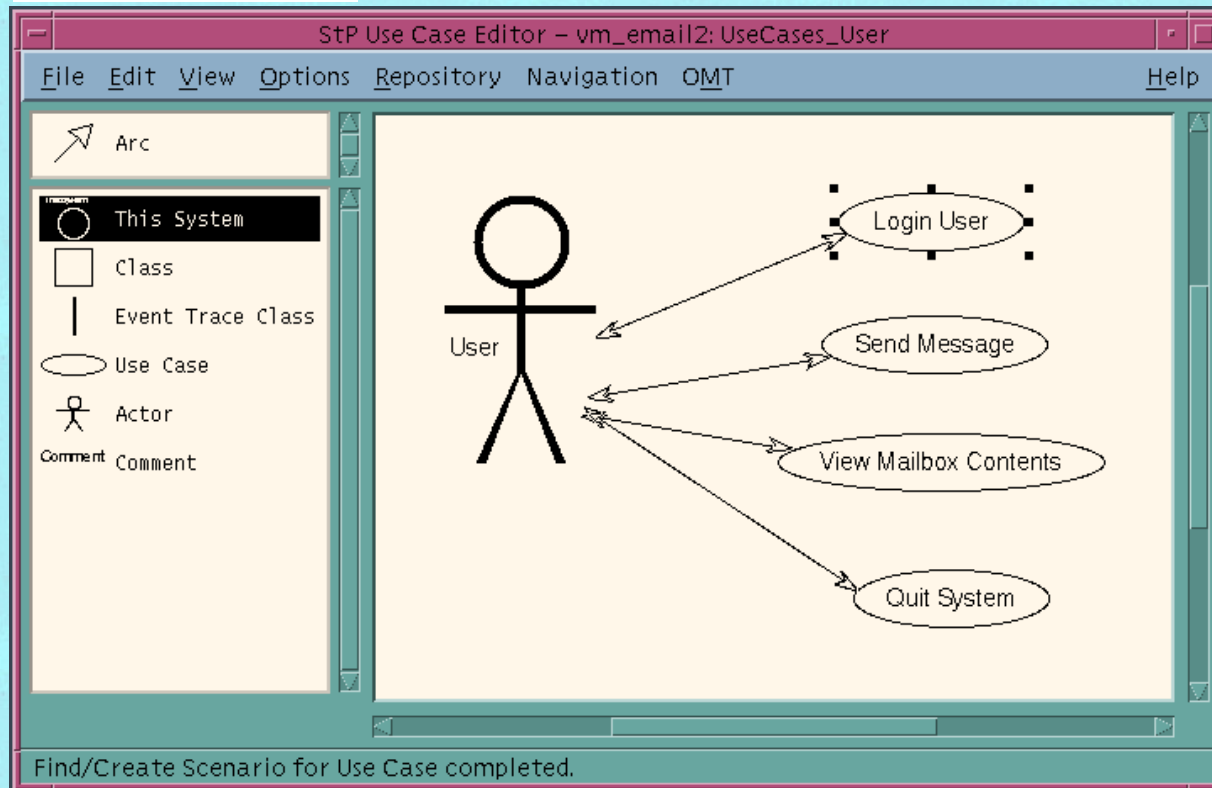


## V-Modell Dokumentengenerierung mit StP

Der weitaus größte Teil des V-Modells bezieht sich auf die konkrete Softwareentwicklung. Seit über zwei Jahren gibt es zu "Software through Pictures (StP)" der Firma AONIX GmbH (früher IDE GmbH) für das Submodell SWE eine Erweiterung, die sowohl im klassischen Modellieren als auch im objektorientierten Modellieren es erlaubt, V-Modell konform zu entwickeln und automatisch die notwendigen V-Modell Produkte, z.B. System-Anforderung, System-Architektur, zu generieren.



Innerhalb der StP-Editoren kann von Darstellungen der Aufteilung von z.B. Segmenten in SWKE's gepushed und gepopped werden, es kann zu den Anforderungen der einzelnen Objekte navigiert werden und die Objekte, bei denen die Anforderungen erfüllt bzw. verfeinert werden, angezeigt werden. Alle Anforderungen werden mit eindeutigen, automatisch erzeugten Schlüssel im Repository abgelegt und können über zusätzliche Reports auf Konsistenz überprüft werden.

## Vorgaben des Vorgehensmodells

Das Vorgehensmodell (V-Modell) regelt die Softwarebearbeitung im Bereich der Bundeswehr/Bundesverwaltung durch die einheitliche und verbindliche Vorgabe von Aktivitäten und Produkten (Ergebnissen), die bei der Softwareerstellung (SWE) und den begleitenden Tätigkeiten für Qualitätssicherung (QS), Konfigurationsmanagement (KM) und technisches Projektmanagement (PM) anfallen.

Die Produkte sind im V-Modell auf Kapitel-Ebene verbindlich geregelt (Anlage 2 des V-Modells vom August 1992). Die Gliederungsschemata zu den einzelnen Produkten sind in der Regel sehr fein strukturiert. Die automatische Dokumentengenerierung hat sich an diese Schemata zu halten.

Das V-Modell, bzw. die Gliederung der Dokumente beruht auf einem funktionalen Designansatz. Ein objektorientierter Ansatz wird in einer überarbeiteten Ausgabe des V-Modells voraussichtlich erst 1997 vorliegen. Eine objektorientierte Entwurfsstrategie ist möglich, solange der vorgegebene Rahmen der Produkte nicht verletzt wird, d.h. die Kapiteleinteilung bleibt, die Inhalte von Funktionsbeschreibungen und Dateninhalten werden zusammen (als Objekte mit Methoden und Daten) behandelt (Details siehe objektorientierte Strategie).

Gegenüber der klassischen Einteilung der Softwareentwicklung in

- Analyse
- Design
- Implementierung
- Test

gliedert das V-Modell das Submodell SWE in Aktivitäten auf die Ebenen, nämlich

- System-Ebene,
- DV-Ebene,
- SW-Ebene und
- Komponenten- und Modul-Ebene.

