

# Break The Glass – User Manual

## Verwendete Konstrukte:

`x = "list($y)"`      =>    z.B. `x = "u,v,w,z"`  
`x = „list(($y,$z))` =>    z.B. `x = „(u,v),(a,b)“`  
`x = „y{,z}“`          =>    `z` ist optional

## 1. Break The Glass (BTG)

<code>&lt;&lt;BTG:</code>	Beginn einer Annotation	Pflicht
<code>&gt;&gt;</code>	Ende einer Annotation	Pflicht
<code>objects = "list(\$objectname)"</code>	Liste aller Ziel-Datenobjekte	Pflicht
<code>rights = "\$right-type"</code>	Rechte für alle Ziel-Datenobjekte	Pflicht
<code>read, write, update</code>	<i>siehe 4.</i>	
<code>BTGAccessor = "\$name"</code>	Kennung des Accessors; Rollen- oder Nutzernamen	optional
<code>AuthnBTGAccessor-attr = "list((attribute,value)), {idp}"</code>	Authentifizierung des Accessors über Attribut-Wert-Paare; benötigt Spezifikation von <code>BTGAccessor</code>	optional
<code>idp = "\$idp-address"</code>	Spezifizierung eines Identitäts-Providers; benötigt Spezifikation von <code>AuthnBTGAccessor-attr</code>	optional
<code>BTGActivator = "\$name"</code>	Kennung des Activators; Rollen- oder Nutzernamen	optional
<code>AuthnBTGActivator-attr = "list(attribute,value), {idp}"</code>	Authentifizierung des Activators über Attribut-Wert-Paare; benötigt Spezifikation von <code>BTGActivator</code>	optional
<code>idp = "\$idp-address"</code>	Spezifizierung eines Identitäts-Providers; benötigt Spezifikation von <code>AuthnBTGActivator-attr</code>	optional
<code>Start = "\$BPCC"</code>	Defintion temporaler Vorbedingungen; ausformuliert mittels eines BPCC	optional
<code>Exec = "\$BPCC"</code>	Defintion konditionaler Vorbedingungen; ausformuliert mittels eines BPCC	optional
<code>Obligations = "list(\$obligation-ID)"</code>	Auflistung aller resultierenden Verpflichtungen	optional
<code>Insert = "\$insertMode"</code>	Einfügemodus	optional
<code>seq, par</code>	(exklusiv)	

## 2. Obligations (OG)

<code>&lt;&lt;Obligation:</code>	Beginn einer Verpflichtung	Pflicht
<code>&gt;&gt;</code>	Ende einer Verpflichtung	Pflicht
<code>id = "\$obligation-ID"</code>	Liste aller Ziel-Datenobjekte	Pflicht
<code>pattern = "\$obligationpattern"</code>	Rechte für alle Ziel-Datenobjekte	Pflicht
<code>OGParameter = "list((name,value))"</code>	Auflistung der Parameter des Patterns	optional
<code>OGCompensator = "\$name"</code>	Kennung des Compensators; Rollen- oder Nutzernamen; nur benötigt, falls Menschen involviert	optional
<code>AuthnOGCompensator-attr = "list(attribute,value), {idp}"</code>	Authentifizierung des Compensators über Attribut-Wert-Paare; benötigt Spezifikation von <code>CompRole</code>	optional
<code>idp = "\$idp-address"</code>	Spezifizierung eines Identitäts-Providers; benötigt Spezifikation von <code>AuthnCompensator-attr</code>	optional
<code>Start = "\$BPCC"</code>	Defintion temporaler Vorbedingungen; ausformuliert mittels eines BPCC	optional
<code>Exec = "\$BPCC"</code>	Defintion konditionaler Vorbedingungen; ausformuliert mittels eines BPCC	optional

### 2.1 Obligation-Patterns

<code>SendEmail</code>	Parameter: <code>from, to, subject, body, attachment</code>
<code>AuditAccess</code>	Parameter: <code>auditpolicy, start, end</code>

### 3. Business Process Context Constraints (BPCC)

Verwendung innerhalb von BTG und Obligations.

