INTERBUS
Konformitätstest

PCP-Test

Inhalt

1 Einleitung

Dieser Teil der Spezifikation für die INTERBUS-Konformitätsprüfung beschreibt den allgemeinen Testrahmen für den Server-, Client- und Profil-Test. Im Vordergrund steht die Beschreibung des Testsystem zum Test von INTERBUS-Peripherals-Communication-Protocol-Geräten (PCP-Geräten). Es wird der Aufbau und die Generierung der PICS-Datei, die Bildung des Testergebnisses (Testurteil) zu einem Testcase-Ablauf und das Bezeichnungsschema der Testcases beschrieben. Die Testcases selber werden hier nicht erläutert. Abweichende Besonderheiten bzw. Ergänzungen, insbesondere beim Test von Geräten ohne PCP, sind in den speziellen Dokumentationen zu den einzelnen Testklassen vermerkt

2 Testziel

Ziel dieses Tests ist es, das Verhalten eines INTERBUS-Prüflings (INTERBUS-Geräts), den ein Hersteller zum Test einreicht, auf eine genau festgelegte Eigenschaft zu überprüfen. In diesem Fall ist es das Verhalten bezüglich seiner Funktion als INTERBUS-Teilnehmer (Server), als INTERBUS-Peripherals-Communication-Protocol-Teilnehmer (Server oder/und Client. Relevant ist dabei das Verhalten des Prüflings, das sich an der jeweiligen Schnittstelle (INTERBUS-Treiber bzw. PCP-ALI) zeigt.

Das Verhalten als PCP-Teilnehmer ist im INTERBUS-PCP-Referenzhandbuch verbindlich vorgeschrieben. Das Testsystem hat die Aufgabe durch Ausnutzen der PCP-Dienstprimitiven den Prüfling zu erforschen und seine Reaktion im Hinblick auf Konformität zum Referenzhandbuch bzw. zu der jeweiligen Profilspezifikation zu überprüfen. Beim Server-Test wird die auf dem Gerät laufende Anwendungssoftware genutzt. Für den Client-Test ist auf Geräteseite eine unterstützende Testsoftware ('Upper-Tester') notwendig. Für den PCP-Server-Test von PC-Slave-Anschaltungen steht eine Testapplikation zur Verfügung, die auf dem PC zu installieren ist.

3 Testumgebung

3.1 Prüf- und Messmittel

Die zur Durchführung der Konformitätsprüfung notwendigen Prüf- und Meßmittel sind im Anhang A des Teil 1 ”Allgemeines” aufgeführt. Für die Durchführung ist die Testsoftware für den PCP-Test unbedingt erforderlich.

3.2 Angaben des Herstellers

Der Hersteller des Prüflings muß zur Durchführung der Konformitätsprüfung eine Diskette im MSDOS-Format beilegen, auf der sich entweder die PICS-Datei, das Format wird nachstehend beschrieben, oder die drei Dateien kbl.dat, vfd.dat und ov.dat befinden, aus der die PICS-Datei erzeugt werden kann. Die drei *.DAT-Dateien werden bei der Implementierung der Protokollsoftware vom Hersteller benötigt.

Zur Generierung der PICS-Datei für Implementierungen mit PCP 2.0 steht ein Hilfswerkzeug zur Verfügung, das bereits beim Hersteller eingesetzt werden kann. Mit Hilfe des PICS-Generators lassen sich bereits Projektierungsfehler erkennen.

3.3 Testkonfiguration

Bilder 1 und 2 zeigen die Testkonfiguration aus Hardwaresicht und Bilder 3 bis 5 aus Softwaresicht. Beim Test des Querverkehrs sind zwei PCs für die Durchführung der Tests notwendig. Parallel zur Testsystemsoftware muß das Phoenix-Monitorprogramm PTED auf dem zweiten PC aktiv sein. Dazu muß nach Starten dieses Programms das PTED-Kommando 'w' zum automatischen Laden der KBL (Auto_Load_KBL) eingegeben werden. Als Interfacekarte im PC wird derzeit die INTERBUS PC AT-T Karte von Phoenix Contact eingesetzt.

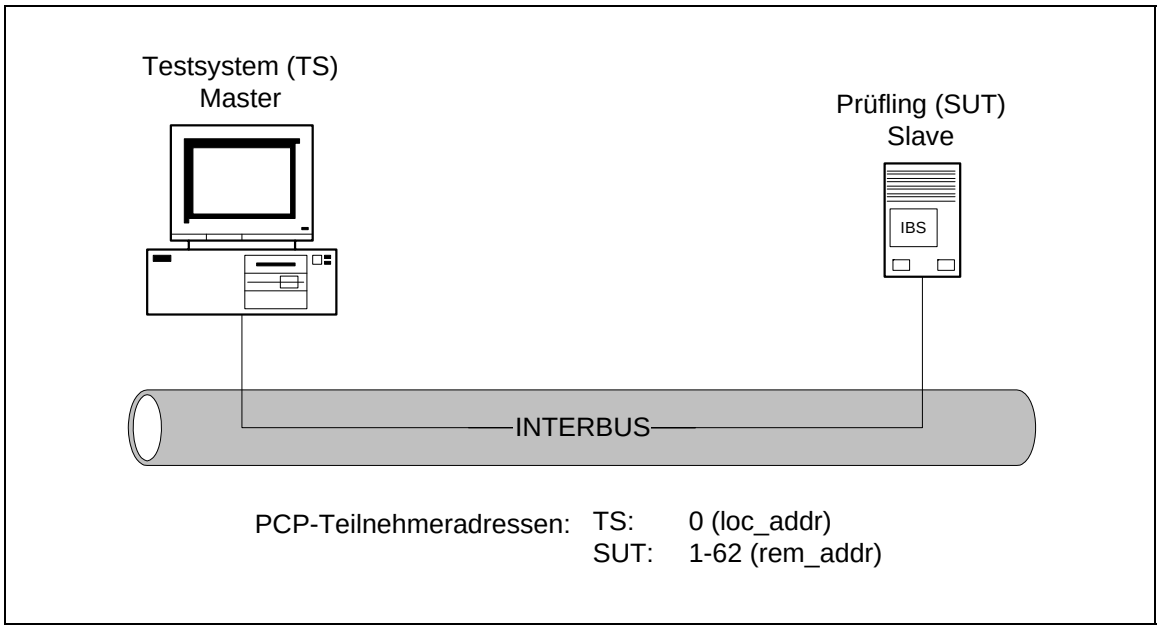

Bild 1: Testkonfiguration (Hardware) bei Test ohne Querverkehr

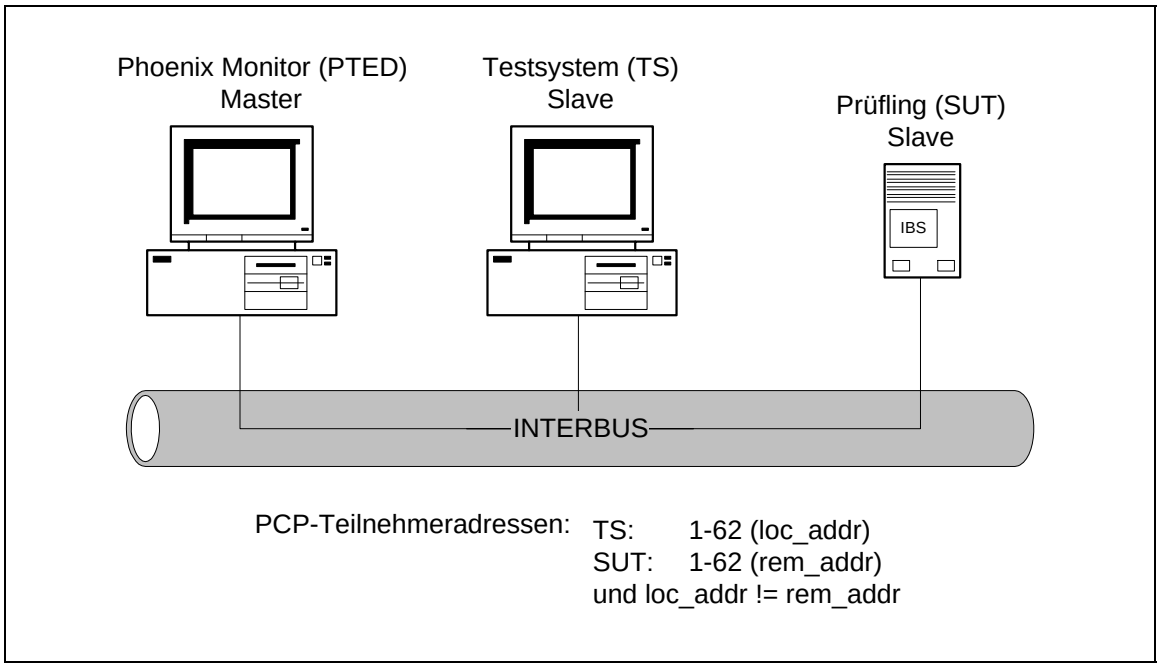

Bild 2: Testkonfiguration (Hardware) bei Test mit Querverkehr

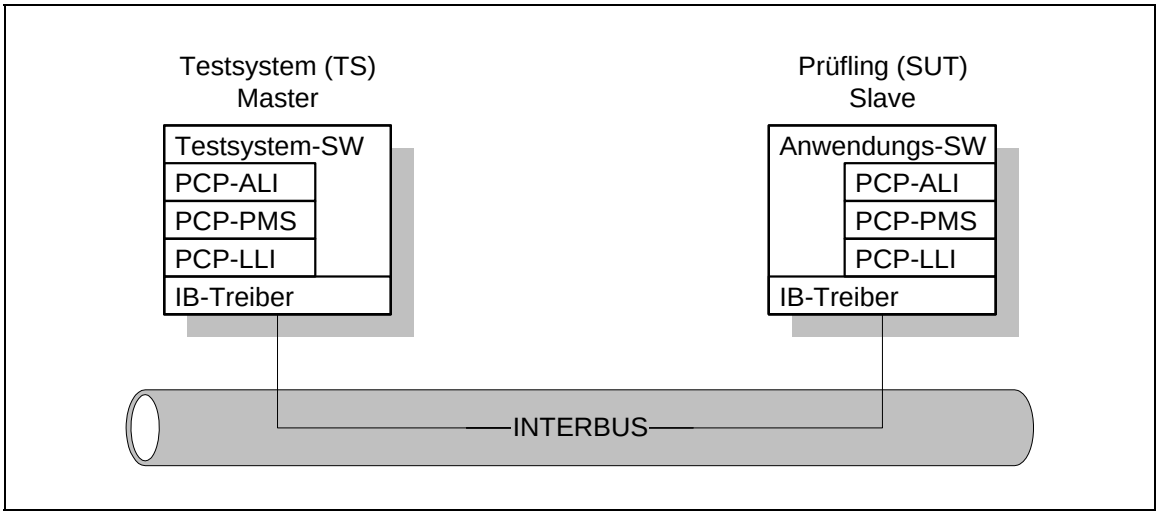

Bild 3: Server-Test (Geräte mit PCP(-Adresse))

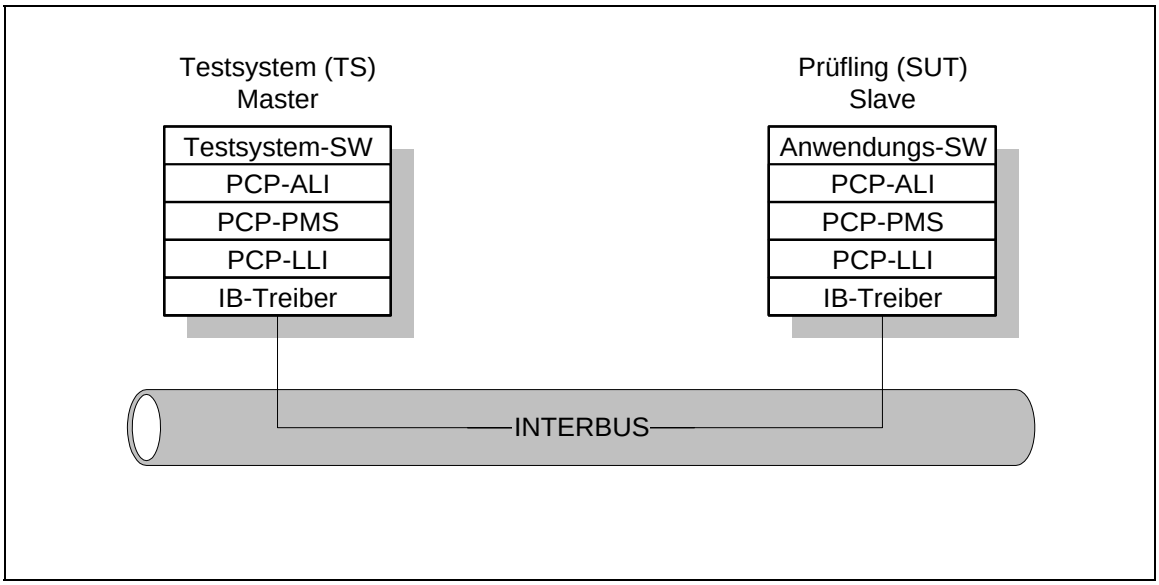

Bild 4: Client-Test (Geräte nur mit PCP(-Adresse))

