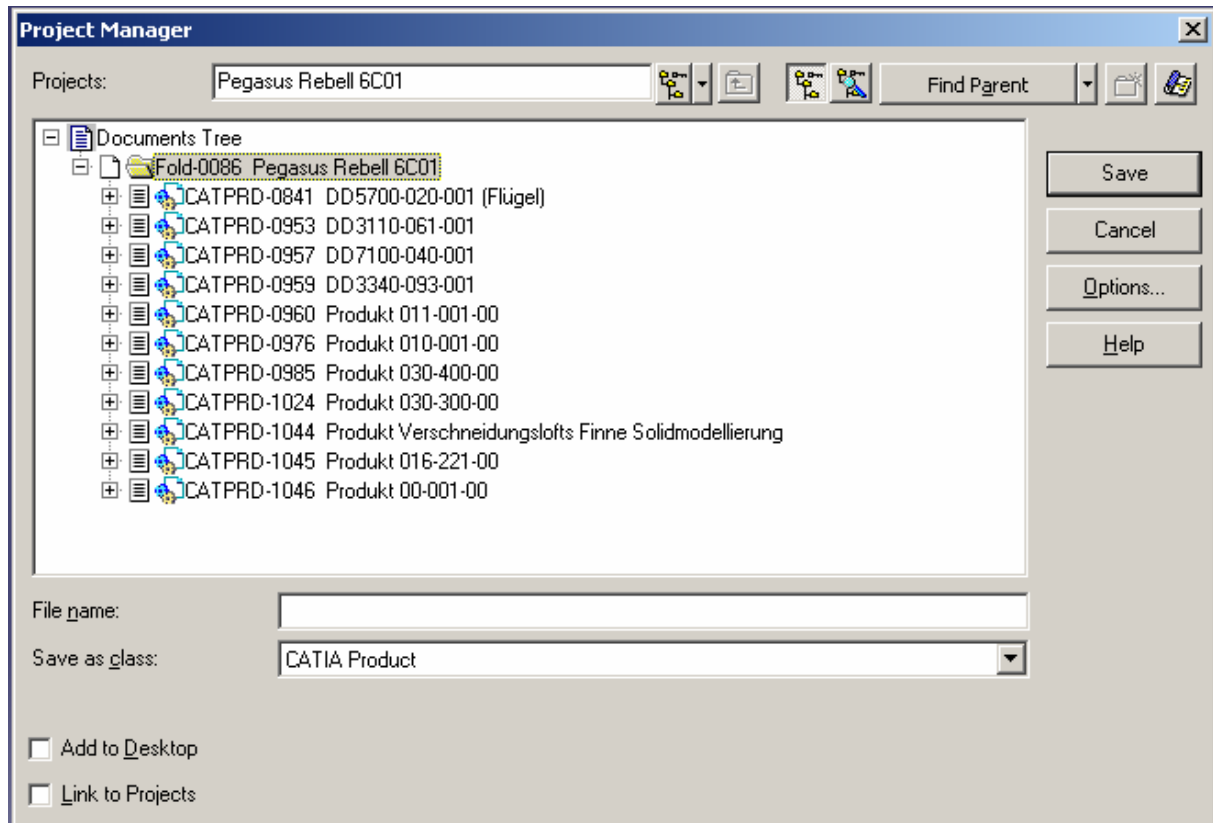


Abbildung 3.3 SmarTeam-Fenster, das nach dem klicken auf „sichern“ erscheint. Hier besteht die Möglichkeit den Ordner des zu sichernden Dokuments/Produkts zu wählen.

Nach dem Klicken auf „sichern“ erscheint das in Abbildung 3.3 dargestellte Fenster. Hier werden die bereits im ausgewählten Ordner gesicherten, bzw. eingechekkten Dokumente und Produkte angezeigt. Beim Klicken auf das Pluszeichen werden die in der Hierarchie weiter unten stehenden Objekte angezeigt. Der Status des jeweiligen Dokuments ist durch das vorangestellte Icon graphisch verdeutlicht.

Durch betätigen des „Save“ Buttons wird die Sicherungsprozedur gestartet, und alle zum gewählten Objekt gehörenden Unterobjekte mit ins SmarTeam geschrieben.