Entscheidend sind diese Ebenen aus Sicht des Softwareentwicklungsprozesses sowohl in den frühen Aktivitäten, um überhaupt zur Software zu gelangen, als auch in den späten Aktivitäten, um von der Software aus zum Gesamtsystem zu kommen. In jeder dieser Ebene wiederholt sich der Vorgang von Analyse und Design, von Festhalten von Anforderungen, Bestimmen von Funktionen und anschließender Zuordnung zu tieferliegenden Strukturen, der Architektur. Dies zeigt sich auch in den Produkten, so gibt es ein

- System-Anforderungsdokument,
- DV-Anforderungsdokument
- SW-Anforderungsdokument

Die Anforderungen einer Ebene werden in der darunterliegenden verfeinert.

Insgesamt besteht das Submodell SWE aus 9 Hauptaktivitäten

**Hauptaktivitäten SWE**

SWE 1	Systemanforderungsanalyse und -Entwurf
SWE 2	DV-Anforderungsanalyse und -Entwurf
SWE 3	SW-Anforderungsanalyse
SWE 4	Grobentwurf
SWE 5	Feinentwurf
SWE 6	Implementierung
SWE 7	SW-Integration
SWE 8	DV-Integration
SWE 9	System-Integration

**Je nach Aufgabe kann SWE 2 wegfallen, z.B. bei kleinen Softwareentwicklungen besteht die ganze Entwicklung nur aus einer Softwarekonfigurationseinheit (SWKE). SWE 1 ist aber immer durchzuführen.**

## Möglichkeiten von StP

### Erweiterung des Repositoriums

StP erlaubt es, an bestimmte Objekte zusätzliche Informationen anzuhängen, die anschließend in der Datenbasis von StP, dem Repository, zur Verfügung stehen und dort miteinander in Beziehung gesetzt werden können, bzw. durch entsprechende Abfragealgorithmen zu V-Modell konformen Ausgabedaten zusammengetragen werden können. Das Anhängen von Information an Objekte wird im Folgenden als Annotation bezeichnet. Diese Daten werden durch den Object Annotation Editor (oae) eingegeben. Bei der Eingabe können bereits Konsistenzchecks eingebaut werden.

Zusätzlich kann sowohl der regelgesteuerte Grafikeditor als auch der Tabelleneditor erweitert werden. So lassen sich sogenannte Displaymarks einführen, die bestimmte Zustände von Objekten markieren, z.B. Vorhandensein von bestimmten Annotationen.

### Dokumentengenerierung

Das Query and Reporting System erlaubt komfortable Dokumente aus den Daten des Repositoriums und der Grafik- oder Tabellendateien zu erzeugen, dazu wird von StP das DTP-System Framemaker genutzt. Auch Kunden ohne eigenen *Framemaker* können mittels des in StP enthaltenen Frameviewer die Dokumente anschauen und ausdrucken (Postscript). Da Grafiken und Tabellen innerhalb der Dokumente sehr sinnvoll sind, wird die Generierung der V-Modell-Dokumente für den Framemaker durchgeführt. Andere Ausgabeformate sind möglich, z.B. *Interleaf*. Es können Indices, Querverweise oder Hypertextmarken erzeugt werden.

An einigen Stellen innerhalb der zu generierenden Dokumente gibt es größere Textpassagen einzugeben, die nicht unbedingt über StP zu verwalten sind (Dies entlastet auch das Repository von unstrukturierten Daten). Diese Texte können außerhalb von StP editiert werden und während des Dokumentengenerierens integriert werden. Durch Referenzieren dieser Dokumente ( z.B. im Framemaker 5 ) können sie auch in einem nicht textuellen Format vorliegen z.B. in Framemaker-Format.

### Requirement-Tracing

Der Requirement-Editor ist in Core 2.x integriert. Bestimmte Prinzipien und Vorgaben dieses Tabelleneditors zur Definition von Anforderungen sind interessant, jedoch für das V-Modell nicht ausreichend genug. Er wurde hauptsächlich entworfen um das Generieren von DoD 2167A-Reports zu erlauben. Deshalb wird von StP/VM-DOK eine modifizierte Version dieses Editors ausgeliefert.

### Hierarchieabbildung in OMT

Eine Abbildung der V-Modell Hierarchie ist durch Bildung von OMT-Subsystems möglich ( um folgend die doppelte Bedeutung von Subsystem zu unterscheiden wird bei Einsatz von Subsystem als OMT-Begriff grundsätzlich von OMT-Subsystem gesprochen, bei Einsatz in der Nomenklatur des V-Modells nur von Subsystem oder V-Modell Subsystem). Rambaugh [5] spricht auch von vertikaler (Partitions) und horizontaler (Layers) Aufteilung eines Systems. Für beide Aufteilungsrichtungen bieten sich OMT-Subsystems an, die auch hierarchisch sein können. Zusätzlich gibt es die Aufteilung von Klassendiagrammen in Modules, ein Zusammenfassen von eng gekoppelten Klassen. DV-Segmente lassen sich durch OMT-Subsystems, SWKE's durch weitere OMT-Subsystems oder durch Module. Falls vom System gleich auf SWKE-Ebene gegangen wird, bieten sich hier einstufige OMT-Subsystems an. Eine Weiteraufteilung der SWKE's in Komponenten und Module entspricht den OMT- Modules und Sheets. Hier bietet sich auch die Prozedardarstellung von Booch an.

## Benutzungsschnittstelle

Für die Methode OMT wurden folgende Oberflächenelemente von StP erweitert:

- [Annotaion Editor](#)
- [Requirement Editor](#)
- [UseCase Editor](#)
- [Klassen Editor](#)