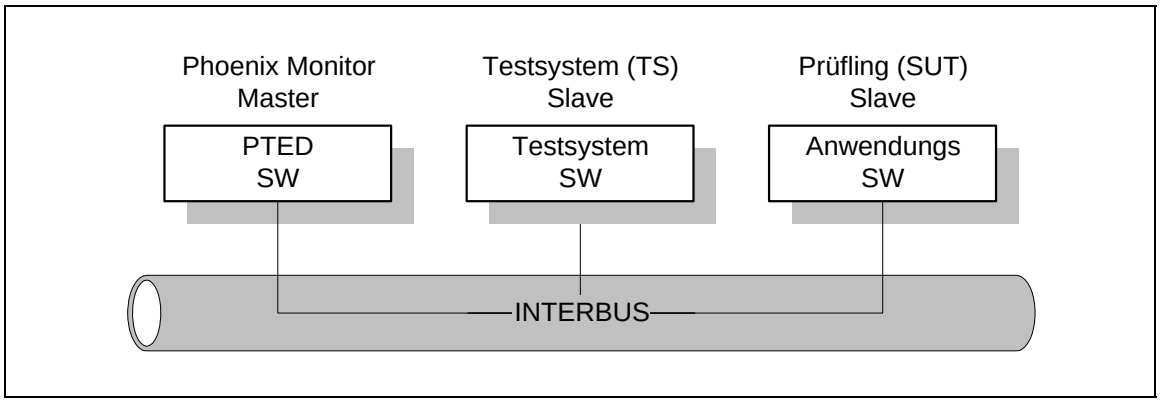

Bild 5: Test des Querverkehrs (Geräte nur mit PCP(-Adresse))

Es ist zu beachten, daß:

- beim PCP-Test die Testsystemsoftware direkt auf dem PCP-ALI (Application Layer Interface) aufsetzt, was den Programmieraufwand für das Testsystem erheblich vermindert. Damit ist das Verhalten des Prüflings nur an der ALI-Programmierschnittstelle für das Testsystem sichtbar. Der Testabdeckungsgrad für das "normale", fehlerfreie Verhalten wird dadurch nicht beeinflußt, wenn auf Testsystemseite eine fehlerfreie Referenzimplementierung zugrunde liegt. Die Erzeugung von Fehlerfällen zum Test der Fehlerdetektions- und Rekonfigurationsmechanismen auf den unteren Ebenen ist jedoch eingeschränkt.

- beim Server-Test die Anwendungssoftware als 'Upper-Tester' genutzt wird, d.h. auf die hier projektierten Objekte werden die im PCP Referenzhandbuch spezifizierten Dienste angewandt.

Vorteile durch den Test ohne 'Upper-Tester':

- Der Gerätehersteller erspart Aufwand an Zeit und Speicherplatz zur Installation bzw. Anpassung des 'Upper-Tester' und

- die Anwendung wird in den Test mit einbezogen.

Nachteile:

- Der Test ist nur vollständig in bezug auf die projektierten Dienste, Objekte, u.s.w. , und

- Probleme beim Zugriff auf die Objekte sind nicht ausgeschlossen, da dem Testsystem die Semantik der Objekte unbekannt ist (Abhilfe siehe Kapitel 4.1.1.3 Testeinschränkungen).

4 PICS/PIXIT-Datei

Die Spezifikationen gegen die getestet wird, lassen dem Gerätehersteller bei der Implementierung viele Freiheiten. So gibt es z.B.

- Pflichtdienste und optionale Dienste,
- Pflichtobjekte und optionale Objekte,
- Parameter, die in gewissen Grenzen frei einstellbar sind.

Das Testsystem muß prinzipiell so ausgelegt sein, daß es alle spezifizierten Eigenschaften prüfen kann. Beim Test eines einzelnen Geräts darf jedoch nur

- das Vorhandensein aller Pflichteigenschaften und
- das Einhalten der Spezifikation für alle Pflichteigenschaften und zusätzlich implementierten optionalen Eigenschaften überprüft werden.

Es ist daher notwendig, daß zu Testbeginn dem Testsystem eine genaue Beschreibung des Prüflings und damit der zu testenden Eigenschaften vorliegt. Dies geschieht über die PICS/PIXIT-Datei.

4.1 Syntax und Semantik

4.1.1 Geräte mit PCP

4.1.1.1 Server- und Client-Test bei Geräten mit PCP

Die PICS/PIXIT-Datei enthält beim Server- und Client-Test in zwei Teilen alle aus PCP-Sicht relevanten Angaben zum Prüfling.

Der erste Teil enthält Angaben zur Testkonfiguration und zum Testablauf:

- eine Angabe über die zu testende PCP-Version (es wird grundsätzlich gegen die aktuelle PCP-Version getestet),
- die PCP-Adresse unter der der Prüfling angesprochen werden kann
- und eine Überwachungszeit in Sekundenangabe, die die maximale Wartezeit des Testsystems auf Dienstausführung im Prüfling festlegt.

Der zweite Teil enthält die Beschreibung der vom Hersteller projektierten Eigenschaften des Prüflings, die zum Aufbau der folgenden PCP-Datenstruk-

turen und aller darin wieder enthaltenen bzw. darauf verwiesenen Datenstrukturen notwendig sind:

- T_KBL_HDR user_kbl_hdr (KBL-Header)
- T_OV_OBJ_DESCR_HDR ov_obj_descr_hdr (Objektverzeichnis)
- T_VFD_OBJECT vfd (VFD-Object)
- T_KBL_ENTRY_STAT user_kbl[] (Kommunikationsbeziehungen)

Die Beschreibungen werden in der PICS/PIXIT-Datei auf vier Hauptabschnitte abgebildet, die jeweils mit dem Schlüsselwort "PICS_PART:" eingeleitet werden. Es sind dies die Abschnitte

- OTHER,
- KBL,
- OV und
- VFD.

Diese Abschnitte können, mit Ausnahme von OTHER, in beliebiger Reihenfolge stehen. Sie selbst enthalten wieder Unterabschnitte, deren Reihenfolge jedoch festliegt. Die Unterabschnitte werden durch Schlüsselwörter eingeleitet und abgeschlossen. Alle vier Hauptabschnitte (OTHER, VFD, OV und KBL) müssen in der PICS/PIXIT-Datei enthalten sein und dürfen nicht mehrfach vorkommen.

Bei der Festlegung des Eingabeformats wurde darauf geachtet, daß die Namen für die Attribute, die Datentypen, die Aufzählungstypen und die Reihenfolge, in der die Werte für Attribute anzugeben sind, ihre Entsprechung im INTERBUS-PCP-Referenzhandbuch bzw. im INTERBUS-PCP-Programmierhandbuch finden. Es wird daher im folgenden weitestgehend darauf verzichtet, auf die Bedeutung der einzelnen Namen und deren Bezug zu PCP hinzuweisen. Zum Verständnis ist das Studium der PCP-Handbücher eine unverzichtbare Voraussetzung.

4.1.1.1.1 Formate für die PICS_PARTs und Beschreibung der Attribute

Es können beliebig viele Kommentarzeilen in der PICS/PIXIT-Datei enthalten sein. Kommentarzeilen werden durch das Zeichen * in der ersten Spalte eingeleitet.