Der nächste Schritt ist der Checkin Vorgang selbst. Die Dateien müssen vorher gesichert werden, sonst ist die Option „checkin“ in SmarTeam nicht verfügbar. In der Benutzeroberfläche für die Unterschrifteneingabe, muss bei allen CATIA-Formaten, bis auf Zeichnungen (CATIA Drawing) die Option „Checkin ohne Unterschrift“ gewählt werden, da in diese Formate keine Unterschrift eingefügt werden kann. Durch Klicken auf „Einchecken“ wird der Vorgang gestartet, und alle zum ausgewählten Projekt gehörenden Dateien werden automatisch eingecheckt. In Abbildung 3.4 ist diese Vorgehensweise noch einmal schematisch dargestellt.

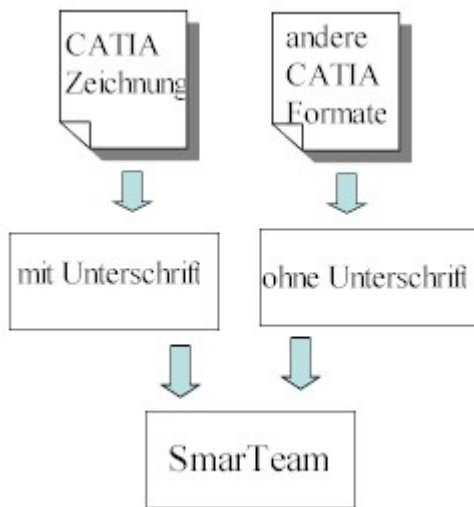


Abbildung 3.4 Vorgehensweise bei verschiedenen CATIA-Formaten

Durch Klicken auf „Einchecken“ wird der Vorgang gestartet, und alle zum ausgewählten Projekt gehörenden Dateien werden automatisch eingecheckt.