### Desktop und StP - System

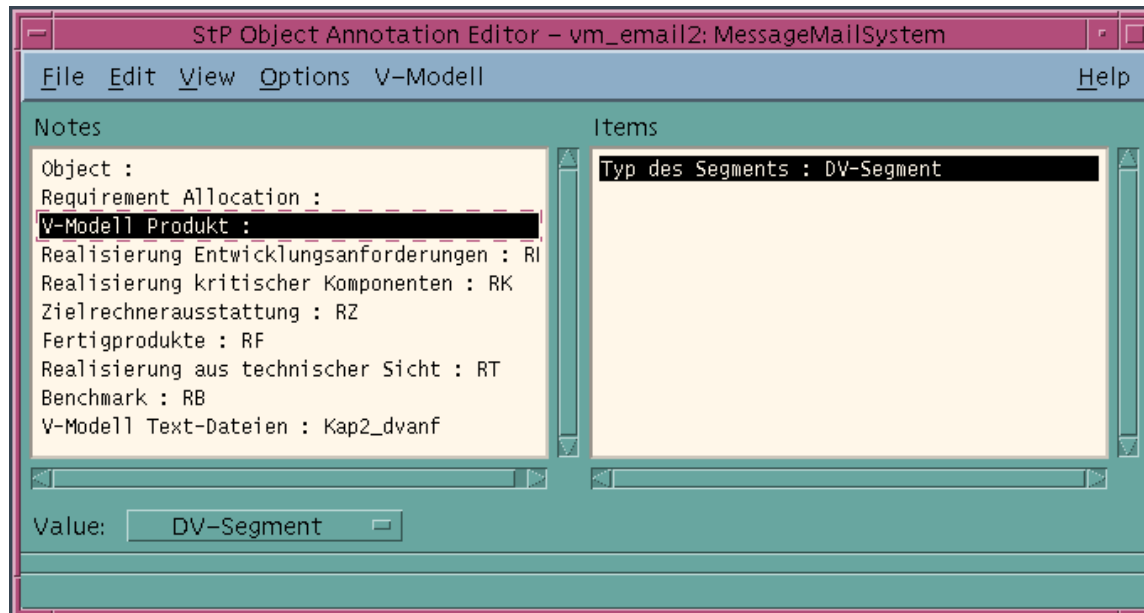
Es wird vorausgesetzt, das pro StP-System genau ein V-Modell System bearbeitet wird. Sollte eine Aufteilung eines V-Modell Systems auf mehrere StP-Systeme erwogen werden, z.B. pro DV-Segment, so muss über bestimmte Import/Export - Funktionen ein Abgleich zwischen den Repositorien geschehen. Eine Eindeutigkeit von Anforderungsschlüsseln ist dann jedoch nicht gewährleistet.

Soll ein System in Subsysteme aufgeteilt werden, so ist jedes Subsystem als neues StP-System zu behandeln. Subsysteme werden im V-Modell wie Systeme behandelt.



## Annotation Editor

Der Editor ist unverändert, je nach Objekt sind zusätzliche Annotationen möglich.



Um eine textuelle Gestaltungsmöglichkeit bestimmter Beschreibungen ( im Description Editor ) im Ausgabedokument zu haben, sind folgende Formatanweisungen im Text der Beschreibung möglich:

- Absatz vom Frametyp Haupttext (Zeilenanfang )
- Absatz vom Typ Strichaufzaehlung (Zeilenanfang )
- Absatz vom Typ Strichaufzaehlung1 (Zeilenanfang )
- Das nächste Wort wird fett gedruckt
- Das nächste Wort wird kursiv gedruckt
- Das nächste Wort wird fett und kursiv gedruckt

Längere Texte werden öfters in externen Dateien gehalten. Dieser Text belastet dann nicht das Repository, sondern ist über eine Dateireferenz im Repository abgelegt.

Dateiimporte werden über die Annotation "V-Modell Text-Dateien" mit dem Item Dateiname als Name ohne Pfad und dem Item Dateityp als Aufzähltyp an bestimmte Objekte definiert. Der absolute Dateipfad wird aus dem aktuellen Projektverzeichnis, dem System, dem Subdirectory vm\_files/ und dem Item *Dateiname* zusammengesetzt.

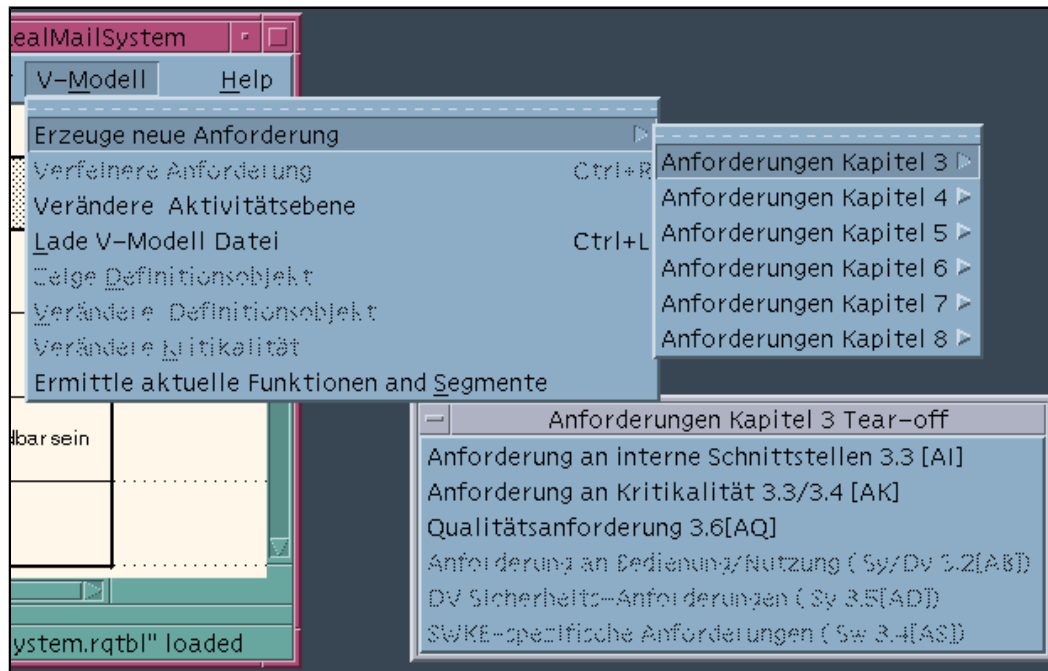
In diesen Dateien sind obige Formatanweisungen auch erlaubt.

## Requirement Editor

Der Requirement-Editor wurde gegenüber dem Original StP-Requirementeditor in wesentlichen Punkten geändert.

### Pulldown-Menü 'V-Modell'

Alle zusätzlichen Menüpunkte sind unter einem neuen Hauptmenu zusammengefaßt:



Bei diesem Beispiel wird sichtbar, daß nicht alle Menüpunkte der Anforderungen für Kapitel 3 erlaubt sind. Dies ist abhängig vom jeweiligen Objekttyp, dessen Anforderungen gerade definiert werden ([Beschreibung der Funktionen](#)).

### Tabellenzeile 2

Die Tabellenzeile 2 ist immer sichtbar und zeigt zum Einen den Namen des aktuellen Definitionsobjekts, d.h. den Namen des Objekts, dessen Anforderungen angezeigt werden, und zum Anderen die aktuell eingestellte Aktivitätsebene.