Format der Beschreibung:

```
Attributname        Datentyp                Bedeutung
*
PICS_PART: OTHER                            * Schlüsselwort
*---------------------------
*
* Alle sonstigen Angaben
*
---other---                                 * Schlüsselwort
simulation:         BOOL                    * nur FALSE erlaubt
loc_addr:           USIGN8                  * PCP-Adresse des
                                            * Testsystems
rem_pms_version:    STRING                  * die zu testende PMS-Version
rem_add_qual:       T_ADDR                  * optional: siehe Testklasse
rem_addr:           USIGN8                  * PCP-Adresse des Prüflings
rem_time_out:       USIGN8                  * max. Wartezeit auf Antwort
                                            * in Sekunden
---end---                                   * Schlüsselwort
*
*
PICS_PART: VFD                              * Schlüsselwort
*----------------------
*
* VFD
*
---vfd---                                   * Schlüsselwort
vendor_name:        STRINGV                 * T_VFD_IDENT
model_name:         STRINGV
revision:           STRINGV
profile_name:       STRING8
logical_status:     T__LOGICAL_STATUS       * T_VFD_STATUS
physical_status:    T__PHYSICAL_STATUS
local_detail:       STRING8
---end---                                   * Schlüsselwort
*
*
PICS_PART: KBL                              * Schlüsselwort
*----------------------
*
* Kommunikationsbeziehungslisten-Header
```

```
*
---hdr_kbl---                                   * Schlüsselwort
com_ref:            T_COMREF
size:               USIGN8
poll_listen_ssap:   USIGN8
ass_abt_ci:         USIGN32
symbol_length:      USIGN8
vfd_ptr_supp:       BOOL
---end---                                       * Schlüsselwort
*
* Kommunikationsbeziehungsliste
*
---usr_kbl---                                   * Schlüsselwort
*
* Anzahl der folgenden Blöcke = size
*
com_ref:            T_COMREF
local_lsap:         USIGN8
rem_addr:           USIGN8
remote_lsap:        USIGN8
conn_type:          T__CONN_TYPE
lli_sap:            USIGN8
conn_attr:          T__CONN_ATTR
max_scc:            USIGN8
max_rcc:            USIGN8
max_sac:            USIGN8
max_rac:            USIGN8
aci:                USIGN32
multiplier:         USIGN8
req_len_h:          USIGN8
req_len_l:          USIGN8
ind_len_h:          USIGN8
ind_len_l:          USIGN8
serv_sup_client:    T__PMS_SERVICE bzw. T__PNM7_SERVICE
serv_sup_server:    T__PMS_SERVICE bzw. T__PNM7_SERVICE
symbol:             STRINGV
vfd_pointer:        USIGN32
*
---end---                                       * Schlüsselwort
*
*
* Bei Existenz einer Default Management Verbindung und
* Unterstützung des remote-seitigen Ladens der KBL:
*
---load_kbl_rem_params---
*
```

```
max_crl_size:         USIGN8
pms_serv_sup_client:  T__PMS_SERVICE
pms_serv_sup_server:  T__PMS_SERVICE
pnm7_serv_sup_client: T__PNM7_SERVICE
*
---end---
*
*
PICS_PART: OV                                   * Schlüsselwort
*---------------------
*
* Objektverzeichnis-Header
*
---hdr_ov---                                    * Schlüsselwort
index:                USIGN16
flag:                 BOOL
length:               USIGN8
protection:           BOOL
version:              USIGN16
len_st_ov:            USIGN16
first_index_s_ov:     USIGN16
len_s_ov:             USIGN16
first_index_dv_ov:    USIGN16
len_dv_ov:            USIGN16
first_index_p_ov:     USIGN16
len_p_ov:             USIGN16
---end---                                       * Schlüsselwort
*
*
* Objektverzeichnis
*
* Alle Nullobjekte können explizit und implizit projektiert
* werden. Für explizit projektierte Nullobjekte
* (obj_code=NULL_OBJECT) gilt:
*
index:                USIGN16
obj_code:             T__OBJ_CODE
*
*
* Standarddatentypen
*
---r_ov---                                      * Schlüsselwort
*
length:               USIGN16                   * Anzahl der folgenden Blöcke
*
* für projektierte Standarddatentypen (obj_code=DATA_TYPE_OBJECT)
```

```
*
index:              USIGN16
obj_code:           T__OBJ_CODE
meaning:            STRINGV
*
---end---                                   * Schlüsselwort
*
*
* Nicht standardisierte Datentypen und Datenstrukturtypen
*
---st_ov---                                 * Schlüsselwort
*
length:             USIGN16                 * Anzahl der folgenden Blöcke
*
* für Objekte mit obj_code = DATA_TYPE_OBJECT
*
index:              USIGN16
obj_code:           T__OBJ_CODE
meaning:            STRINGV
*
* für Objekte mit obj_code = TYPE_STRUCT_OBJECT
*
index:              USIGN16
obj_code:           T__OBJ_CODE
no_of_elements:     USIGN8
*
index_of_type:      USIGN16                 * no_of_elements Blöcke
length:             USIGN8
.
.
index_of_type:      USIGN16
length:             USIGN8
*
---end---                                   * Schlüsselwort
*
*
* Objekte des Statischen Objektverzeichnisses
*
* Objekte des Statischen Objektverzeichnisses können als Bereichs-
* Objektbeschreibung vorgegeben werden, wenn sie lückenlos hinter-
* einanderliegen und gleiche Eigenschaften besitzen. Dies gilt
* auch für Nullobjekte. Die Indexangabe wird dabei durch eine
* Bereichsangabe mit index_first und index_last ersetzt:
*
* index_first:        USIGN16
* index_last:         USIGN16
```

```
* obj_code:          T__OBJ_CODE
* .
* .
*
* Bei Vorgabe des Zugriffsrechts W_USER_DATA können Daten
* vorgegeben werden, die beim Schreibzugriff auf das Objekt im
* Test zu verwenden sind. Die Angabe erfolgt unter user_data
* am Ende der Objektbeschreibung:
*
* .
* .
* extension:         STRING8
* user_data:         STRING8
*
*
---s_ov---                                  * Schlüsselwort
*
length:             USIGN16                 * Anzahl der folgenden Blöcke
*
* für Objekte mit obj_code = SIMPLE_VAR_OBJECT
*
index:              USIGN16
obj_code:           T__OBJ_CODE
index_of_type:      USIGN16
length:             USIGN8
pass_word:          T_PASSWORD
acc_groups:         T__ACCESSGROUPS
acc_right:          T__ACCESSRIGHTS_V
symbol:             STRINGV
extension:          STRING8
*
* für Objekte mit obj_code = STRING_VAR_OBJECT
*
index:              USIGN16
obj_code:           T__OBJ_CODE
index_of_type:      USIGN16
length:             USIGN8
pass_word:          T_PASSWORD
acc_groups:         T__ACCESSGROUPS
acc_right:          T__ACCESSRIGHTS_V
symbol:             STRINGV
extension:          STRING8
*
* für Objekte mit obj_code = ARRAY_OBJECT
*
index:              USIGN16
```

```
obj_code:           T__OBJ_CODE
index_of_type:      USIGN16
length:             USIGN8
nof_elements:       USIGN8
pass_word:          T_PASSWORD
acc_groups:         T__ACCESSGROUPS
acc_right:          T__ACCESSRIGHTS_V
symbol:             STRINGV
extension:          STRING8
*
* für Objekte mit obj_code = RECORD_OBJECT
*
index:              USIGN16
obj_code:           T__OBJ_CODE
index_of_type:      USIGN16
pass_word:          T_PASSWORD
acc_groups:         T__ACCESSGROUPS
acc_right:          T__ACCESSRIGHTS_V
symbol:             STRINGV
extension:          STRING8
*
---end---                                   * Schlüsselwort
*
*
* Objekte des Dynamischen Variablenlistenverzeichnisses
*
---dv_ov---                                 * Schlüsselwort
*
length:             USIGN16                 * Anzahl der folgenden Blöcke
*
* für Objekte mit obj_code = VARIABLE_LIST_OBJECT
*
index:              USIGN16
obj_code:           T__OBJ_CODE
no_of_elements:     USIGN8
index:              USIGN16                 * no_of_elements index-Werte
.
.
index:              USIGN16
pass_word:          T_PASSWORD
acc_groups:         T__ACCESSGROUPS
acc_right:          T__ACCESSRIGHTS_V
deletable           BOOL
symbol:             STRINGV
extension:          STRING8
*
```

```
---end---                                   * Schlüsselwort
*
*
* Objekte des PI-Verzeichnisses
*
---p_ov---                                  * Schlüsselwort
*
length:          USIGN16                   * Anzahl der folgenden Blöcke
*
* für Objekte mit obj_code = INVOCATION_OBJECT
*
index:           USIGN16
obj_code:        T__OBJ_CODE
no_of_domains:   USIGN8
pass_word:       T_PASSWORD
acc_groups:      T__ACCESSGROUPS
acc_right:       T__ACCESSRIGHTS_PI
deletable:       BOOL
reusable:        BOOL
symbol:          STRINGV
extension:       STRING8
*
---end---                                   * Schlüsselwort
```

Entsprechend dem vorgesehenen Typ müssen die Werte zu den Attributen wie folgt beschrieben werden:

USIGN8, USIGN16, T_PASSWORD, T_COMREF:	Hexadezimale Zahl. Die Darstellung erfolgt ohne Voranstellung von 0x.
STRING:	Zeichenkette.
STRINGV:	Zeichenkettenlänge<Zeichenkette>. Nicht darstellbare Zeichen werden durch \NNN beschrieben, wobei NNN eine dreistellige Dezimalzahl im Bereich 000-127 ist. Das Zeichen \ wird durch \\ beschrieben.
STRING8:	Zeichenkettenlänge<Zeichenkette>. Die einzelnen Zeichen werden als zweistellige hexadezimale Zahl, getrennt durch Blank, dargestellt. Die Darstellung erfolgt ohne Voranstellung von 0x.

Bei STRINGV und STRING8 muß die angegebene Zeichenkettenlänge mit der Anzahl der Zeichen übereinstimmen, die durch die äußeren Zeichen < und > begrenzt sind. Die Zeichenkettenlänge ist als Dezimalzahl anzugeben.

BOOL,
T__OBJ_CODE,
T__ACCESSGROUPS,
T__ACCESSRIGHTS_V,
T__ACCESSRIGHTS_PI,
T__CONN_TYPE,
T__CONN_ATTR,
T__LOGICAL_STATUS,
T__PHYSICAL_STATUS,
T__SERVICE: Aufzählungstyp. Mehrere zulässige Alternativen werden durch Komma getrennt.

Als Wertebereich sind definiert:

USIGN8: 0x00 <= Wert <= 0xff.

USIGN16: 0x00 <= Wert <= 0xffff.

STRING: Zeichenkette bestehend aus druckbaren Zeichen (ohne Blank).

STRINGV: Zeichenkette bestehend aus Zeichen im Bereich 0-127.

STRING8: Zeichenkette bestehend aus hexadezimalen Zeichen im Bereich 0x00-0xff.

BOOL = {TRUE, FALSE}.

T__OBJ_CODE = {NULL_OBJECT, OV_OBJECT,
DATA_TYPE_OBJECT, TYPE_STRUCT_OBJECT,
SIMPLE_VAR_OBJECT, STRING_VAR_OBJECT,
ARRAY_OBJECT, RECORD_OBJECT,
VARIABLE_LIST_OBJECT,
INVOCATION_OBJECT}.

T__ACCESSGROUPS = {1, 2, 3, 4, 5, 6, 7, 8, -}:
1 = Zugriffsgruppe 1
2 = Zugriffsgruppe 2
3 = Zugriffsgruppe 3
4 = Zugriffsgruppe 4
5 = Zugriffsgruppe 5
6 = Zugriffsgruppe 6
7 = Zugriffsgruppe 7
8 = Zugriffsgruppe 8
- = keine Zugriffsgruppe

T__ACCESSRIGHTS_V = {R, W, Rg, Wg, Ra, Wa, NO_R, NO_W, NO_RW,
W_USER_DATA, -}:

R	= siehe PCP-Referenzhandbuch
W	= siehe PCP-Referenzhandbuch
Rg	= siehe PCP-Referenzhandbuch
Wg	= siehe PCP-Referenzhandbuch
Ra	= siehe PCP-Referenzhandbuch
Wa	= siehe PCP-Referenzhandbuch
NO_R	= kein erlaubter Lesezugriff auf Objekt im Test
NO_W	= kein erlaubter Schreibzugriff auf Objekt im Test
NO_RW	= kein erlaubter Zugriff auf Objekt im Test
W_USER_DATA	= Schreibzugriff mit herstellerspezifischen Daten
-	= es besteht kein Zugriffsrecht

T__ACCESSRIGHTS_PI = {S, H, Sg, Hg, Sa, Ha, NO_SH, H_USER, -}:

S	= siehe PCP-Referenzhandbuch
H	= siehe PCP-Referenzhandbuch
Sg	= siehe PCP-Referenzhandbuch
Hg	= siehe PCP-Referenzhandbuch
Sa	= siehe PCP-Referenzhandbuch
Ha	= siehe PCP-Referenzhandbuch
NO_SH	= kein erlaubtes Starten/Stoppen einer PI im Test
H_USER	= Applikation beeinflußt PI-Zustandsmaschine im Zustand PI_RUNNING (Übergang nach PI_IDLE oder PI_STOPPED)
-	= es besteht kein Zugriffsrecht

T__CONN_TYPE = {MMAZ}.

T__CONN_ATTR = {CONN_ATTR_D,CONN_ATTR_O}.

T__LOGICAL_STATUS = {
L_STAT_KOMMUN_BEREIT, L_STAT_BEGR_ANZ_SVC,
L_STAT_READY_FOR_COMM, L_STAT_LIMITED_NO_SVC}.

T__PHSICAL_STATUS = {
P_STAT_BETRIEBS_BEREIT, P_STAT_TEILW_BEREIT,
P_STAT_NICHT_BEREIT, P_STAT_WARTUNG_ERFORDERL,
P_STAT_READY_FOR_OP, P_STAT_PARTIALLY_READY,
P_STAT_NOT_READY_FOR_OP, P_STAT_SERVICE_REQUIRED}.

T__PMS_SERVICE = {MANDATORY, INFO_REP, GET_OV_LONG, PI,
READ, WRITE}.

MANDATORY	= alle Mandatory-Services werden unterstützt
INFO_REP	= Information-Report-Service wird unterstützt

```
  GET_OV_LONG    = Get-OV-Service wird in Langversion unterstützt
  PI             = Start-/Stop-Service wird unterstützt
  READ           = Read-Service wird unterstützt
  WRITE          = Write-Service wird unterstützt

T__PNM7_SERVICE = {MANDATORY, LOAD_KBL_REM,
                   READ_KBL_REM, -}
  MANDATORY      = alle Mandatory-Services werden unterstützt
  LOAD_KBL_REM   = Load-KBL-Rem-Service wird unterstützt
  READ_KBL_REM   = Read-KBL-Rem-Service wird unterstützt
```