#### 3.1 Funktionen

Aufbau: Funktion (Argument{, Zahl}), wobei das Argument entweder Datenobjekt oder Aktivität ist. Die optionale Angabe einer Zahl gibt an, wie viele Ausführungen bzw. Datenzugriffe betrachtet werden (Standard: 1).

Angabe einer Funktion als Argument einer anderen Funktion ist zulässig.

Funktionsname	Argumenttyp	Rückgabewert
data-user	Datenobjekt{, Recht}	Individuum, das auf dieses Datenobjekt zugreift
owner	Datenobjekt	Betroffener, dessen persönliche Daten im Datenobjekt stehen
performer	Aktivität	Individuum, das diese Aufgabe (Aktivität) ausführt
start-time	Datenobjekt{, Recht}	Zeitpunkt, zu dem Zugriff auf dieses Datenobjekt beginnt
	Aktivität	Zeitpunkt, zu dem Ausführung dieser Aktivität beginnt
end-time	Datenobjekt{, Recht}	Zeitpunkt, zu dem Zugriff auf dieses Datenobjekt beendet ist
	Aktivität	Zeitpunkt, zu dem Ausführung dieser Aktivität beendet ist
data-object	Aktivität{, Recht}	Alle Datenobjekte, auf die während dieser Aktivität zugegriffen wird
tasks	Rolle	Aktivitäten, die von dieser Rolle bereits ausgeführt wurden
	Individuum	Aktivitäten, die von diesem Individuum bereits ausgeführt wurden
duration	Aktivität	Zeit, die eine Aktivität bereits aktiv ist bzw. gedauert hatte
frequency	Datenobjekt{, Recht}	Anzahl der Nutzungen des Datenobjekts
	{, Group}	zusätzliche Gruppierung der betrachteten Aktivitäten
fulfilled	Gateway, Bedingung	„true“, falls die genannte Bedingung bei Gateway eingetroffen ist
	Ereignis	„true“, falls ein Ereignis eingetreten ist
	Nachricht	„true“, falls Nachrichtenfluss ankommt
executed	list(Aktivität)	„true“, falls Aktivitäten ausgeführt wurden
owned-objects	Individuum	Datenobjekte, die persönliche Daten des Betroffenen enthalten
used-objects	Individuum{, Recht}	Datenobjekte, auf die das Individuum zugegriffen hat
role	Aktivität	Rolle, die die Aktivität ausführt

#### 3.2 Funktions-Operatoren

Prüfung von Funktionsrückgabewerten

>	Korrekt, falls Rückgabewert einer Funktion größer als gegebener Wert ist
<	Korrekt, falls Rückgabewert einer Funktion kleiner als gegebener Wert ist
>=	Korrekt, falls Rückgabewert einer Funktion gleich groß oder größer als gegebener Wert ist
<=	Korrekt, falls Rückgabewert einer Funktion gleich groß oder kleiner als gegebener Wert ist
==	Korrekt, falls Rückgabewert einer Funktion gleich einem gegebenen Wert ist
≠	Korrekt, falls Rückgabewert einer Funktion ungleich einem gegebenen Wert ist
∈	Korrekt, falls Rückgabewert einer Funktion Element der gegebenen Menge ist
∉	Korrekt, falls Rückgabewert einer Funktion kein Element der gegebenen Menge ist

#### 3.3 Komplexe Operatoren

Logische Verknüpfung von Funktionen (sowie Funktionen mit Funktions-Operatoren)

∧	UND-Verknüpfung zweier Funktionen
∨	ODER-Verknüpfung zweier Funktionen
==	Gleichheit zweier Funktionen
≠	Ungleichheit zweier Funktionen

### 4. Rechte

Rechte: „read“ und „write“ sind disjunkt, „update“ ist die Vereinigung. „read“ ist Standard und wird genutzt, falls kein Recht spezifiziert wurde.