**Aktivitätsebenen**

Kürzel für Aktivitätsebene	Aktivitätsebene
Sy	System-Ebene
Dv	DV-Segment-Ebene
Sw	SWKE-Ebene
(leer)	Komponenten- und Modul-Ebene

**Tabellenspalte 'Anforderung'**

Im Requirement-Editor ist das Feld für den Namen der Anforderung vom Benutzer nicht direkt editierbar, da alle Anforderungen mit einem einheitlichen Schlüssel versehen werden. Er setzt sich zusammen aus

- der Aktivitätsebene, in der die Anforderung definiert wird: Sy,Dv,Sw oder leer
- der Klassifizierung der Anforderung, z.B. SEU für SW-Entwicklungsumgebung
- einer laufenden vierstelligen Nummer zur weiteren Unterscheidung

Durch Einstellung der Aktivitätsebene sowie durch Auswahl der Klassifizierungskennung gibt der Benutzer den Präfix des Schlüssels vor. Der Requirement-Editor sucht die nächste freie laufende Nummer dieses Präfixes und trägt den Schlüssel selbst in die Tabelle ein. Somit wird gewährleistet, daß der Schlüssel innerhalb eines StP-Systems eindeutig ist. Diese Schlüssel werden bei der Dokumentengenerierung ausgewertet.Folgende Klassifizierungen der Anforderungen ist möglich:

**Anforderungsklassifizierung**

für Kapitel der Anforderungsdokumente	Klassifizierung	Beschreibung
3.2	AB	Bedienung/Nutzung
3.3	AI	Realisierung interner Schnittstellen
3.4	AK	Kritikalität des Segments
3.5	AD	DV-Sicherheit
3.6	AQ	sonstige Qualitätsmerkmale
4	SEU	Entwicklungs- und SWPÄ-Umgebung
5.1.1	ET	Technische Einsatzumgebung
5.1.2	EO	Organisatorische Einbettung
5.1.3	EN	Nutzung
5.2.2.s.1	EV	Verwendung der Schnittstelle
5.2.2.s.2	EA	Anforderungen an die Schnittstelle
6.3.f.1.1	FN	Normalfall
6.3.f.1.2	FA	Ausnahmefall
6.3.f.1.3	FIF	Informationsfluß (allgemein)



6.3.f.1.3.1	FIE	Eingabeinformationen
6.3.f.1.3.2	FIA	Ausgabeinformationen
6.3.f.1.3.3	FII	Interne Informationen
6.3.f.2	FQE	Anforderungen an Effizienz
6.3.f.3	FQZ	Anforderungen an Zuverlässigkeit
6.3.f.4	FQK	Kritikalität der Funktion
6.3.f.5	FQB	Anforderung an die Benutzbarkeit
6.3.f.6	FQA	Anforderung an die Änderbarkeit
6.3.f.7	FQU	Anforderung an die Übertragbarkeit
7.2	DI	Daten - Inhalt
7.2	DA	Daten - Aufbau
7.2	DM	Daten Mengen
7.2	DQ	Daten Qualität
8	AFV	Funktionsabläufe/Verfahren (allgemein)

In der Schlüsselspalte werden noch weitere Informationen abgespeichert. So wird an allen Schlüsseln der Klassifizierung 'FQK' die Kritikalitätsstufe mit abgelegt. Folgende Stufen sind selektierbar:

<b>Kritikalität</b>	<b>Bedeutung laut V-Modell bei administrativen Informationssystemen</b>
keine	Fehlverhalten beeinträchtigt die zugesicherte Eigenschaft nicht wesentlich
niedrig	Fehlverhalten verhindert Zugang zu Informationen, die regelmäßig benötigt werden
mittel	(Fehlverhalten macht insensitive Daten für unberechtigte Personen zugänglich oder verfälscht abgeleitete Daten)
hoch	Fehlverhalten macht sensitive Daten für unberechtigte Personen zugänglich oder verhindert administrative Vorgänge oder führt zu Fehlentscheidungen infolge fehlerhafter Daten

Bei verfeinerten Anforderungen wird das Definitionsobjekt mit in der Schlüsselspalte in der zweiten Zeile vermerkt.

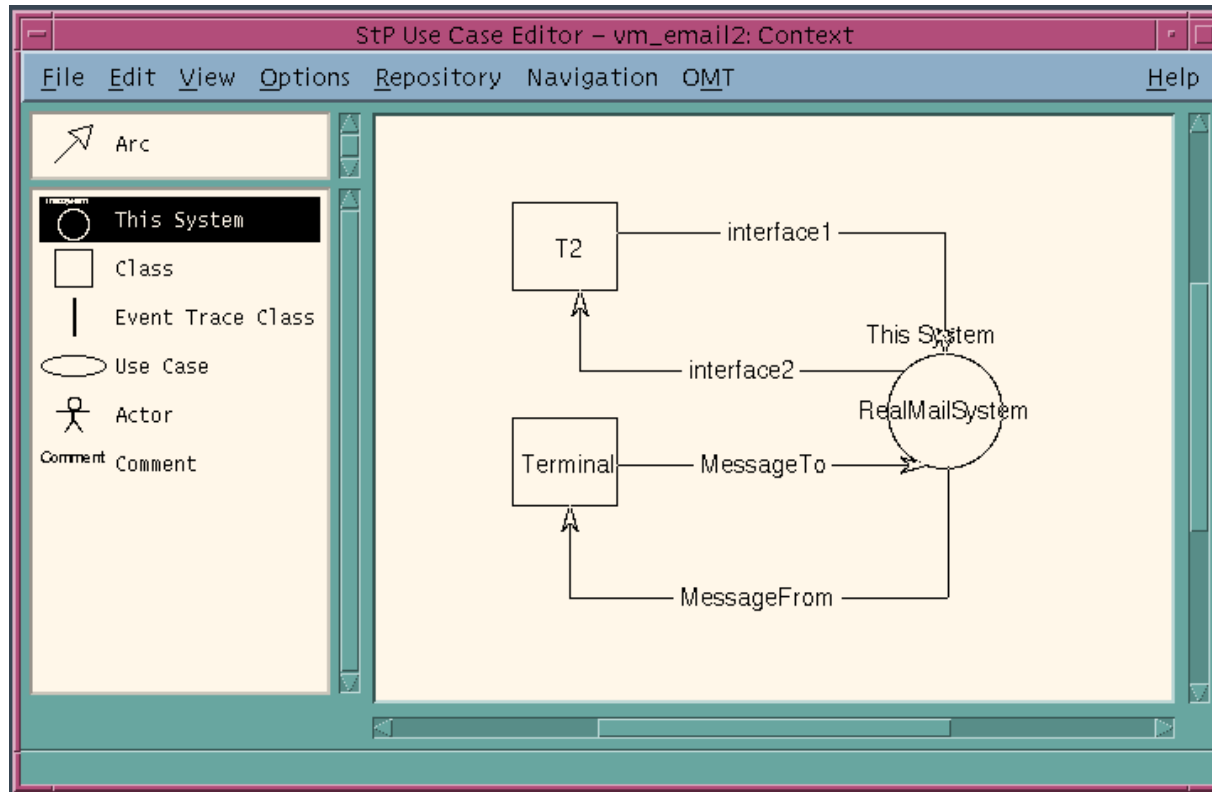
### Tabellenspalte 'Definition'