4.1.1.1.2 Formate für die PICS_PARTs und Beschreibung der Attribute

Es können beliebig viele Kommentarzeilen in der PICS/PIXIT-Datei enthalten sein. Kommentarzeilen werden durch das Zeichen * in der ersten Spalte eingeleitet.

```
Format der Beschreibung:
Attributname        Datentyp                Bedeutung
*
PICS_PART: OTHER_PROFILE                    * Schlüsselwort
*-----------------------------------------
*
* Identifikationskopf für Profil
*
---other---                                 * Schlüsselwort
prof_name:          STRING                  * Profilname
prof_version:       STRING                  * Profilversion
prof_number:        STRING8                 * Profilnummer
---end---                                   * Schlüsselwort
*
*
PICS_PART: OV_OBJ_PROFILE                   * Schlüsselwort
*-----------------------------------------
*
* Die Objektbeschreibung ist identisch zu der Beschreibung unter
* PICS_PART OV in 4.1.1.1.1
*
*
PICS_PART: OV_PAR_PROFILE                   * Schlüsselwort
*-----------------------------------------
*
* Wertebeschreibung und Objektfunktion
```

```
*
* Standarddatentypen
*
---r_ov---                                  * Schlüsselwort
*
length:             USIGN16                 * Anzahl der folgenden Blöcke
*
index:              USIGN16                 * Objektindex
class:              T__CLASS                * Objektklasse
*
---end---
*
*
* Datenstrukturtypen
*
---st_ov---                                 * Schlüsselwort
*
length:                                     * Anzahl der folgenden Blöcke
*
index:              USIGN16                 * Objektindex
class:              T__CLASS                * Objektklasse
no_of_elem_to_supp: USIGN8                  * Anzahl der Elemente, die
                                            * unterstützt werden müssen
*
---end---                                   * Schlüsselwort
*
*
* Objekte des Statischen Objektverzeichnisses
*
---s_ov---                                  * Schlüsselwort
*
length:             USIGN16                 * Anzahl der folgenden Blöcke
*
* für Objekte mit obj_code = NULL_OBJECT
*
index:              USIGN16                 * Objektindex
class:              T__CLASS                * Objektklasse
*
* für Objekte mit obj_code = SIMPLE_VAR_OBJECT
*
index:              USIGN16                 * Objektindex
class:              T__CLASS                * Objektklasse
process:            T__PROCESS              * Prozeßdatenabbildung
error_codes:        T__ERROR_CODE           * Fehlercodes
range:              T__RANGE                * Profilspez. Wertebereich
range_to_supp:      T__RANGE                * Profilspez. Pflichtbereich
```

```
default:                T__VAL                  * Ersatzwert
*
* für Objekte mit obj_code = ARRAY_OBJECT bzw. RECORD_OBJECT
*
index:                  USIGN16                 * Objektindex
class:                  T__CLASS                * Objektklasse
process:                T__PROCESS              * Prozeßdatenabbildung
error_codes:            T__ERROR_CODE           * Fehlercodes
no_of_elem_to_supp: USIGN8                      * Anzahl der Elemente, die
                                                * unterstützt werden müssen
*
* Beschreibung der Elemente
*
subindex:               USIGN8                  * Subindex
range:                  T__RANGE                * Profilspez. Wertebereich
range_to_supp:          T__RANGE                * Profilspez. Pflichtbereich
default:                T__VAL                  * Ersatzwert
*
*
---end---                                       * Schlüsselwort
*
*
* Objekte des Dynamischen Variablenlistenverzeichnisses
*
---dv_ov---                                     * Schlüsselwort
*
length: 0
*
---end---                                       * Schlüsselwort
*
*
* Objekte des PI-Verzeichnisses
*
---p_ov---                                      * Schlüsselwort
*
length: 0
*
---end---                                       * Schlüsselwort
*
*
*
PICS_PART: OV_PAR_DICS                          * Schlüsselwort
*--------------------------------------
*
* profilunabhängige profilspezifische Eigenschaften des Prüflings
*
```

```
* Objekte des Statischen Objektverzeichnisses
*
---s_ov---                                  * Schlüsselwort
*
length:             USIGN16                 * Anzahl der folgenden Blöcke
*
* für Objekte mit obj_code = SIMPLE_VAR_OBJECT
*
index:              USIGN16                 * Objektindex
range_supp:         T__RANGE                * Herstellerspez. Wertebereich
*
*
* für Objekte mit obj_code = ARRAY_OBJECT bzw. RECORD_OBJECT
*
index:              USIGN16                 * Objektindex
no_of_elem_supp:    USIGN8                  * Anzahl der Elemente,
                                            * die unterstützt werden.
*
* Beschreibung der Elemente
*
subindex:           USIGN8                  * Subindex
range_supp:         T__RANGE                * Herstellerspez. Wertebereich
*
*
---end---                                   * Schlüsselwort
*
*
* Objekte des Dynamischen Variablenlistenverzeichnisses
*
---dv_ov---                                 * Schlüsselwort
*
length: 0
*
---end---                                   * Schlüsselwort
*
*
* Objekte des PI-Verzeichnisses
*
---p_ov---                                  * Schlüsselwort
*
length: 0
*
---end---                                   * Schlüsselwort
*
*
*
```

Entsprechend dem vorgesehenen Typ müssen die Werte zu den Attributen wie folgt beschrieben werden:

USIGN8, USIGN16, USIGN32, INT16:	Hexadezimale Zahl. Die Darstellung erfolgt ohne Voranstellung von 0x.
STRING, STRING8:	Siehe 4.1.1.1.1
T__CLASS, T__PROCESS:	Aufzählungstyp.
T__ERROR_CODE:	Anzahl_der_Error-Codes{Error-Codes}.
T__RANGE:	Bereichstyp Anzahl_der_Werte{Werte}.
T__VAL:	Default-Wert.

Als Wertebereiche sind definiert:
USIGN8, USIGN16, STRING, STRING8: Siehe 4.1.1.1.1

USIGN32:	0x00000000 <= Wert <= 0xffffffff.
INT16:	0x8000 <= Wert <= 0x7fff.

T__CLASS = {OPTIONAL_OBJECT, MANDATORY_OBJECT}:

OPTIONAL_OBJECT	= optionales Objekt (kann projektiert sein),
MANDATORY_OBJECT	= mandatory Objekt (muß projektiert sein).

T_PROCESS = {PE, PA, PEA, -}:

PE	= Prozeßeingangsdatenabbildung möglich,
PA	= Prozeßausgangsdatenabbildung möglich,
PEA	= Prozeßdatenabbildung für Ein- und Ausgangsdaten möglich,
-	= keine Prozeßdatenabbildung möglich.

T_RANGE:

Bereichstypen, denen keine Werte zugeordnet sind:

EQ_DEF	= Bereich entspricht dem Definitions-/Wertebereich,
NO_DEF	= Es erfolgt keine weitere Einschränkung.

Bereichstypen, denen ein Wert zugeordnet ist:

EQ_VAL 1{Wert1}	= Als Wert wird nur Wert1 unterstützt,
NE_VAL 1{Wert1}	= Alle Werte außer Wert1 werden unterstützt,
IN_MASK 1{Wert1}	= Alle Bits innerhalb der mit Wert1 vorgegebenen Maske werden unterstützt.

Bereichstypen, denen zwei Werte zugeordnet sind:

IN_RANGE 2{Wert1, Wert2} = Werte mit Wert1<=Wert<=Wert2

werden unterstützt.

Bei der Spezifikation der Wertebereiche werden folgende Bereichstypen verwendet:

Spezifikation des profilspezifischen Wertebereichs (range):

EQ_DEF:	Wertebereich = Definitionsbereichs des projektierten Datentyps,
NE_VAL:	Wertebereich wie bei EQ_DEF, nur ohne den angegebenen Wert.

Spezifikation des profilspez. Pflichtbereichs (range_to_supp):

NO_DEF:	Es gibt keinen Pflichtbereich,
EQ_VAL:	Wert muß unterstützt werden,
IN_RANGE:	Der angegebene Bereich muß unterstützt werden,
IN_MASK:	Die Bits innerhalb der Maske müssen unterstützt werden (nur bei Datentyp OCTET_STRING).

Spezifikation des herstellerspez. Wertebereichs (range_supp):

EQ_DEF:	Herstellerspezifischer Wertebereich = profilspezifischer Wertebereich,
EQ_VAL:	Nur dieser Wert und ein mit EQ_VAL spezifizierter Pflichtwert wird unterstützt,
IN_RANGE:	Der angegebene Bereich und ein mit EQ_VAL spezifizierter Pflichtwert wird unterstützt,
IN_MASK:	Die Bits innerhalb der Maske werden unterstützt (nur bei Datentyp OCTET_STRING).

4.1.1.2 Testeinschränkungen

Bei bestimmten Anwendungen kann es notwendig sein, daß der Zugriff auf einzelne projektierte Objekte während des Test nur eingeschränkt möglich oder im Extremfall verboten ist. Dem Ersteller der PICS/PIXIT-Datei wird deshalb die Möglichkeit gegeben, vor Übernahme der PICS/PIXIT-Datei in das Testsystem, die Zugriffsrechte ('acc_right') für den Test (Lesen/Schreiben eines Objekts des Statischen Objektverzeichnisses, Starten/Stoppen einer PI (Program-Invocation)) einzuschränken, ohne die in PMS definierten Rechte zu beeinflussen. Im Klartext: Laut PMS-Zugriffsrecht ist z.B. das Starten einer PI erlaubt, die Einschränkung des Erstellers der PICS/PIXIT-Datei sorgt jedoch dafür, daß dieses Recht während des Tests nicht ausgeübt wird. Die Einschränkungen gelten

jedoch nicht für Tests bei denen ein fehlerhafter Zugriff abgewiesen werden muß!

Neben den in PMS definierten Zugriffsrechten (R, W, Rg, Wg, Ra, Wa, S, H, Sg, Hg, Sa, Ha) sind als einschränkende Zusätze erlaubt:

Bei Zugriffen auf Objekte des Statischen Objektverzeichnisses:

NO_W:	Kein erlaubter Schreibzugriff
NO_R:	Kein erlaubter Schreibzugriff Lesezugriff
NO_RW:	Kein erlaubter Schreibzugriff Schreib-/Lesezugriff
W_USER_DATA:	Erlaubte Schreibzugriffe nur unter Verwendung der herstellerspezifischen Daten

Bei Zugriffen auf Objekte des PI-Verzeichnisses:

NO_SH:	Kein erlaubtes Starten/Stoppen einer PI
H_USER:	Applikation beeinflußt PI-Zustandsmaschine im Zustand PI_RUNNING (Übergang nach PI_IDLE oder PI_STOPPED)

Mit den Testeinschränkungen NO_xx sollte sparsam umgegangen werden, da deren Verwendung zum Testurteil "inconclusive", also "nicht aussagefähiger Test", führen kann.

Bei der Ausführung des Write-Service hat der Ersteller der PICS-Datei die Möglichkeit für Objekte des Statischen Objektverzeichnisses die Daten für den Schreibpuffer festzulegen. Diese Daten werden dann auch bei der Ausführung des Write-Service auf diese Objekte über Variablenlisten genutzt.

Für das Statische Objektverzeichnis ist die Vorgabe von Bereichs-Objektbeschreibungen möglich. Diese können genutzt werden wenn lückenlos hintereinanderliegende Objekte gleiche Eigenschaften (außer den Attributen index und symbol) aufweisen. Im Test, Ausnahme Get-OV-Test über Startindex, wird dann nur das erste Objekt verwendet.

4.2 Generierung

Bild 6 zeigt den Generierungsweg für PICS/PIXIT-Dateien zum Test von Geräten mit PCP. PICS/PIXIT-Dateien zum Test von Geräten ohne PCP sind vom Gerätehersteller mit Hilfe eines Editors zu erstellen. Musterdateien erleichtern die Erstellung.