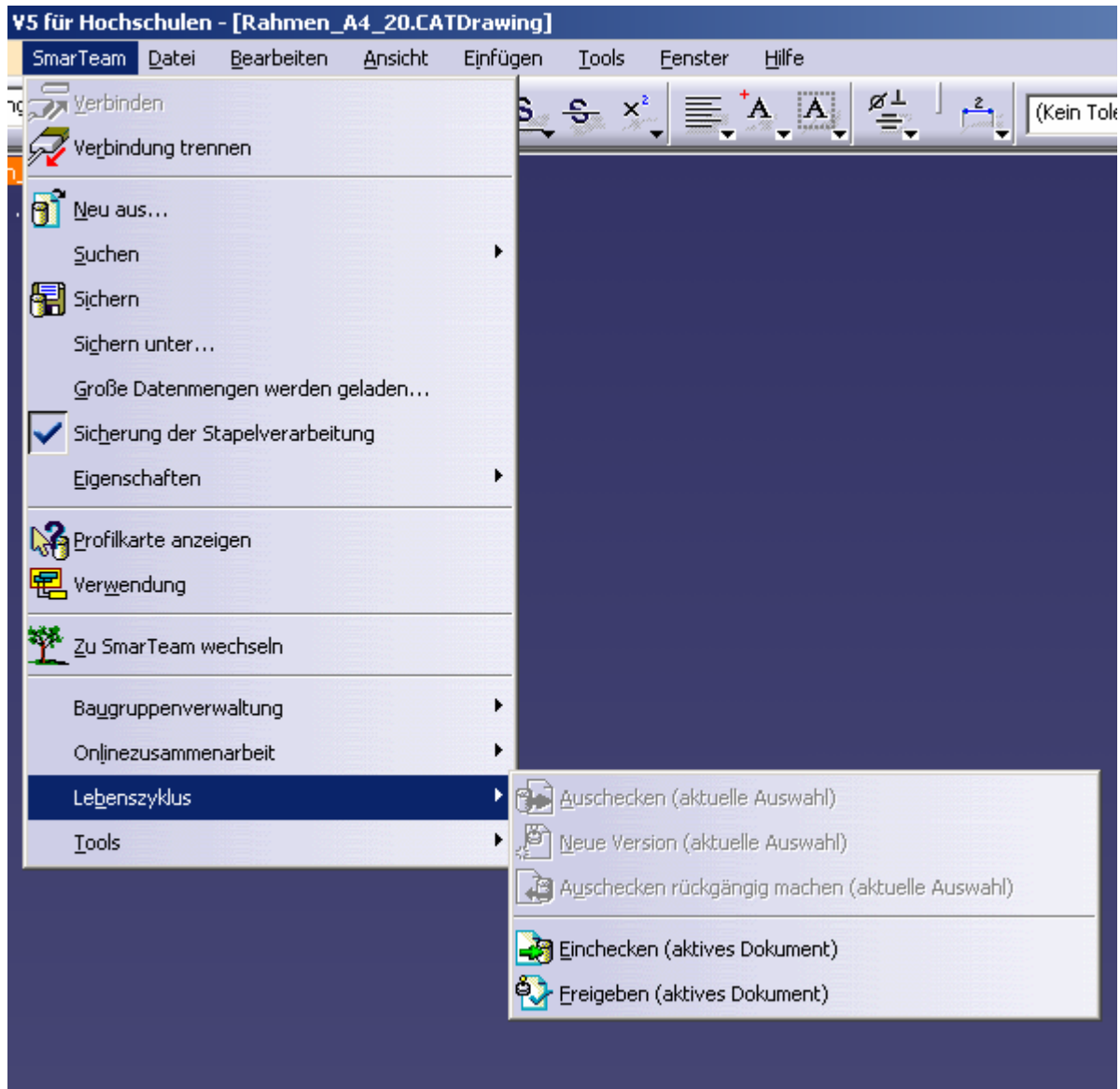


Abbildung 3.5 durch Klicken auf „Einchecken“ im Untermenü „Lebenszyklus“ wird der Vorgang gestartet

Nach dem Klicken auf „Einchecken“ erscheint die Benutzeroberfläche mit der Aufforderung zur Eingabe der Unterschrift.

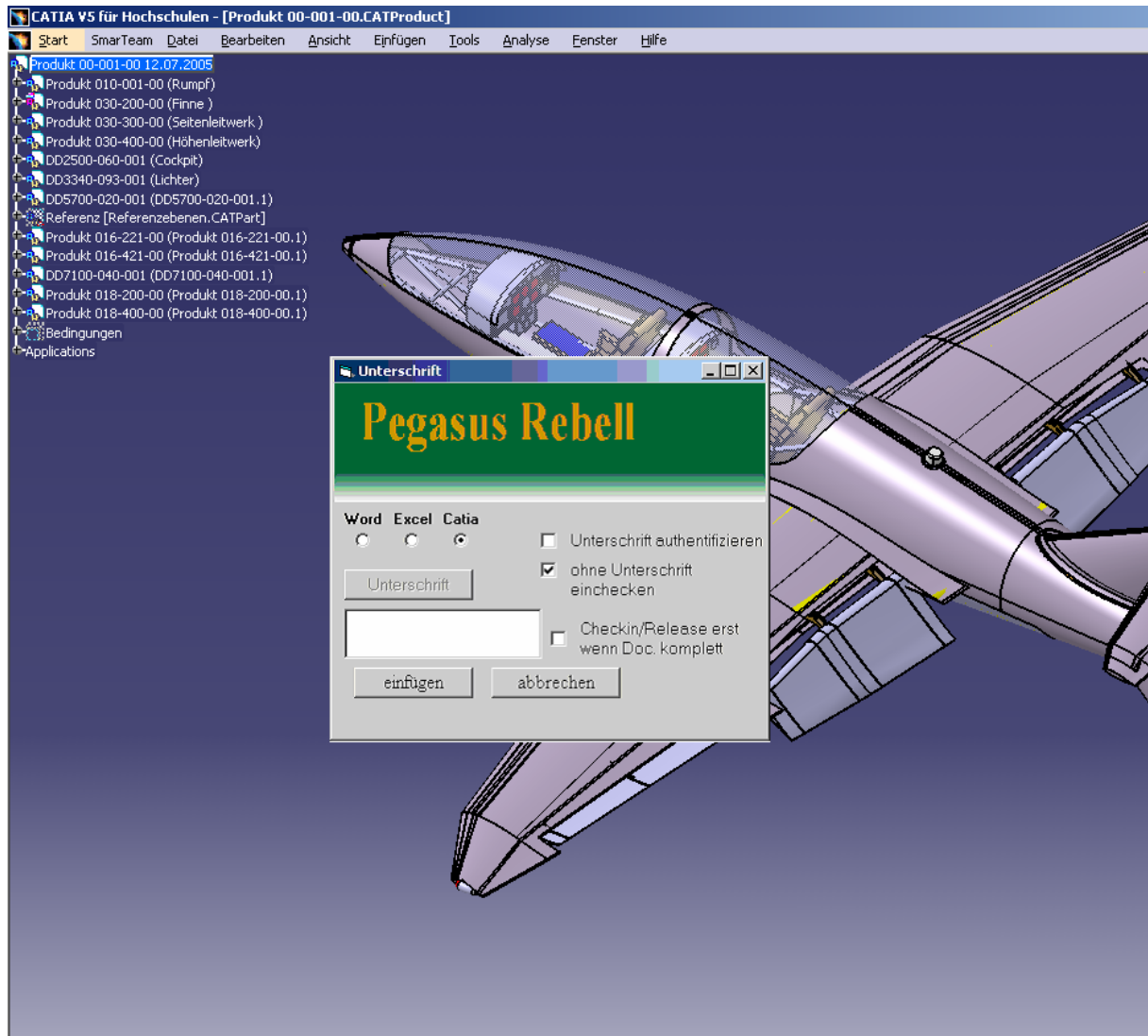


Abbildung 3.6 Die kompletten Daten des gesamten Flugzeugs werden ohne Unterschrift ins SmarTeam eingechekkt.