An dieser Spalte wurde keine Änderung durchgeführt. Hier wird die Beschreibung der Anforderung eingegeben. Um eine textuelle Gestaltungsmöglichkeit im Ausgabedokument zu haben, sind die gleichen Formatanweisungen wie im [Annotationseditor](#) oder in zu importierenden Textdateien erlaubt.

## UseCase Editor

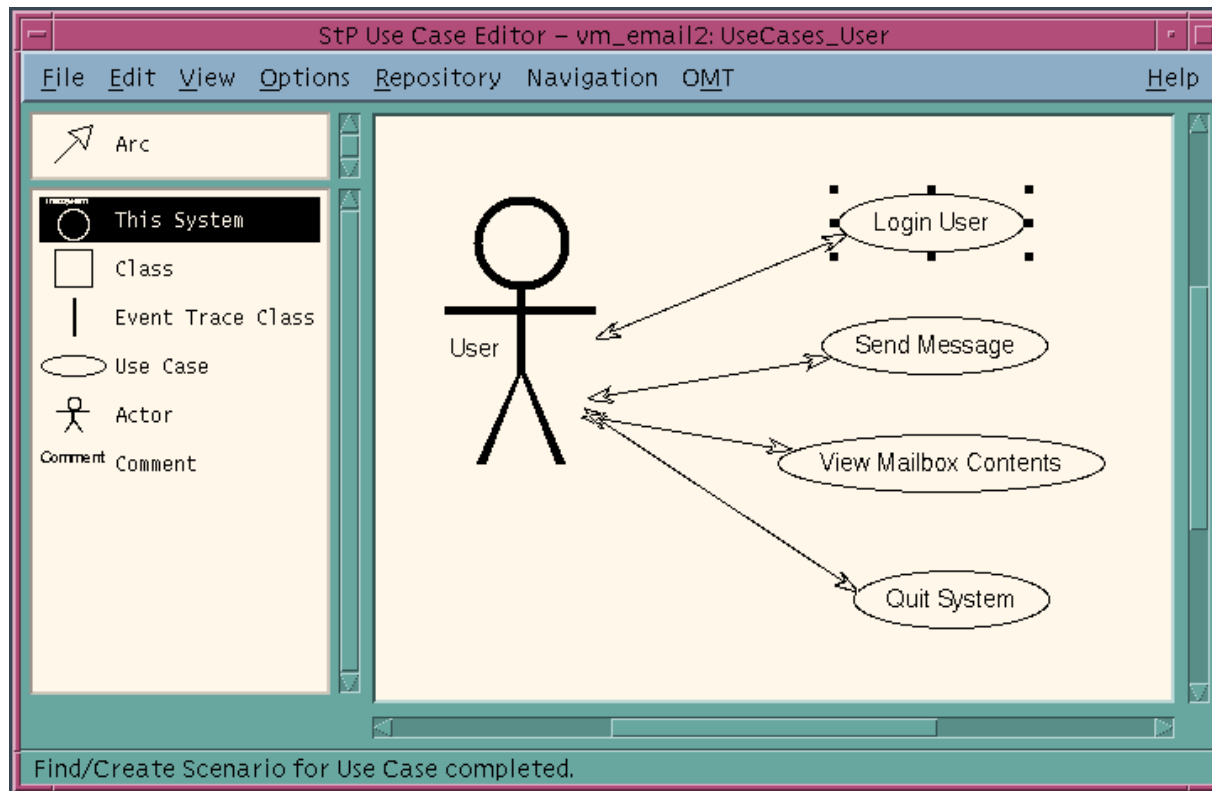
### Kontext-Diagramm

Das Kontextdiagramm '**Context**' (OMTdiagramm mit System und Datenflüssen von/zu Aktoren) wird als Verankerung der allgemeinen Anforderungen des Systems genutzt. Die Anforderungen werden durch Navigation zu einer oder mehreren (z.B. pro Anforderungstyp) Requirement-Tabelle eingetragen. Am Knoten 'ThisSystem' wird über die Beschreibung an der Annotation **Note Object** das System kurz beschrieben. Über die gleiche Annotation werden die Externals und die Datenfüsse (externen Schnittstellen) beschrieben.