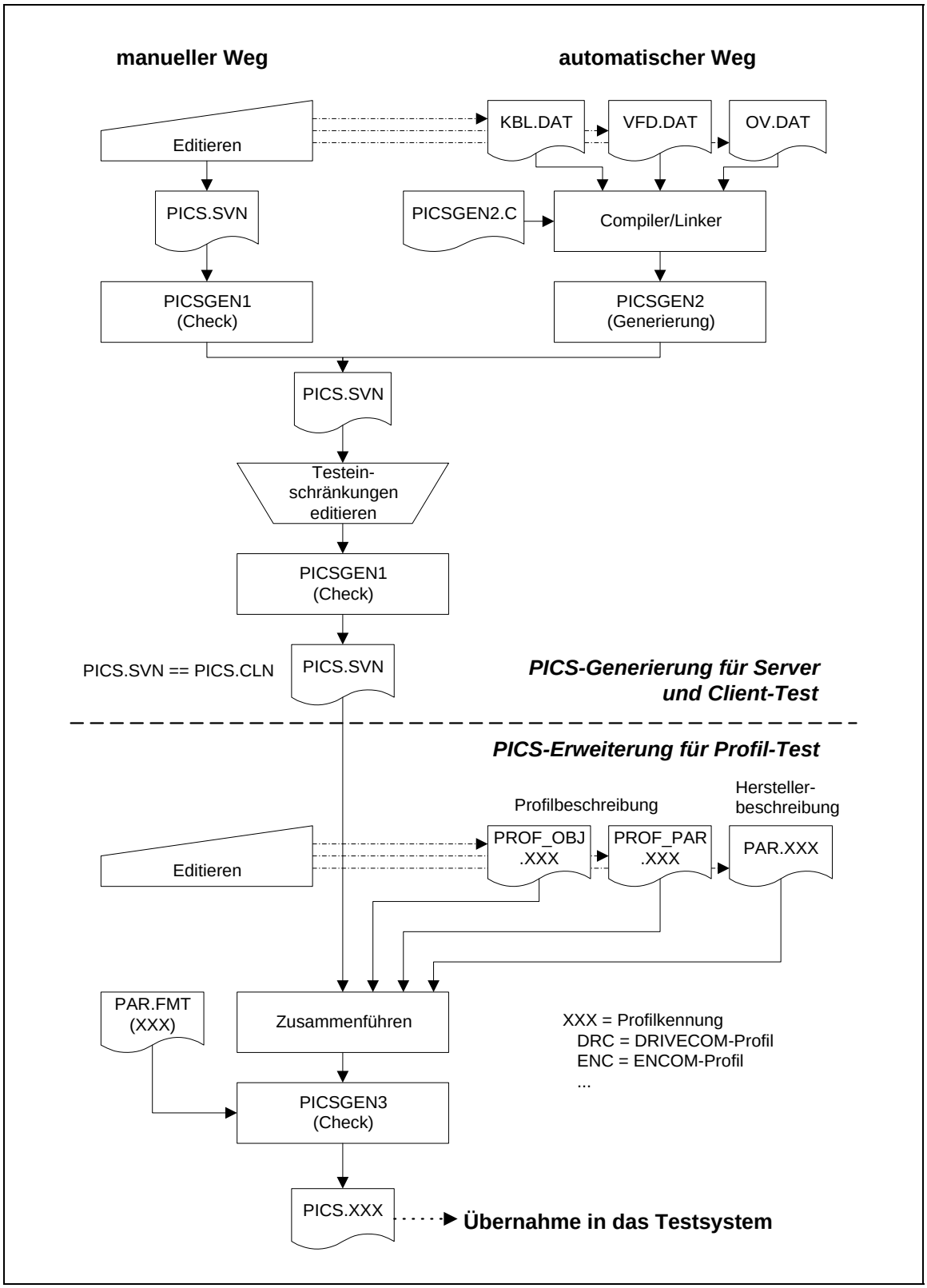

Bild 6: Erstellungsweg für die PICS/PIXIT-Datei

4.2.1 Generierung beim Server- und Client-Test

Die PICS/PIXIT-Dateien 'pics.svn' und 'pics.cln' sind vom Aufbau her identisch. Es gibt zwei Erstellungswege:

- Die manuelle Erstellung über einen Editor und

- die automatische Generierung über die c-Quellfiles des Prüflings 'kbl.dat', 'vfd.dat' und 'ov.dat', sofern diese entsprechend den Vorgaben aus dem PCP-Programmierhandbuch erstellt wurden. Wird in den '.dat'-Files auf Variable verwiesen, die in diesen Files nicht definiert sind, so müssen diese Files zuvor entsprechend aufbereitet werden.

Bei der manuellen Erstellung wird die Datei 'pics.svn' mit Hilfe eines Editors, z.B. Norton Editor, im geforderten PICS/PIXIT-Format erstellt. Die so erstellte Datei kann dann mit Hilfe des Programms 'picsgen1' auf Vollständigkeit, Einhaltung der Syntaxregeln, Konsistenz und PCP-Konformität überprüft werden.

Bei der automatischen Erstellung werden die Dateien 'kbl.dat', 'vfd.dat' und 'ov.dat', die der Hersteller in der Regel schon erstellt hat, als include-Files für den c-Quellfile 'picsgen2.c' verwendet und daraus das Programm 'picsgen2' erzeugt. Die Ausführung dieses Programms vollzieht dann die gewünschte Umsetzung in das PICS/PIXIT-Format und erzeugt als Ergebnis die Datei 'pics.svn'.

Die Datei 'pics.svn' kann für den Server-Test zur Testeinschränkung nach den Regeln aus Abschnitt 4.1.1.3 weiterbearbeitet werden.

Für die Angaben unter PICS_PART OTHER werden Default-Vorgaben eingetragen, die der Ersteller der PICS/PIXIT-Datei überprüfen und eventuell korrigieren muß. Defaulteinstellungen:

```
simulation:          FALSE          * nur FALSE erlaubt
loc_addr:            0
rem_pms_version:     2.0            * aktuelle PCP-Version
rem_addr_qual:       PMS_ADDR
rem_addr:            1              * PCP-Adresse des Prüflings
rem_time_out:        1              * max. Wartezeit auf Antwort
                                    * in Sekunden
```

Das Schlüsselwort 'rem_addr_qual' dient zur weiteren Festlegung des Zugriffsverfahrens. Beim Client-Test wird hier z.B. die Adresse des 'Upper-Tester-Agent'-Objekts vermerkt. Das Schlüsselwort kann fehlen wenn keine weitere Qualifizierung für den Zugriff notwendig ist. In diesem Fall ist die unter 'rem_addr' angegebene Adresse automatisch die PCP-Adresse. Bei Nutzung

findet sich eine Beschreibung in der Dokumentation zu der jeweiligen Testklasse.

4.2.2 Generierungsaufruf

Für die automatische Generierung von PICS/PIXIT-Dateien wird der Aufruf 'picsgen' mit folgender Syntax verwendet:

picsgen <pics_path> <class>

<pics_path>:	Pfad zu dem Directory mit den PICS/PIXIT-Dateien
<class>:	Testklasse der die PICS/PIXIT-Dateien angehören

Beispiel: picsgen a:\pics drc (Generierung für DRIVECOM-Profil)

Die Generierung findet abgestuft statt:

- Existiert die Datei 'pics.drc' unter dem angegebenen Directory, so wird nur 'picsgen3' ausgeführt,

- existiert die Datei 'pics.drc' nicht und existiert die Datei 'pics.svn', so wird die Datei 'pics.drc' aus den Dateien 'pics.svn', 'prof_obj.drc', 'prof_par.drc' und 'par.drc' erzeugt,

- existiert keine der Dateien 'pics.svn' und 'pics.drc', so wird 'picsgen2' ausgeführt und zuerst die Datei 'pics.svn' aus den Dateien 'kbl.dat', 'vfd.dat' und 'ov.dat' gebildet und danach die Datei 'pics.drc' erzeugt.

4.3 Validierung

4.3.1 Validierung beim Server- und Client-Test

Die PICS/PIXIT-Datei 'pics.svn' wird beim Einlesen überprüft auf

- Vollständigkeit und
- syntaktische Korrektheit (siehe 4.1),
- Konsistenz
- und Versionskonformität (hier zur PCP-Version V2.0).

4.3.1.1 Konsistenz beim Server- und Client-Test

Während des Einlesevorgangs der PICS/PIXIT-Datei finden folgende Überprüfungen zu den Unterabschnitten statt:

kbl_hdr:

com_ref	=	0,	
size	<	128,	
symbol_length	<	SYMBOL_LEN (12).	

usr_kbl:

com_ref:	muß bei Vorhandensein einer Default-Management-Verbindung mit 1 ansonsten mit 2 beginnen und dann in lückenloser Reihenfolge fortgesetzt sein,
rem_addr:	jede Adresse darf innerhalb der PMS- bzw. PNM7-KBL nur einmal vorhanden sein.

hdr_ov:

index	=	0,	
length	<	SYMBOL_LEN,	
len_st_ov	>=	0xe,	
first_index_s_ov	>	len_st_ov,	**
first_index_dv_ov	>=	first_index_s_ov + len_s_ov,	**
first_index_p_ov	>=	first_index_dv_ov + len_dv_ov,	**
first_index_p_ov + len_p_ov - 1 <= 0xffff.			**

**: Die Tests werden nur durchgeführt, wenn die Objektbereiche vorhanden sind (Länge != 0).

ov:

index und index_of_type:	die Werte dazu müssen in ihren jeweiligen Bereichen liegen,
index:	muß eine aufsteigende Aufeinanderfolge besitzen,
index_of_type:	es darf keine Referenzen auf Nullobjekt (obj_code = NULL_OBJECT oder nicht explizit projektiertes Nullobjekt) geben,
length:	die Längenangaben zu den realen Objekten basierend auf PMS-Standarddatentypen müssen den Längenangaben aus dem PMS-Referenzhandbuch entsprechen. Bei Objekten vom Typ VISIBLE_STRING, OCTET_STRING und BIT_STRING muß die Länge > Null sein.

r_ov:

obj_code:	darf nur die Werte DATA_TYPE_OBJECT oder NULL_OBJECT annehmen.

st_ov:

obj_code:	darf nur die Werte DATA_TYPE_OBJECT, TYPE_STRUCT_OBJECT oder NULL_OBJECT annehmen,
index_of_type:	muß Datentyp referenzieren.

s_ov:

obj_code:	darf nur die Werte SIMPLE_VAR_OBJECT, STRING_VAR_OBJECT, ARRAY_OBJECT, RECORD_OBJECT oder NULL_OBJECT annehmen,
index_of_type:	muß bei Objekten mit obj_code = RECORD_OBJECT Datenstrukturtyp, ansonsten Datentyp referenzieren.

dv_ov:

obj_code:	darf nur die Werte VARIABLE_LIST_OBJECT oder NULL_OBJECT annehmen,
index:	es dürfen nur SIMPLE_VAR_OBJECT, ARRAY_OBJECT oder RECORD_OBJECT-Objekte referenziert werden.

p_ov:

obj_code:	darf nur die Werte INVOCATION_OBJECT oder NULL_OBJECT annehmen.

s_ov/p_ov:

Innerhalb der einzelnen Objektverzeichnisbereiche darf, sofern symbolische Adressierung möglich ist, jeder Name nur je einmal benutzt sein. Namen mit der Länge Null werden nicht in den Vergleich einbezogen.

Ist das Zugriffsrecht Rg, Wg, Sg oder Hg gesetzt, so muß mindestens eine Zugriffsgruppe angegeben sein. Bei Vergabe der Zugriffsrechte R, W, S und H muß das Paßwort ungleich Null sein.

Für Programm-Invocations gilt, daß bei eingeschaltetem Zugriffsrechtetest mindestens eines der Zugriffsrechte S, Sg oder Sa gesetzt sein muß.

4.3.1.2 Versionskonformität beim Server- und Client-Test

Folgende Attribute werden auf Konformität zur PCP-Version V2.0 geprüft:

hdr_kbl:

poll_listen_ssap	=	NIL,
ass_abt_ci	<=	65535

symbol_length <= 0xb,
vfd_ptr_supp = FALSE.

usr_kbl:
Allgemein:
local_lsap = NIL,
remote_lsap = NIL,
conn_type = MMAZ,
aci = 0 oder
aci = 0xffffffff,
multiplier = NIL,
req_len_h = 0,
ind_len_h = 0,
vfd_pointer = 0.

Für Default-Management-Verbindung (com_ref = 1):
lli_sap = 1,
conn_attr = CONN_ATTR_O
max_scc = 0,
max_rcc = 1,
max rac = 0,
max_sac = 0,
53 >= req_len_l <= 246,
53 >= ind_len_l <= 246,
serv_sup[0] = 0.

Für PMS-Verbindung:
lli_sap = 0,
conn_attr = CONN_ATTR_D
max_scc = 1,
max_rcc = 1,
max rac = 1,
max_sac = 1,
51 >= req_len_l <= 246,
51 >= ind_len_l <= 246.

Für PNM7-Verbindung (com_ref != 1):
lli_sap = 1,
conn_attr = CONN_ATTR_D
max_scc = 1,
max_rcc = 0,
max rac = 0,
max_sac = 0,
53 >= req_len_l <= 246,
53 >= ind_len_l <= 246,
serv_sup[1] = 0.

vfd:
profile_name[0] = 2.

hdr_ov:
length <= 0xb.

ov:
dv-Objekte:
deletable = FALSE

p-Objekte:
no_of_domains = 0,
deletable = FALSE,
reusable = TRUE.

5 Testcases

Aufgabe des Testsystems ist es, das Verhalten eines INTERBUS-Prüflings (Geräts) auf eine genau festgelegte Eigenschaft zu überprüfen, seine Funktion als INTERBUS-Teilnehmer (Server oder/und Client) mit und ohne PCP bzw. sein Verhalten als Angehöriger einer bestimmten Anwendungsklasse (Profil). Dazu werden in einer Testspezifikation die einzelnen zu überprüfenden Testfälle festgelegt. In den Testfällen ist festgelegt wie sich ein Prüfling unter gegebenen Randbedingungen zu verhalten hat.