Ist der Checkin-Vorgang beendet, werden alle Dateien von ihrem ursprünglichen Ort gelöscht und in den *Vault* verschoben. Dabei wird die ursprüngliche Baumstruktur, wie sie vor dem Einchecken bestand, übernommen. In Abbildung 3.7 ist ein Teil dieser Struktur der eingespielten Dateien dargestellt.

3.2 Release von unterschriebenen Dokumenten

Wie schon in Kapitel 2.5 erwähnt, müssen Zeichnungen von drei Personen unterschrieben werden um den Status „release“ zu erlangen. Die Zeichnung wird dann je nach Wahl des Benutzers entweder nach jeder erfolgten Unterschrift in SmarTeam eingechekkt, oder sie bleibt so lange auf dem Rechner des Benutzers, bis alle Unterschriften vorhanden sind. Dann erfolgt der Release Vorgang automatisch.

Ist die Zeichnung erstellt und soll die erste Unterschrift erfolgen, muss auch hier die Zeichnung zuerst in SmarTeam gesichert werden. Danach ist dann die Option „checkin“ möglich. Nach dem Klicken auf „checkin“ erscheint daraufhin die Oberfläche zur Eingabe der Unterschrift.

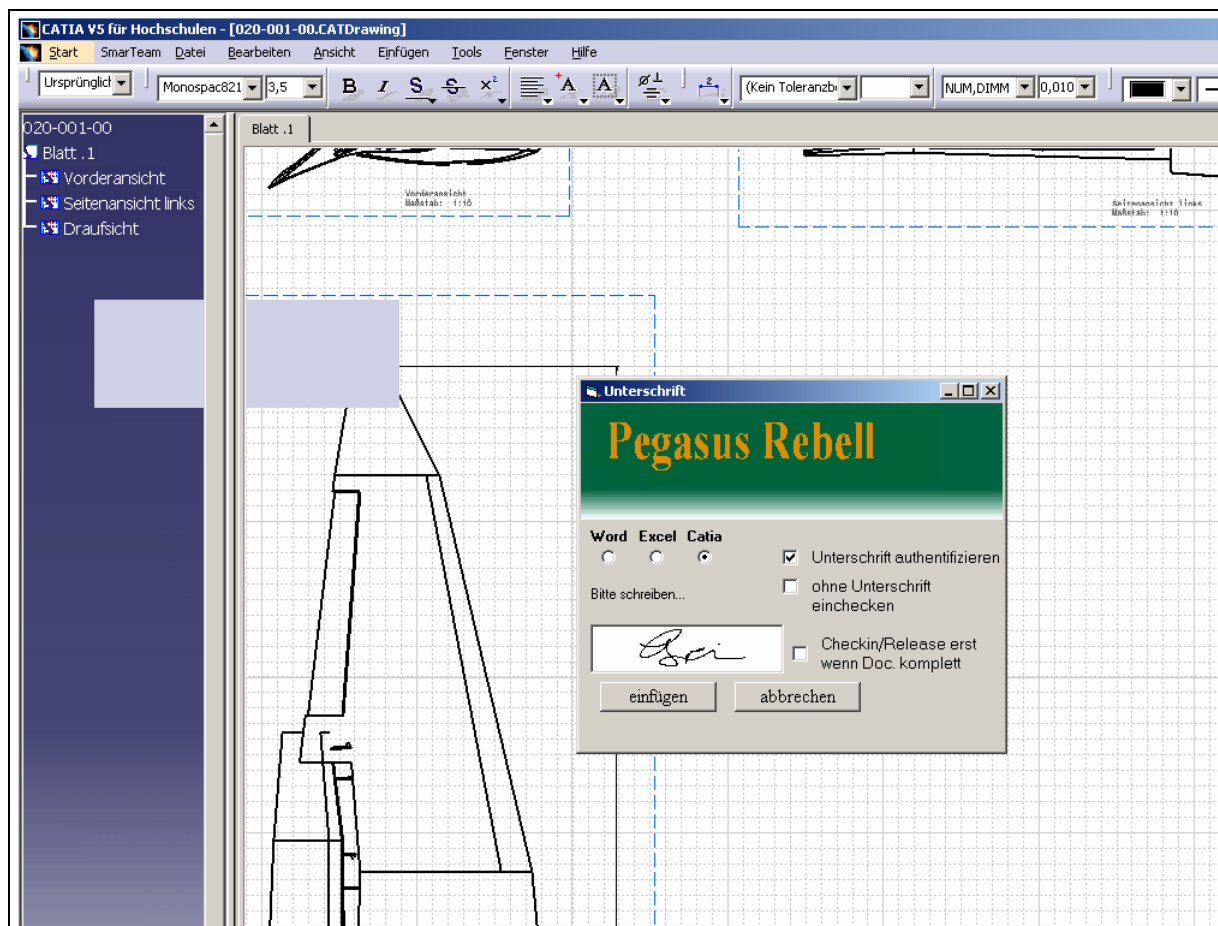


Abbildung 3.8 Einfügen der Unterschrift in eine CATIA Zeichnung

Sind in der Zeichnung nun schon zwei Unterschriften vorhanden, so wird bei einem erneuten Klicken auf „einchecken“ automatisch die Unterschrift zum Freigeben angefordert. Die Oberfläche ist die gleiche, wie bei den Vorgängen zuvor. Jedoch wird nun nach dem Klick auf „einfügen“ das Dokument freigegeben. Die Unterschrift wird daraufhin als Datei in SmarTeam eingecheckt.