### Use Cases Diagramm

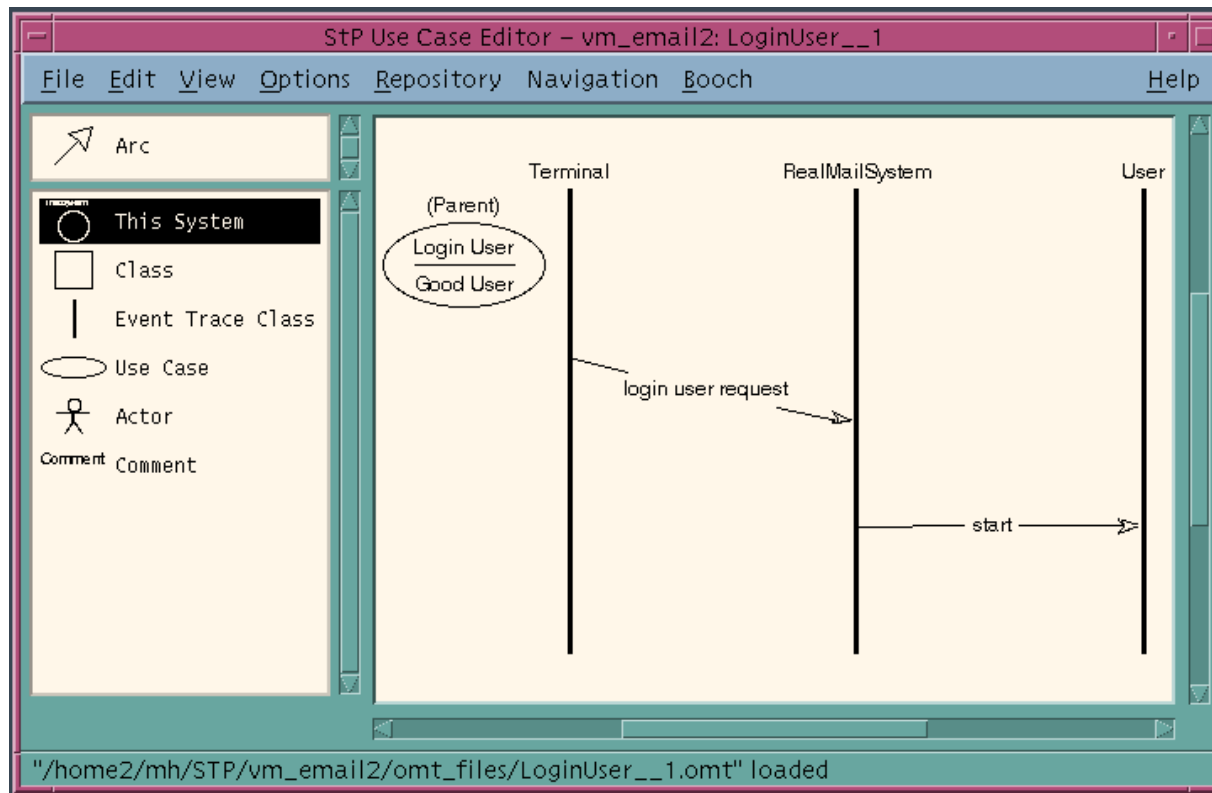
Alle externen Systeme, seien es externe Programme, Menschen, Maschinen, die mit dem System interagieren werden als **Actors** dargestellt und alle Szenarien (UseCase), die mit einem oder mehreren Actor zusammenhängen, werden als Funktionen aufgefaßt.



An jeden **UseCase** werden über Navigation zu einer Requirement-Tabelle die funktionsorientierten Anforderungen definiert.

## Event Traces

Alle Szenarien (Event-Traces) zu den einzelnen **UseCases** werden dargestellt und beschrieben. Dazu muß jedes darzustellende Szenario mit einer Annotation V-Modell Produkt:Dokumentiert in: ausgestattet werden. Bei größeren Systemen ( die z.B. eine Unterteilung in DV-Segmente haben ), bietet es sich an, auf Systemebene nur alle Normalfälle in den Szenarien zu betrachten, Fehlerfälle werden dann erst in den Funktionsabläufen der DV-Segmente oder SWKE dargestellt.



## Neue Menüpunkte im Pulldown-Menü 'Navigations'

### Knopf 'See Requirements'

Über diesen Knopf wird zu einer Requirementdatei navigiert, in dem Anforderungen des gerade selektierten Objekts stehen. Gibt es mehrere Requirement-Dateien, die mit dem Namen des selektierten Objekts beginnen, wird dem Benutzer eine Auswahl angeboten. Existiert noch keine Datei, so wird eine angelegt.

### Knopf 'See Functions of System'

Dieser Knopf ist bei dem Objekt ThisSystem aktiviert. Über diesen kann zu einem der UseCase-Dateien 'UseCases\*' zu navigiert werden. Existieren mehrere Dateien, wird eine Auswahlliste angeboten. Existiert keine, so wird eine mit Namen 'UseCases' angelegt.

### Knopf 'Push Subsystem'

Vom Objekt ThisSystem, d.h. aus dem Kontextdiagramm, kann zum Dekompositionsdiagramm (Klassendiagramm mit Aggregationen) des Systems navigiert werden.

### Knopf 'See Subsystem'

Aus UseCase Diagrammen, d.h. bei selektiertem UseCase, kann über diesen Knopf zum entsprechenden Dekompositionsdiagramm navigiert werden.