5.1 Begriffe

Die Testfälle werden auf sogenannte Testcases in Form von Testprogrammen abgebildet mit denen dann die einzelnen Eigenschaften des Prüflings getestet werden. Testcases, die in ihrer Gesamtheit dazu dienen eine übergeordnete Eigenschaft zu prüfen, werden zu Testklassen zusammengefaßt. Beispiele für implementierte Testklassen sind:

- Testklasse SVN für den Test der PCP-Server-Funktionalität,
- Testklasse CLN für den Test der PCP-Client-Funktionalität und
- Testklasse DRC für den Test der Funktionalität entsprechend der DRIVECOM-Profilspezifikation.
- Testklasse ENC für den Test der Funktionalität entsprechend der ENCOM-Profilspezifikation.

Innerhalb einer Testklasse können die Testcases in Test-Schedules zusammengefaßt werden. Bei Ausführung eines Test-Schedules werden die Testcases dann in der Reihenfolge ihrer Einkettung ausgeführt. In den Testklassen SVN und CLN sind Testcases, die gleiche Dienste abtesten, in Test-Schedules zusammengefaßt. Der Test-Schedule 'all' beinhaltet in der Regel alle Testcases einer Testklasse, deren fehlerfreie Ausführung zur Erlangung des Zertifikats für diese Testklasse notwendig ist.

5.2 Klassifizierungsschema

Für die Namengebung zu den Testcases wurde die Klassifizierung nach Bild 7 und nachfolgenden Erläuterungen verwendet:

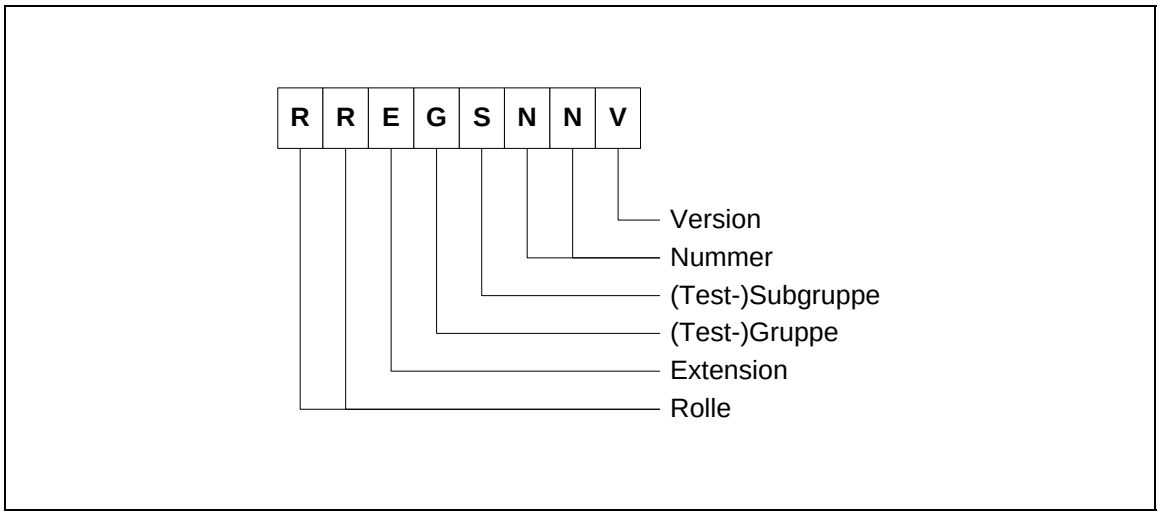

Bild 7: Namengebung für die Testcases

Erläuterung zu Bild 7:

Für den Server- und den Client-Test gilt:

Rolle:	SV	=	Test der PCP-Server-Funktionalität eines Prüflings
	CL	=	Test der PCP-Client-Funktionalität eines Prüflings
Extension:	E	=	erweiterter Test, d.h. Test über Schicht 2 (zur Zeit nicht realisiert)
	N	=	normaler Test, d.h. Test über PCP/ALI-Anbindung

Für den Profil-Test gilt:

Rolle:	DRC	=	DRIVECOM-Profil-Test
	ENC	=	ENCOM-Profil-Test
	...		

Für den Profil-Test wird eine eventuelle Detaillierung in den jeweiligen Profilpapieren beschrieben, für den Server- und den Client-Test gilt die nachfolgende Detaillierung.

CHECK-Test-Cases

Gruppe:

0	=	Projektierung

PMS Subgruppe:

0	=	Objektverzeichnis
1	=	PMS-Services-Objekte
2	=	PMS-Services-Objekte-Pufferlängen
3	=	Objekte-Objektattribute

PNM7 Subgruppe:

6	=	Kommunikationsbeziehungsliste (KBL)
7	=	PNM7-Services-KBL

PMS-Test-Cases

Gruppe:

- 1 = Context-Management

  Subgruppe:

  - 0 = Initiate-Service (verträglicher PMS-/LLI-Context)
  - 1 = Initiate-Service (unverträglicher ")
  - 2 = Abort-Service
  - 3 = Reject-Service
  - 4 = Zustandsmaschine
- 2 = VFD-Support
  - 0 = Identify-Service
  - 1 = Status-Service
- 3 = OV-Management
  - 0 = Get-OV-Service (mandatory)
  - 1 = Get-OV-Service long (optional)
- 4 = Program-Invocation-Management
  - 0 = Start-Service
  - 1 = Stop-Service
  - 2 = Resume-Service
  - 3 = Reset-Service
  - 4 = Zustandsmaschine
  - 5 = Zustandsmaschine nach power-on
- 5 = Variable-Access
  - 0 = Read-Service (Statisches OV)
  - 1 = Write-Service (Statisches OV
  - 2 = Read-Service (Dyn. Variablenlisten OV)
  - 3 = Write-Service (Dyn. Variablenlisten OV)
  - 4 = Information-Report-Service

PNM7-Test-Cases

Gruppe:

- 6 = Context-Management
- 7 = Configuration-Management

PMS-PNM7-Test-Cases

Gruppe:

- 8 = PMS- und PNM7-Services

Der Testcase mit der Bezeichnung SVN10000 gehört demnach zur Klasse der SVN-Testcases, d.h. er testet die Funktionalität eines Servers über die PCP-ALI-Schnittstelle. Er gehört weiterhin zur Gruppe 1 und zur Subgruppe 0, d.h. er ist Bestandteil des Konformitätstests zum Context-Management, hier speziell des Initiate-Services.

6 Testablauf

6.1 Adaption Testsystem-Prüfling

Da die Spezifikationen dem Gerätehersteller viele Freiheiten bei seiner Implementierung lassen, müssen die Testcases sich an das real zu testende Gerät anpassen können. Pflichtfunktionen müssen von den Geräten immer erfüllt und damit auch getestet werden, optionale Funktionen können erfüllt und müssen auch nur dann getestet werden. Parameterfreiheitsgrade müssen generell beachtet werden. Der Testcase (bzw. der Testcase-Ersteller) entnimmt dieses Wissen aus der Kenntnis der Spezifikation und aus der Gerätebeschreibung in der PICS/PIXIT-Datei, abgebildet auf die 'pics'-Datenstruktur. Vor dem Starten eines Testcases müssen gegebenenfalls im Testsystem die Kommunikationsbeziehungsliste (KBL), das Objektverzeichnis (OV) und das VFD-Objekt an den Prüfling bzw. an den Testcase angepaßt werden.

6.1.1 Kommunikationsbeziehungsliste

Ausgangsbasis für alle PCP-Tests ist, daß mit Testbeginn die Kommunikationsreferenzen in der Kommunikationsbeziehungsliste des Testsystems zum Prüfling ein "spiegelbildliches" Abbild der Kommunikationsreferenzen in der Kommunikationsbeziehungsliste des Prüflings zum Testsystem sind (Bild 8).

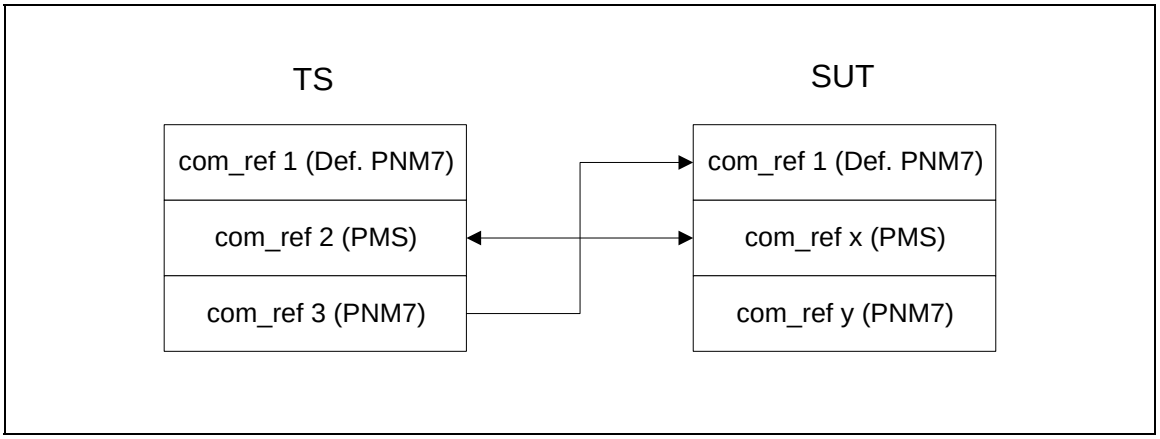

Bild 8: Beziehungen zwischen den Kommunikationsreferenzen von Testsystem und Prüfling

Das Attribut "remote" soll im folgenden immer den Prüfling und das Attribut "local" immer das Testsystem kennzeichnen.

Der Aufbau der Kommunikationsreferenzen im Testsystem geschieht wie folgt:

com_ref 0:

size	=	3	
poll_listen_ssap	=	NIL	
ass_abt_ci	=	0.9	* rem_time_out
symbol_length	=	0	
vfd_ptr_supp	=	0	

com_ref 1:

local_lsap	=	NIL
rem_addr	=	ALL
remote_lsap	=	NIL
conn_type	=	MMAZ
lli_sap	=	1
conn_attr	=	CONN_ATTR_O
max_scc	=	0
max_rcc	=	1
max_sac	=	0
max_rac	=	0
aci	=	0
multiplier	=	NIL
req_len_h	=	0
req_len_l	=	53
ind_len_h	=	0
ind_len_l	=	246
serv_sup	=	optionale Services werden nicht unterstützt
symbol	=	nicht belegt
vfd_pointer	=	0

com_ref 2:

local_lsap	=	NIL
rem_addr	=	PCP-Adresse des Prüflings
remote_lsap	=	NIL
conn_type	=	MMAZ
lli_sap	=	0
conn_attr	=	CONN_ATTR_D
max_scc	=	1
max_rcc	=	1
max_sac	=	1
max_rac	=	1
aci	=	aci (remote)
multiplier	=	NIL
req_len_h	=	0
req_len_l	=	ind_len_l (remote)
ind_len_h	=	0

ind_len_l	=	req_len_l (remote)
serv_sup	=	remote .req/.cnf werden zu local .ind/.res remote .ind/.res werden zu local .req/.cnf
symbol	=	nicht belegt
vfd_pointer	=	0

com_ref 3:

local_lsap	=	NIL
rem_addr	=	PCP-Adresse des Prüflings
remote_lsap	=	NIL
conn_type	=	MMAZ
lli_sap	=	1
conn_attr	=	CONN_ATTR_D
max_scc	=	1
max_rcc	=	0
max_sac	=	0
max_rac	=	0
aci	=	aci (remote)
multiplier	=	NIL
req_len_h	=	0
req_len_l	=	ind_len_l (remote)
ind_len_h	=	0
ind_len_l	=	req_len_l (remote)
serv_sup	=	remote .req/.cnf werden zu local .ind/.res remote .ind/.res werden zu local .req/.cnf
symbol	=	nicht belegt
vfd_pointer	=	0

Die Verträglichkeit des local PMS-und LLI-Contextes zum remote PMS-und LLI-Context ist mit dieser Abbildung sichergestellt (hier sogar Idealfall).

6.1.2 Objektverzeichnis

Das lokale Objektverzeichnis muß nur beim Client-Test an den jeweiligen CLN-Testcase angepaßt werden. Die Anpassung dient dazu, die geeigneten PMS-Objekte zum Test der Requests des Prüflings als Client bereitzustellen.