Ebenfalls wäre es möglich, nach der zweiten Unterschrift auch auf „Freigeben“ zu klicken. Hier öffnet sich dann die spezielle Oberfläche zum Unterschreiben für die Freigabe. Das Resultat beim Benutzten diese Oberfläche ist das gleiche, wie beim oben beschriebenen Weg.

Die Dateien mit den Unterschriften und die CATIA-Zeichnung werden dann in SmarTeam, wie in Abbildung 3.10 abgebildet, dargestellt.

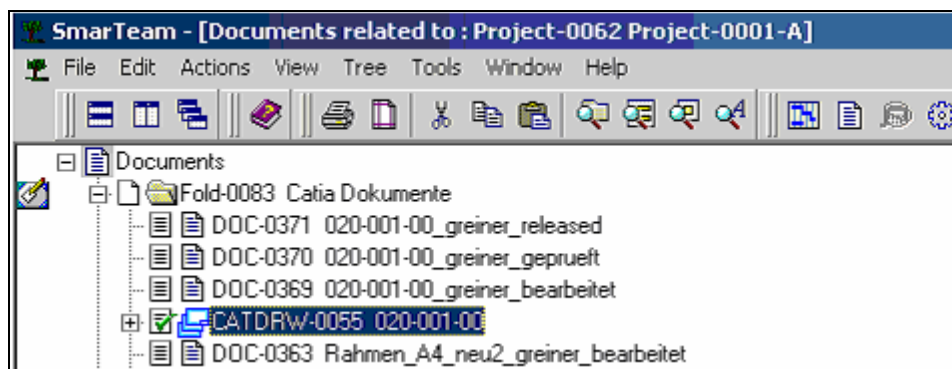


Abbildung 3.9 freigegebene CATIA Zeichnung mit den drei Unterschriften, als Dateien im gleichen Ordner wie die Zeichnung abgelegt.

3.3 Freigeben von Dokumenten, die nicht unterschrieben werden können

Während CATIA Konstruktionen wie Produkte und Teile noch ohne Unterschrift einzuchecken waren, so sind auch hier für die Freigabe Unterschriften nötig. Wie schon erwähnt, werden die Unterschriften in einer separaten Datei im selben Verzeichnis im SmarTeam abgelegt. Der Status der Unterschriftendatei ist dabei „checked in“.

Wenn alle freizugebenden Dateien eingchecked sind, kann durch Klicken auf „freigeben“ die Oberfläche zum Eingeben der Unterschrift aufgerufen werden. Es ist auch möglich, Dateien direkt freizugeben, ohne sie vorher einzuchecken. Allerdings müssen sie auf jeden Fall vorher im SmarTeam gesichert worden sein, sonst stehen diese Optionen nicht zur Verfügung.

Beim Freigeben werden, wie beim Einchecken auch, die in der Hierarchie darunter liegenden Komponenten ebenfalls freigegeben. Es ist somit ausreichend, das in der Hierarchie höchste Element freizugeben.