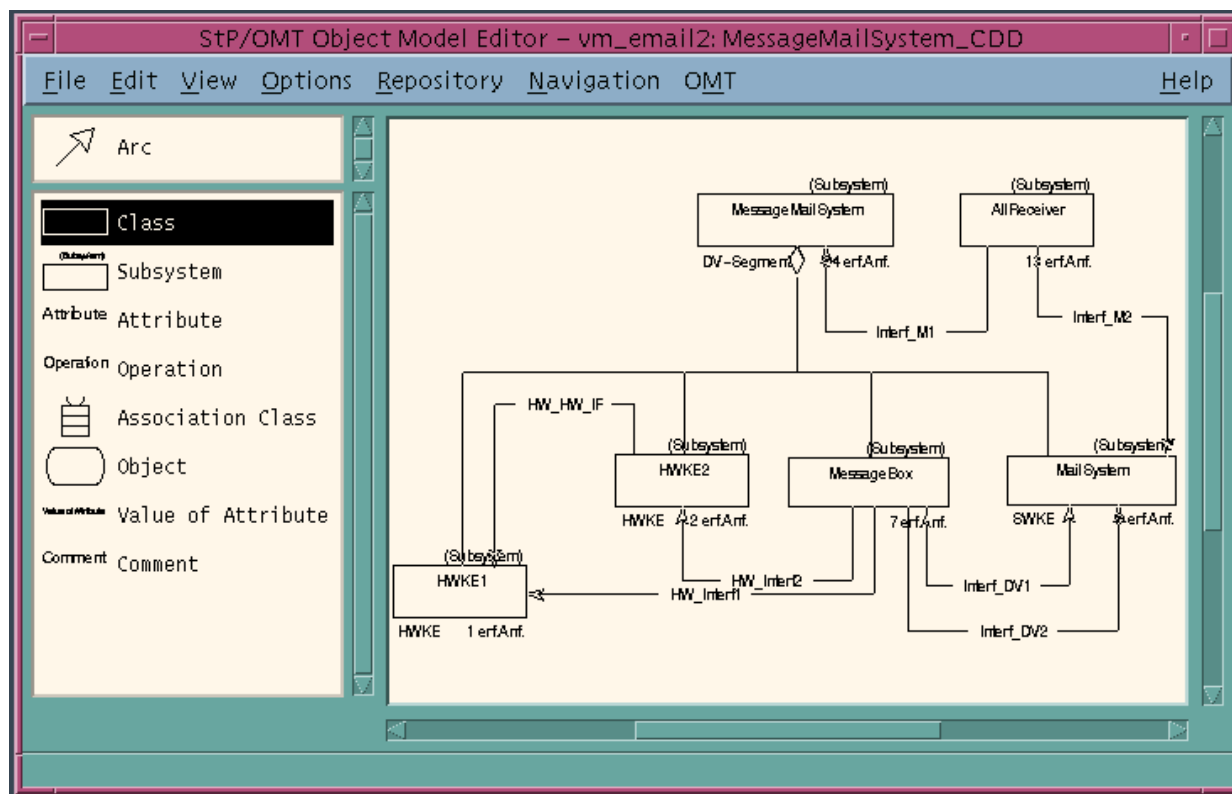
## Klassen Editor

### Neue 'Display Marks'

An OMT-Subsystemen wird angezeigt, welcher Typ von V-Modell Segment sie darstellen und es wird angezeigt, wieviele Anforderungen von Ihnen jeweils erfüllt werden.

### Dekompositionsdiagramme

Die Aufteilung eines Segments (z.B. DV-Segment) in Untersegmente (z.B. SWKE's) wird in einem speziellen Klassendiagramm dargestellt. Der Name des Diagramms setzt sich zusammen aus dem Namen des Segments und dem Suffix '\_CDD', damit es dem **Large System Support for StP/OMT, Dez. 1995** entspricht.



In diesem *Category Decomposition Diagram* wird über eine Aggregation die Aufteilung des Segments beschrieben. Alle Segmente werden als StP-Subsystems angelegt und erhalten über die Annotation **Note 'V-Modell Produkt'** mit Aufzähltyp **Item 'Typ des Segments'** einen eindeutigen Segmenttyp zugeordnet.

### mögliche Segmenttypen

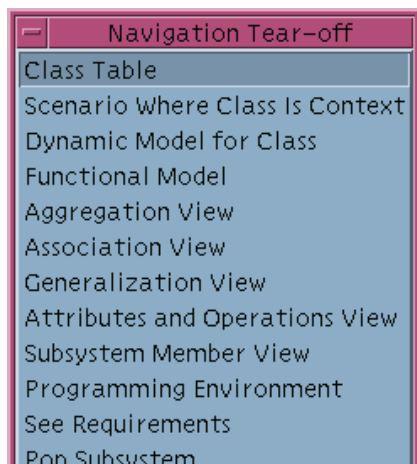
Aufzähltyp	Bedeutung
System	Gesamtsystem
Subsystem	Bei größeren Systemen, die sich in mehrere meist örtlich getrennte Anteile mit jeweils DV- und Nicht-DV-Anteilen aufteilen lassen, werden Subsysteme gebildet.
DV-Segment	DV-Anteil eines Systems. Bei reinen Softwaresystemen kann diese Aufteilungsstufe ausgelassen werden (Tailoring).
SWKE	Software-Konfigurationseinheit. Dies kann ein Programmpaket sein ( z.B. RDBMS ), eine Laufzeitbibliothek o.ä., d.h. meist ein elementar zu verwaltes Objekt innerhalb des Zielsystems.
HWKE	Hardware-Konfigurationseinheit. Wird nicht weiter aufgeteilt.
Komponente	Eine SWKE kann in der Entwicklungsumgebung hierarchisch in Komponenten und Module gegliedert werden. Eine Komponente kann weitere Komponenten und Module nutzen.
Modul	Ein Modul ist die kleinste Softwareeinheit, z.B. ein Kompilierungsmodul, die von der Konfigurationsverwaltung noch erfaßt wird, d.h. meist auf Dateiebene.
Nicht-DV-Segment	Bestimmte allgemeine Anforderungen beziehen sich auch auf Nicht-DV Anteile ( z.B. Sicherheitsschleuse vor Rechnerraum). Sie dienen hier nur zur Anforderungszuordnung, werden danach jedoch nicht weiter behandelt.

Die Schnittstellen werden über gerichtete Assoziationslinks vom Client zum Server gezeichnet, d.h. der Pfeil zeigt zum Server. Durch gleichzeitige Selektion von Link und Server und Auswahl des Knopfs '**Toggle Role Traversal**' unter dem Menü '**Assoc**' wird der gerichtete Assoziationslink angelegt. Der Name des Links ist der Name der Schnittstelle. Die Beschreibung der Schnittstelle wird an die Role geknüpft, die zum Server-Subsystem zeigt.

Wird die Schnittstelle mehrfach genutzt, genügt die Beschreibung der Schnittstelle an einer der **Role**'s. Sind mehrere Beschreibungen vorhanden, so werden sie alle zur Dokumentengenerierung genutzt.

Alle Anforderungen an die Schnittstelle sind im Requirementeditor zu definieren. Hier obliegt es dem Benutzer, die Anforderungen entsprechend einer mehrfachen Nutzung der Schnittstelle auf entsprechend mehrere Requirement-Dateien aufzuteilen.

### Neue Menüpunkte im Pulldown-Menü 'Navigations'



#### Knopf 'See Requirements'

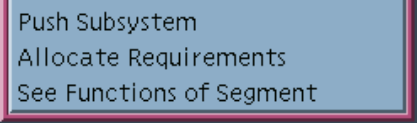
Über diesen Knopf wird zu einer Requirementdatei navigiert. Er ist aktiv bei einer selektierter Assoziation und einem StP-Subsystem. Das Verhalten ist das gleiche wie bei gleichnamigen Knopf im [UseCase-Editor](#).

#### Knopf 'See Functions of Segment'

Dieser Knopf ist bei einem StP-Subsystem aktiviert. Über diesen kann zu einem der UseCase-Dateien '\*UseCases\*' zu navigiert werden. Existieren mehrere Dateien, wird eine Auswahlliste angeboten. Existiert keine, so wird eine mit Namen '<SegmentName>\_UseCases' angelegt.