6.1.3 VFD-Objekt

Das lokale VFD-Objekt wird nur beim Client-Test zur Testmustergenerierung für die VFD-Dienste verändert.

6.2 Testcase-Bearbeitung

Bild 9 zeigt den prinzipiellen Ablaufrahmen in den Testcases eingebettet sind.

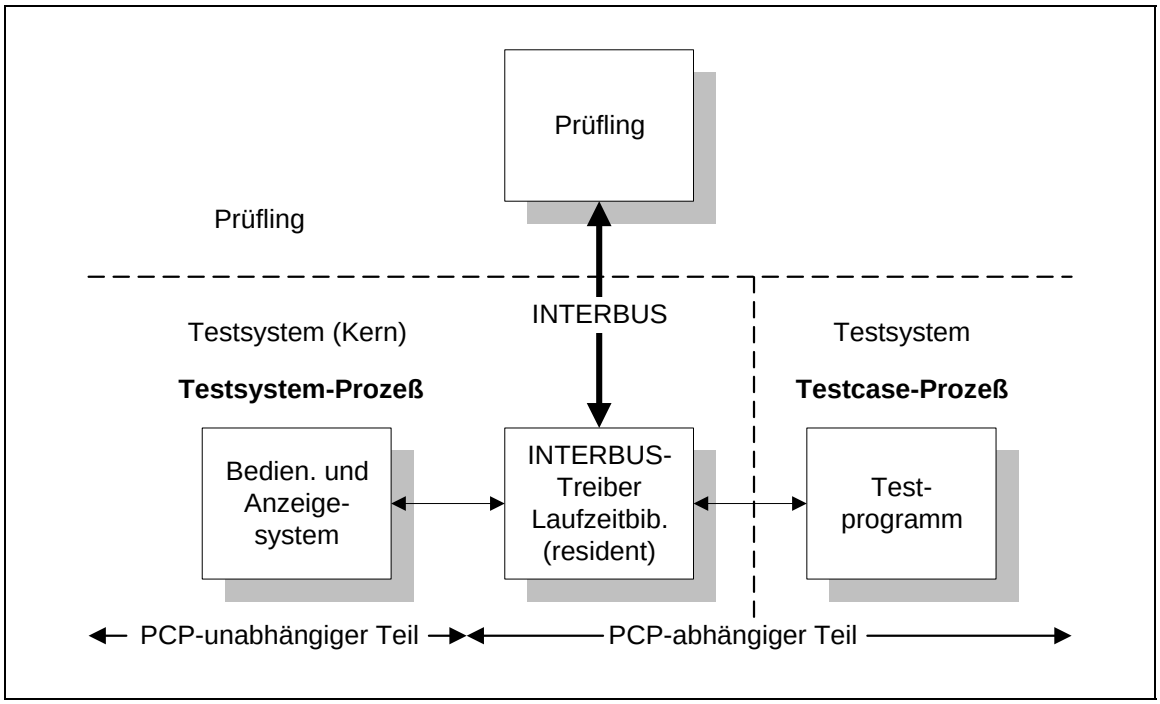

Bild 9: Prinzipieller Testcase-Ablaufrahmen

Das Testsystem besteht aus einem PCP-unabhängigen Bedien- und Anzeigesystem, einem PCP-abhängigen Teil, der im wesentlichen den Treiber für den INTERBUS beinhaltet und dem Testprogramm, das den Prüfling auf Konformität zu einer bestimmten Spezifikation testet.

6.2.1 Testsystem-Prozeß/Testcase-Prozeß

Prinzipiell könnte man alle Testfälle in ein einziges Testprogramm integrieren und über geschachtelte 'switch'-Anweisungen die einzelnen Fälle nacheinander abarbeiten. Dies bringt jedoch einige große Nachteile mit sich

- Speicherprobleme, weil der zur Verfügung stehende Speicher im Testsystem begrenzt ist,
- hohe Compilierungszeiten bei der Programmentwicklung und
- Zwang zur Neucompilierung des Testprogramms bei Änderungen im Bedien- und Anzeigesystem.

Beim INTERBUS-Testsystem wurden zur Vermeidung dieser Nachteile die einzelnen Testfälle als selbstständige Testcase-Programmeinheiten implementiert.

Im Testsystem sind während des Tests zwei Prozesse wechselweise aktiv:

– der Testsystem-Prozeß und

– der Testcase-Prozeß.

Der Testsystem-Prozeß, der das Bedien- und Anzeigesystem sowie den INTERBUS-Treiber beinhaltet, ist resident im Speicher des Testsystems geladen. Der Testcase-Prozeß wird vom Testsystem-Prozeß, entsprechend den Vorgaben des Bedieners, in den Speicher geladen und gestartet. Beide Prozesse kommunizieren miteinander indem sie sich gegenseitig sequentiell unter Übergabe von Parametern aufrufen. Am Testende gibt sich der Testcase-Prozeß selbst auf.

6.2.2 Aufrufschnittstellen

Zum Verständnis des Ablaufs eines Testcases innerhalb des Testsystems wird Bild 9 durch die Bilder 10 und 11 weiter detailliert.

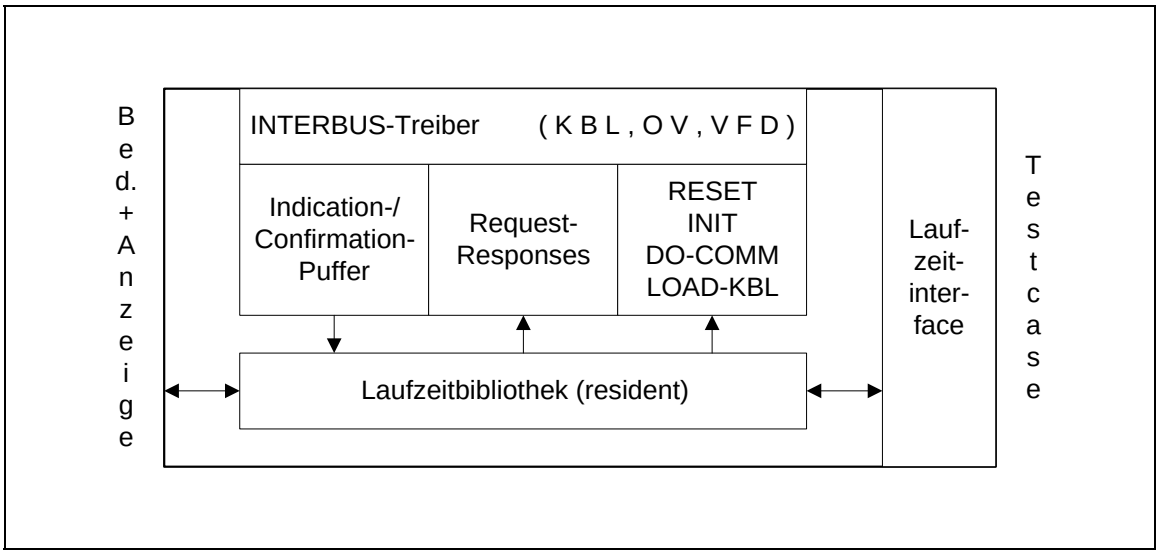

Bild 10: PCP-abhängiger Teil des residenten Testsystems

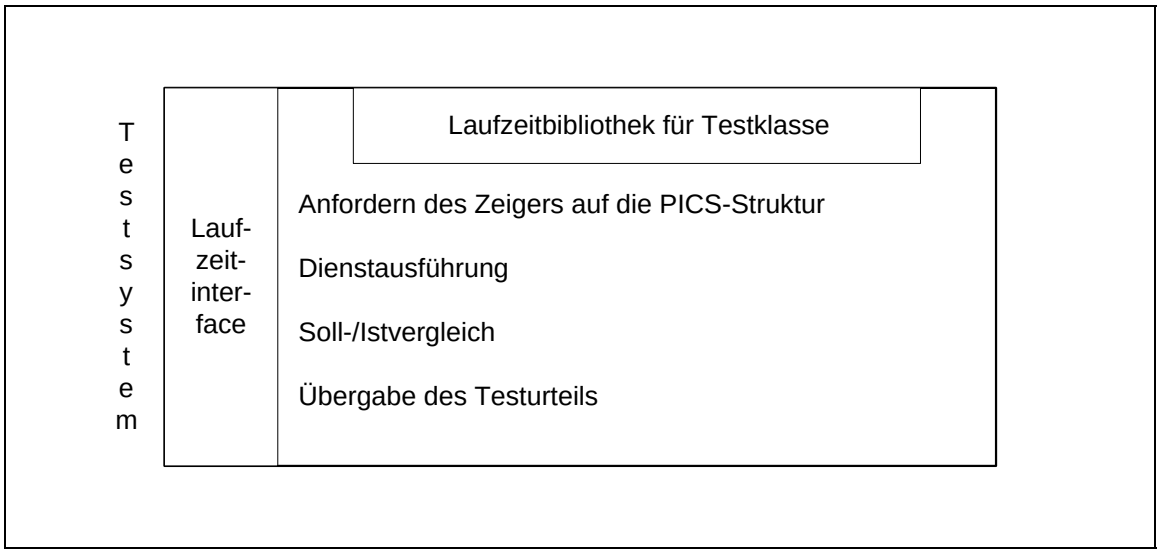

Bild 11: Testcase

Ein Testcase läuft im einfachsten Fall nach folgendem Schema ab:

- Anfordern des Zeigers auf die 'pics'-Struktur über das Laufzeitinterface und
- Entscheidung im Testcase durch Auswertung der 'pics'-Struktur ob und mit welchen Parametern der Test mit dem Prüfling durchzuführen ist. Kann der Test nicht durchgeführt werden, so wird das Testurteil 'Passed: Full restriction' an den Testsystem-Prozeß übergeben und der Testcase-Prozeß beendet.
- Stellen des Requests mit den vorbereiteten Dienstparametern und
- Anstoßen einer Warteschleife im Testsystem-Prozeß für die erhoffte Confirmation.
- Auswerten der empfangenen Confirmation mit Routinen aus der Laufzeitbibliothek im Testcase-Prozeß,
- Bildung des Testurteils
- und Übergabe dieses Urteils an den Testsystem-Prozeß.

6.2.3 Protokollierung

Ein Testcase-Ablauf wird über drei Wege protokolliert:

- Protokoll der Aufrufe an der ALI-Schnittstelle im Testsystem (1),

- Protokoll der Aufrufe zwischen Testcase und Laufzeitinterface (2)

- und Protokollierung des Testcase-Ablaufs projektiert durch den Testcase-Ersteller durch sogenannte Test-Messages. Die Test-Messages können wiederum unterschiedlichen Klassen angehören. Protokoll (3) durch Aufruf 'print_to_log' und Protokoll (4) durch Aufruf 'print_to_prof_log'.

Über die im Bedien- und Anzeigesystem einstellbaren 'Loggings' werden diese Protokolle zur Aufzeichnung und späteren Auswertung durch den Bediener genutzt und zwar

- TEST-LOG: 1+2+3,
- PMS-LOG: 1+3
- und PROF-LOG: 4

Jeder Protokolleintrag wird mit einem Zeitstempel versehen.

6.2.4 Fehlermeldungen

Ein Testcase wird bei Entdeckung eines Fehlers in der Regel unter Abgabe einer Fehlermeldung sofort abgebrochen (bei den Testklassen SVN und CLN ist dies generell der Fall). Im allgemeinen hat eine Fehlermeldung drei Komponenten. Einen Verweis auf den Fehlerort, z.B. den fehlerbehafteten Dienstparameter in Form des Namens, den Sollwert (mit vorangestelltem 'E: ' für expected) und den Istwert (mit vorangestelltem 'R: ' für received). Zum Beispiel:

Failed: locally_generated: E: ff R: 0

In diesem Fall wurde ein Fehler vom remote-Teilnehmer entdeckt und die Verbindung von diesem mit den Abort-Service abgebrochen anstatt vom lokalen Teilnehmer. Die Interpretation der Fehlermeldungen auf INTERBUS-Ebene setzt die detaillierte Kenntnis der PCP-Spezifikation mit allen Diensten und Parametern voraus.