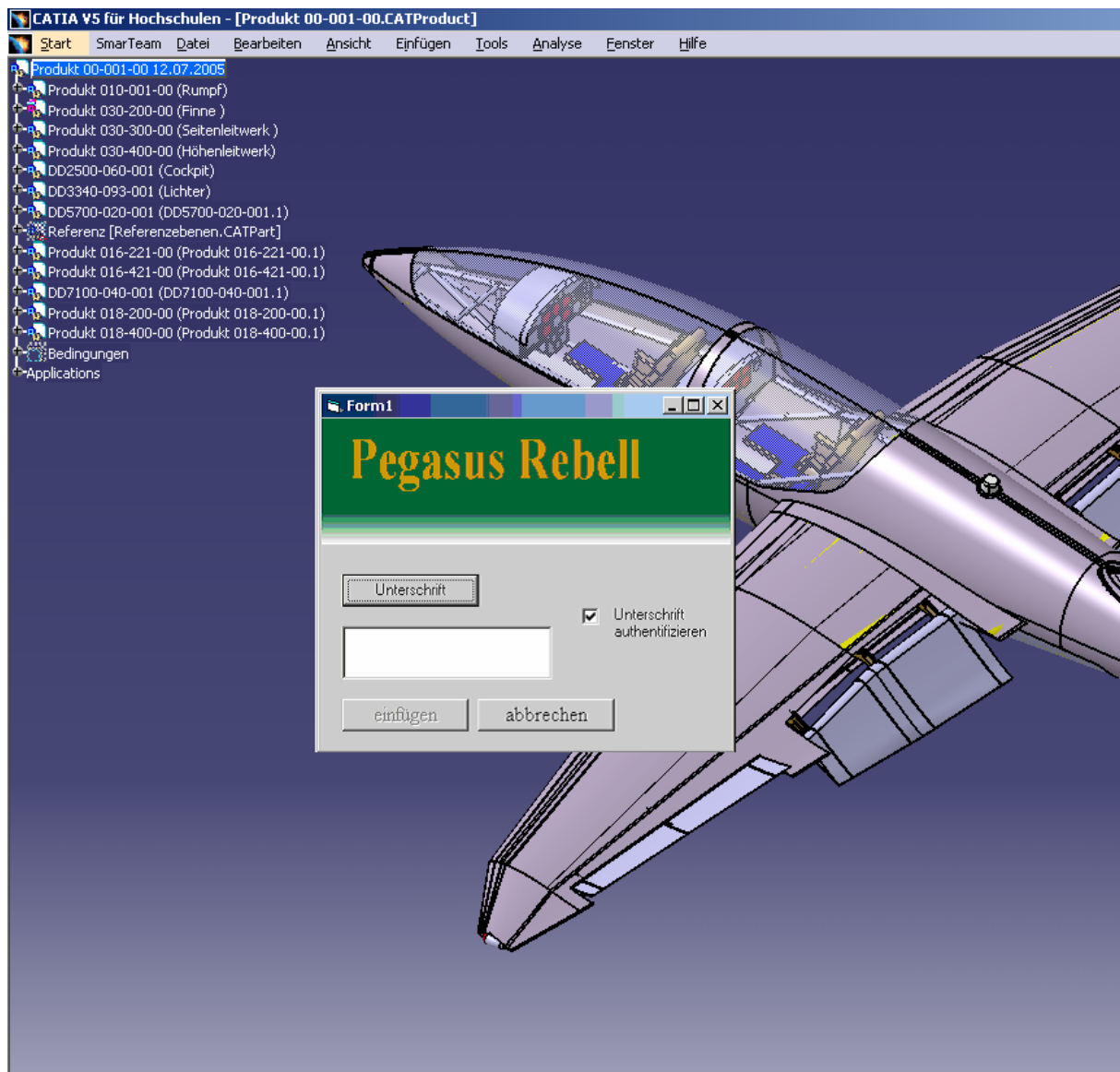


Abbildung 3.10 nach dem Klicken auf „freigeben“ erscheint die Benutzeroberfläche zur Eingabe der Unterschrift für die Freigabe.

In Abbildung 3.11 sind die nun freigegebenen Dateien in der SmarTeam Baumstruktur dargestellt. Graphisch verdeutlicht wird die Freigabe noch durch einen grünen Haken, vor der Bezeichnung des Dokuments. Ebenfalls im gleichen Ordner befinden sich die Unterschriftendateien.