#### Knopf 'Push Subsystem'

Vom einem selektierten Segment, sprich StP-Subsystem, kann zum Dekompositionsdiagramm ( Klassendiagramm mit Aggregationen ) dieses



Push Subsystem  
Allocate Requirements  
See Functions of Segment

Segments navigiert werden. Existiert noch keine Datei, so wird sie angelegt und mit dem StP-Subsystem automatisch gefüllt.

**Knopf 'Pop Subsystem'**

Von einem selektierten Segment kann zum übergeordneten Dekompositionsdiagramm navigiert werden. Ist das selektierte Segment das System, so wird in das Kontextdiagramm navigiert.



**Knopf 'Erzeuge neue Anforderung'**

Dies ist ein zweistufiges Untermenü, die erste Stufe unterteilt sich in die Kapitel 1 bis 8 des Anforderungsdokuments, die zweite Stufe gibt die möglichen [Anforderungstypen](#) an einschließlich der Abschnittsnummer des Anforderungsdokuments.

Je nach aktuellem Definitionsobjekt werden entsprechende Auswahlknöpfe gesperrt oder freigegeben.

**Knopf 'Verfeinere Anforderung'**

Darunter verbirgt sich das übliche **'Refine'**. Die Funktionalität ist jedoch erweitert, d.h. es wird das jeweilige neue Definitionsobjekt abgefragt, zu dem die verfeinerte Anforderung gehört. Das ist nicht mit einer Zuordnung (Assignment) der Anforderung zu verwechseln. Normalerweise ist die verfeinerte Anforderung an dem Objekt definiert, in dem die übergeordnete Anforderung erfüllt wurde.

Es ist jedoch nicht notwendig, daß die übergeordnete Anforderung bereits zugeordnet ist. Die Zuordnung geschieht implizit durch diese Verfeinerung. Das neue Definitionsobjekt wird dann als Zuordnungsobjekt der übergeordneten Anforderung angenommen.

**Knopf 'Verändere Aktivitätsebene'**

Hiermit läßt sich die Aktivitätsebene umschalten. Normalerweise wird diese vom Requirement-Editor beim Laden einer Datei automatisch ermittelt. Will man jedoch in einer Datei, in der bereits Verfeinerungen durchgeführt worden sind, neue und keine abgeleiteten Anforderungen einfügen, so kann hiermit die Aktivitätsebene eingestellt werden.

**Knopf 'Lade V-Modell Datei'**

Hiermit kann strukturiert auf Requirement-Dateien zugegriffen werden. Zuerst wird durch die Abfrage der Aktivitätsebene über ein Menü die Zahl der möglichen Dateien reduziert. In einem zweiten Menü werden alle bisher existierenden Segmente, Schnittstellen oder Funktionen einer Aktivitätsebene angezeigt. Nach Auswahl einer Zeile werden alle zu dem ausgewählten Objekt gehörenden ( d.h. mit dem Prefix des Objektnamen versehenen ) Dateien zur Auswahl angezeigt. Zusätzlich erscheint noch eine Zeile mit dem Inhalt '--NEU--'. Bei Auswahl dieser Zeile wird eine neue Datei angelegt. Der Name der Datei setzt sich aus dem Namen des Objekts und einer laufenden Nummer verbunden durch ein '\_' .

**Knopf 'Zeige Definitionsobjekt'**

Hiermit wird eine Navigation zum Definitionsobjekt ausgelöst:

- Ist das Definitionsobjekt 'ThisSystem' oder eine externe Schnittstelle, so wird in das Kontextdiagramm navigiert.
- Ist das Definitionsobjekt eine Funktion, wird zu einem UseCase-Diagramm navigiert. Ist das Objekt in mehreren Diagrammen vorhanden, so wird dem benutzer ein Auswahlmenü mit diesen Diagrammen angezeigt.
- Ist es eine interne Schnittstelle oder ein Segment des Systems, so wird in das entsprechende Klassendiagramm 'Subsystem\*' navigiert.

**Knopf 'Verändere Definitionsobjekt'**

Dieser Knopf ist nur aktiv, wenn eine verfeinerte Anforderungszelle in der Tabelle selektiert ist. Über ein Auswahlmenü kann ein neues Definitionsobjekt ausgesucht werden, das dann in der zweiten Zeile der Anforderungszelle unter den Anforderungsschlüssel eingetragen wird.

**Knopf 'Verändere Kritikalität'**

Dieser Knopf ist nur freigegeben, wenn eine Anforderungszelle mit dem Type **xxFQKxxxx** ausgewählt ist. Bei Betätigen erscheint ein Auswahlmü, mittels dem die Kritikalität einstellbar ist.

**Knopf 'Ermittle aktuelle Funktionen und Segmente'**

Der Requirement-Editor liest aus Performance-Gründen nur einmal beim Start Daten über vorhandene Segmente, Funktionen und Schnittstellen aus dem Repository. Der Benutzer muß ggf. bei Veränderungen in Klassen- oder Usecase-Diagrammen diesen Knopf betätigen, damit die aktuellen Daten aus dem Repository gelesen werden.

**modifizierte Funktionalität****Knopf 'Create'**

Dieser Knopf ist umbenannt in **'Create Requirement with same type'**. Bei Betätigen des Knopfes wird eine neue Zeile erzeugt, deren Anforderungsschlüssel das gleiche Prefix wie die zuvor erzeugte Anforderungszeile hat. Deshalb ist der Knopf nach dem Start des Requirement-Editors bis nach dem Erzeugen einer Anforderung gesperrt.

**Knopf 'Import Requirements from ASCII File'**

Die Funktionalität wurde derart erweitert, daß Anforderungsschlüssel automatisch numeriert werden.

Dazu muß nur der Schlüsseltyp mit einem '+' als Prefix in der ASCII-Datei stehen, z.B. '+FN'. Es können auch verfeinerte Anforderungen eingelesen werden, es wird dann automatisch ein Schlüssel der nächsten Aktivitätsebene generiert. Der jeweils erste Schlüssel kann optional auch ein Kürzel direkt nach dem '+' für die Aktivitätsebene beinhalten. Dies übersteuert für diese Zeile die eingestellte Aktivitätsebene des Editors. Das Beispiel '+SyFA%+FA%+FA' erzeugt folgende Zeile:

Requirement			Definition
SyFA0002	DyFA0001	SwFA0002	3. desc

**Knopf 'Refine'**

Dieser Knopf wurde entfernt und liegt nun im V-Modell Menü.