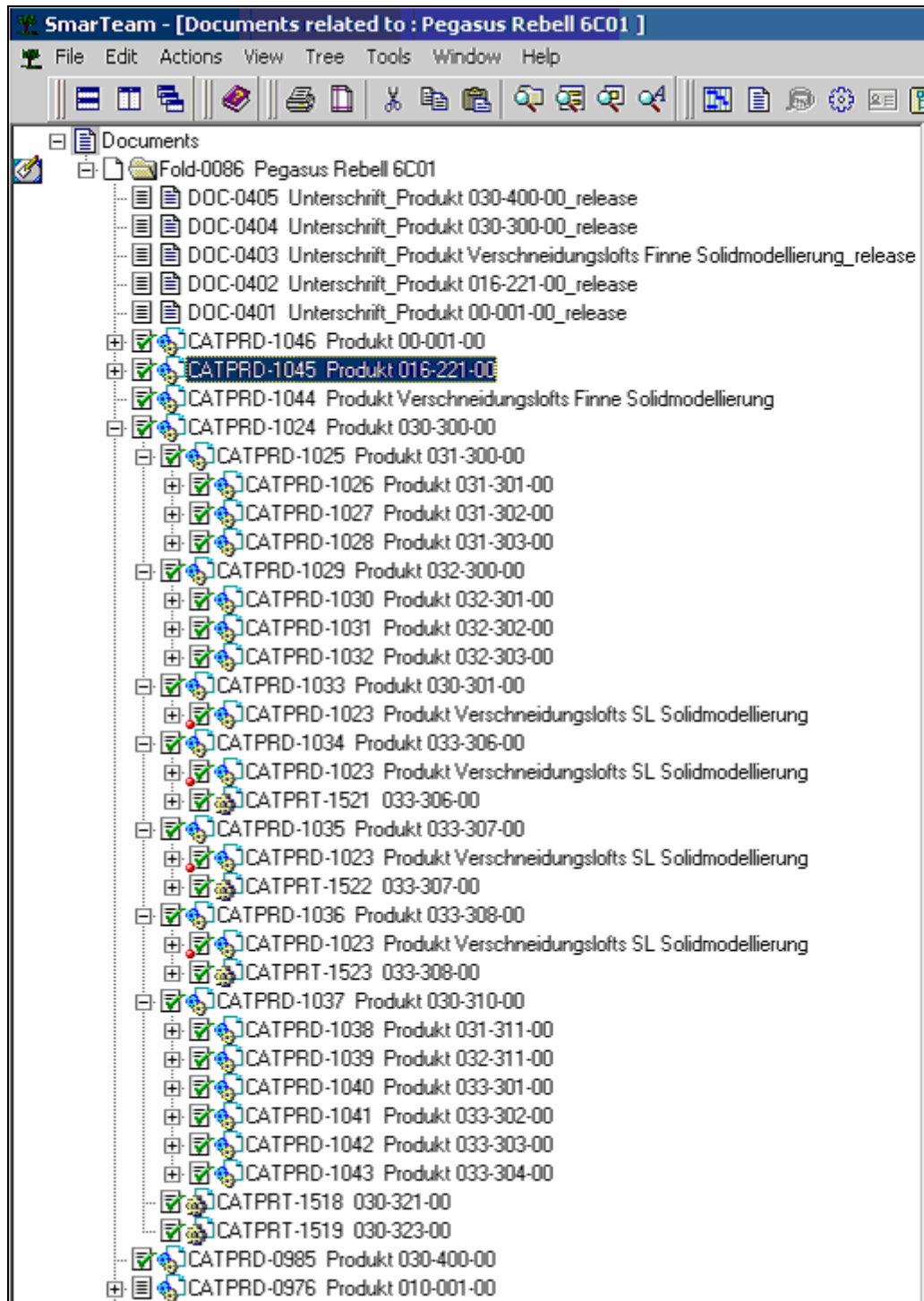


Abbildung 3.11 SmarTeam Baumstruktur mit freigegebenen Objekten und den dazugehörigen Unterschriften

3.4 Die Funktionen „New Release“ und „Obsolete“

Außer den bereits schon mehrfach erwähnten Funktionen „checkin/out“ und „release“ gibt es noch die weiteren Funktionen „New Release“ und „obsolete“.

Die Funktionen „New Release“ ist dazu gedacht, eine bereits freigegebene Datei doch noch einmal zu ändern. Diese Funktion ist mit dem Checkout-Vorgang vergleichbar. Das zu ändernde Dokument wird ausgecheckt und erhält ebenfalls eine neue Revisionsnummer. Allerdings beginnt diese Nummer beim Durchführen der Funktion „New Release“ mit einem „b“ (bei normalem Checkout-Vorgang mit „a“). Das Dokument kann jetzt erneut eingespielt und freigegeben werden.

Die Funktion „obsolete“ dient dazu, Dokumente die nicht mehr benötigt werden, als solche zu kennzeichnen. Die Dokumente können zwar weiterhin noch von den Benutzern angeschaut werden, für die Produktion oder das Unternehmen sind diese Dokumente aber nicht mehr von Bedeutung. Das Dokument ist ungültig und wird als archiviert angesehen.

Zusammenfassung und Ausblick

Zusammenfassend kann gesagt werden, dass das am IFB vorhandene System aus SmarTeam, CATIA sowie Word und Excel zum ersten Mal mit einem System zur elektronischen Unterschriftenerkennung erfolgreich kombiniert wurde. Es wurde gezeigt, dass es möglich ist, ein solches Handschriften Erkennungssystem in einem modernen Unternehmen einzusetzen.

Somit ist es möglich, unterschreibungspflichtige Dokumente direkt in einem elektronischen System zu unterschreiben und zu archivieren, ohne dass das Dokument dazu ausgedruckt werden muss. Dies spart zum Einen Geld und zum Anderen auch Zeit, da ein solches System viel flexibler ist.

Es ist möglich geworden rechtlich anerkannte Unterschriften in Dokumenten aus Word, Excel aber auch in CATIA Zeichnungen einzufügen. Dabei handelt es sich nicht nur um ein Bild wie es bei einer eingescannten Unterschrift der Fall wäre. Das Bild der Unterschrift ist nur ein kleiner Teil des Ganzen. Viel wichtiger sind die ebenfalls aufgezeichneten Daten über die Schreibgeschwindigkeit und die Drucktiefe. Diese Daten werden dem Dokument ebenfalls beigefügt und dienen auch zur Verifizierung des Bildes der Unterschrift, sollten einmal Zweifel an der Echtheit bestehen.

Zusätzlich ist das System in der Lage, Unterschriften bestimmten Personen zuzuordnen, wenn Vergleichsunterschriften vorliegen. Durch vergleichen der Daten der Unterschriften, ist es möglich zu bestimmen, ob die Unterschriften von der selben Person sind. Dadurch ist es möglich schon im Vorfeld festzustellen, ob der Unterschreibende auch derjenige ist, der er vorgibt zu sein. So wird sichergestellt, dass auch nur die dazu berechtigten Personen ein Dokument unterschreiben.

In den nächsten Jahren könnten sich solche Systeme immer mehr in allen Bereichen durchsetzen. Nicht nur in Unternehmen, in denen es zu unterschreibende Dokumente gibt, ist es sinnvoll. Auch für den Privatanwender besteht einiges an Potential. So können z.B. Verträge oder dergleichen per Mail geschickt werden, was deutlich schneller gehen würde. In diesem Bereich bietet sich ein breites Spektrum an Einsatzmöglichkeiten.

Mit dem System für elektronische Unterschriften ist es auch deutlich schwerer geworden eine solche Unterschrift zu fälschen. Das simple Nachfahren einer Unterschrift ist nicht mehr möglich, da hier die Drucktiefe und die Geschwindigkeit nicht nachgemacht werden können.

Obwohl bei der Entwicklung des in dieser Arbeit vorgestellten Systems mit großer Sorgfalt darauf geachtet wurde, dass Fälschungen auszuschließen sind, ging es jedoch im Wesentlichen darum zu zeigen, dass ein solches Zusammenspiel der Komponenten überhaupt möglich ist. Weiterführend könnte noch eine sehr genaue Sicherheitsprüfung stattfinden, sowie ein Installationsalgorithmus entwickelt werden, der selbstständig alle notwendigen Einstellungen vornimmt, so dass es auch von potentiellen Kunden leicht installiert werden kann.

Literaturverzeichnis

- [1] M.Eigener, R.Stelter Produktdatenmanagement-Systeme, Springer Verlag 2001
- [2] Peter Monadjemi, Visual Basic 6 Programmieren mit Windows 9x, NT und 2000 Markt + Technik Verlag 2002
- [3] RRZN Hannover, Visual Basic 6.0 Grundlagen, RRZN Hannover 2003
- [4] RRZN Hannover, Visual Basic 6.0 Fortgeschrittene Programmierung, RRZN Hannover 2001
- [5] Smarteam V5 R14 Documentation, Smarteam Corporation Ltd 2004
- [6] CATIA V5 Release 14 Dokumentation, Dassault Systèmes 2004
- [7] www.hesy.de