6.3 Testurteile

Jeder Testcase prüft genau eine festumrissene Funktion. Er kann dabei mehrere Teilprüfungen (Subtests) beinhalten.

In den Testklassen SVN und CLN können Subtests aus weiteren Testabschnitten (in der Regel enthalten diese Datenmustertests) bestehen. Alle Tests werden hier mit allen in der PICS/PIXIT-Datei angegebenen Objekten, auch den nicht explizit projektierten Nullobjekten, durchgeführt. Das kostet zwar Rechenzeit, da eine Funktion an in der Eigenschaft ähnlichen Objekten mehrfach getestet werden kann, entdeckt jedoch unter Umständen Programmierfehler. Bei nicht existenten Objekten (Index liegt nicht im Bereich eines OV-Verzeich-

nisbereichs) wird jeder Test nur einmal durchgeführt. Datenmustertests werden in der Regel nur einmal durchgeführt.

Welche Tests auszuführen und welche nicht auszuführen sind, wird in jedem Testcase zur Laufzeit durch Auswertung der internen Datenstruktur 'pics' entschieden. Für jeden Testcase wird, unmittelbar nach Bearbeitung, im Testcase-Process ein Testurteil gebildet und, mit eventuellen Fehlermeldungen, an den Testsystem-Process weitergegeben. Es gibt drei grundsätzliche Testurteile:

- passed,
- failed und
- inconclusive.

Die Gruppe 0 der Testcases beim Server- und Client-Test prüft

- die Projektierung von KBL und OV auf "Unschönheiten", z.B. leere OV-Bereiche mit lauter Nullobjekten, sowie
- die Ausführbarkeit der Services bei der gegebenen bzw. remote ladbaren KBL-Projektierung und der gegebenen OV-Projektierung, wobei alle Objekte für die Services erreichbar sein sollten.

Gegenstand der Tests dieser Gruppe ist die Aufdeckung von möglichen Projektierungsunzulänglichkeiten. Da es sich um Tests handelt, die keine Fehlfunktionen im Sinne der PCP-Spezifikation überprüfen, werden nur Warnungen in Verbindung mit dem Testurteil "inconclusive" ausgegeben.

Die Gruppe 0 der Testcases beim Profil-Test überprüft die PICS-/PIXIT-Datei (insbesondere den profilspezifischen Teil) auf Vollständigkeit und Konsistenz in bezug auf das Profil.

6.3.1 Testurteil "passed"

Das Testurteil "passed" bedeutet, daß die Tests im Testcase keinen Fehler entdecken konnten. Dies ist nicht zu verwechseln mit der Aussage "der Prüfling hat alle im Testcase festgelegten Prüfungen ohne Fehler bestanden". Einige Beispiele:

- Testcases können Subtests beinhalten, die sich für ein Prüfobjekt gegenseitig ausschließen,

- Tests müssen für einen Prüfling nicht durchgeführt werden, wenn die zu prüfenden Funktionen optional sind und diese für den Prüfling nicht spezifiziert sind (z.B. optionale Services),

- Tests können nicht durchgeführt werden, wenn es keine entsprechenden Objekte im Objektverzeichnis des Prüflings gibt, die den Test ermöglichen.

Trotzdem gilt der Test in allen Fällen als bestanden ("passed"), da die zu testende Funktionalität im realen Betrieb des Gerätes nicht angesprochen wird.

Zur weiteren Wertung eines bestandenen Tests gibt es aber zusätzliche Angaben, die eine Aussage über den Abdeckungsgrad der implementierten bzw. nutzbaren Funktionalität im Prüfling zulassen:

- No restrictions,
- some restrictions (PICS: Anzahl),
- und full restriction (PICS).

Das Testurteil "Passed: No restrictions" bedeutet, daß alle im Testcase implementierten Tests ausgeführt und keine Fehler im Prüfling erkannt wurden.
Das Testurteil "Passed: Some restrictions" bedeutet, daß nicht alle im Testcase implementierten Tests ausgeführt wurden, aber die Tests, die ausgeführt wurden, keine Fehler im Prüfling erkannt haben. Die Anzahl der nicht ausgeführten Tests wird angegeben.

Das Testurteil "Passed: Full restriction" bedeutet, daß im Testcase keiner der implementierten Tests ausgeführt werden konnte.

6.3.2 Testurteil "failed"

Das Testurteil "failed" bedeutet, daß ein Test im Testcase einen Fehler entdeckt hat. Der Testcase wird unmittelbar nach Entdeckung des ersten Fehlers abgebrochen. Die Beseitigung eines Fehlers bedeutet damit nicht, daß der Testcase danach fehlerfrei abläuft, da bei Beseitigung des Fehlers erst alle nachfolgenden Tests aktiviert werden können.

6.3.3 Testurteil "inconclusive"

Das Testurteil "inconclusive" bedeutet, daß letztendlich keine automatische Aussage in Richtung "passed" oder "failed" getroffen werden konnte. Im Fall "inconclusive" ist es notwendig, daß ein Prüfer anhand der PICS-Datei und der Testprotokolle dieses Testurteil fällt. Das Testurteil "inconclusive" wird auch immer dann gemeldet, wenn zulässige Zugriffe auf Objekte des Objektverzeichnisses durch die Optionen NO_R, NO_W, NO_RW und NO_SH verboten wurden. Es gibt jedoch zusätzliche Angaben, die dem Prüfer die Entscheidung etwas erleichtern:

- Some restrictions (USER: Anzahl),
- some restrictions (PICS: Anzahl, USER: Anzahl)
- und full restriction (PICS, USER).

Das Testurteil "Inconclusive: Some restrictions (USER)" bedeutet daß alle im Testcase implementierten Tests ausgeführt und keine Fehler im Prüfling erkannt wurden, daß aber diese Tests nicht für alle in der PICS-Datei definierten Objekte durchgeführt wurden, auf die diese Tests hätten anwendbar sein können. Die Anzahl der Objekte, die der Ersteller der PICS-Datei (USER) vom Test ausgeschlossen hat, wird angegeben.

Das Testurteil "Inconclusive: Some restrictions (PICS, USER)" bedeutet, daß nicht alle im Testcase implementierten Tests ausgeführt wurden, aber die Tests, die ausgeführt wurden, keine Fehler im Prüfling erkannt haben. Auch hier wurden die Tests nicht für alle in der PICS-Datei definierten Objekte durchgeführt auf die diese Tests hätten anwendbar sein können. Im Gegensatz zum vorherigen Urteil wurden jedoch nicht alle implementierten Tests ausgeführt. Der Prüfer sollte hier nachsehen, ob sich der Abdeckungsgrad erhöht hätte, wenn Objekte nicht vom Test ausgeschlossen worden wären. Die Anzahl der nicht ausgeführten Tests und die Anzahl der Objekte, die der Ersteller der PICS-Datei (USER) vom Test ausgeschlossen hat, wird angegeben.

Das Testurteil "Inconclusive: Full restriction (PICS, USER)" bedeutet, daß im Testcase keiner der implementierten Tests ausgeführt werden konnte. Der Prüfer sollte hier nachsehen, ob Tests hätten durchgeführt werden können, wenn Objekte nicht vom Test ausgeschlossen worden wären. Die Anzahl der nicht ausgeführten Tests und die Anzahl der Objekte, die der Ersteller der PICS-Datei (USER) vom Test ausgeschlossen hat, wird angegeben.

Entscheidend für eine Abänderung des Testurteils von "inconclusive" nach "passed" durch den Prüfer ist in jedem Fall der Nachweis, daß der Abdeckungsgrad durch die Sperrung von Objekten für den Test nicht beeinflußt wurde.

Sonstige Testurteile mit "inconclusive" deuten darauf hin, daß ein Test wegen sonstiger Randbedingungen nicht ausgeführt werden konnte. Zum Beispiel:

- Ein Ereignis, das vom Testsystem nicht beeinflußbar ist, ist nicht aufgetreten, z.B. Erhalt einer Information-Report-Indication.

Der Prüfer muß die hier mit dem Testurteil "inconclusive" übergebene Meldung im Einzelfall bewerten und seine Entscheidung treffen.

6.4 Zertifizierungslauf

Der Zertifizierungslauf wird mit den Zertifizierungs-Schedule der jeweiligen Testklasse ausgeführt. Die Testcases im Zertifizierungs-Schedule werden in der Reihenfolge ihrer Einkettung ausgeführt und zu jedem Testcase ein Testurteil gebildet. Die einzelnen Testurteile werden in der 'result'-Datei als Testreport

abgelegt. Am Ende der Test-Schedule-Bearbeitung ist jedem ausgeführten Testcase eines der Urteile 'passed', 'failed' oder 'inconclusive' zugeordnet. Die Testcase-Urteile in der 'result'-Datei sind alphanumerisch geordnet. Testcases, die zwar der ausgeführten Testklasse angehören aber nicht im Zertifizierungs-Schedule eingekettet sind, erhalten kein Testurteil. Sie sind für die Erteilung des Zertifikats nicht relevant. Solche Testcases sind meist schon für spätere Versionen vorgesehen und können vorab schon für den Entwicklungstest genutzt werden. Der Testreport enthält neben den Testurteilen statistische Angaben zum Testablauf und die für den Test zugrunde gelegte Beschreibung des Prüflings in der PICS/PIXIT-Datei.

Zertifikate werden von der jeweiligen Zertifizierungsstelle unter Einreichung des Testreports (mit den zugeordneten Protokolldateien) des Testlabors erteilt. Ein getestetes Gerät erhält das Zertifikat für eine bestimmte Testklasse, wenn alle Testcases des Zertifizierungs-Schedules zu dieser Testklasse mit dem Testurteil 'passed' versehen sind. Das Zertifikat wird im allgemeinen nicht erteilt, wenn auch nur ein Testcase das Testurteil 'failed' besitzt. Beim Testurteil 'inconclusive' kann keine einfache Aussage getroffen werden. Hier muß die Zertifizierungsstelle, gegebenenfalls unter Einschaltung einer Schiedsstelle, fallweise entscheiden.

7 Anhang A: Testcaseliste für PCP 2.0

Bei der Auswahl des Zertifizierungsschedule sind folgende Eigenschaften des Prüflings zu beachten:

1. Prüfling ohne Remote Management
2. Prüfling mit Remote Management,
   keine Unterstützung des Service load_kbl_rem
3. Prüfling mit Remote Management,
   Unterstützung des Service load_kbl_rem

Für eine Konformitätsprüfung der PCP-Version 2.0 bei Prüflingen ohne Remote Management müssen folgende 241 Testcases durchlaufen werden:

1. CTPMS	33. SVN13120	65. SVN30530	97. SVN40110
2. SVN00010	34. SVN13140	66. SVN30540	98. SVN40120
3. SVN00020	35. SVN13220	67. SVN30550	99. SVN40130
4. SVN01000	36. SVN13230	68. SVN30560	100. SVN40140
5. SVN01010	37. SVN13240	69. SVN30570	101. SVN40160
6. SVN01100	38. SVN14000	70. SVN30580	102. SVN40170
7. SVN01110	39. SVN14010	71. SVN30590	103. SVN40180
8. SVN02000	40. SVN14020	72. SVN30600	104. SVN40200
9. SVN02010	41. SVN14030	73. SVN31010	105. SVN41000
10. SVN02020	42. SVN20000	74. SVN31030	106. SVN41020
11. SVN02030	43. SVN21000	75. SVN31050	107. SVN41030
12. SVN03000	44. SVN30000	76. SVN31070	108. SVN41040
13. SVN03010	45. SVN30020	77. SVN31080	109. SVN41100
14. SVN03020	46. SVN30040	78. SVN31090	110. SVN41110
15. SVN03030	47. SVN30060	79. SVN31110	111. SVN41120
16. SVN03040	48. SVN30070	80. SVN31130	112. SVN41130
17. SVN06000	49. SVN30080	81. SVN31150	113. SVN41140
18. SVN07000	50. SVN30100	82. SVN31170	114. SVN41160
19. SVN10000	51. SVN30120	83. SVN31190	115. SVN41170
20. SVN10020	52. SVN30140	84. SVN31270	116. SVN41180
21. SVN10030	53. SVN30160	85. SVN31280	117. SVN41200
22. SVN10050	54. SVN30180	86. SVN31290	118. SVN42000
23. SVN11000	55. SVN30260	87. SVN31310	119. SVN42020
24. SVN11010	56. SVN30270	88. SVN31330	120. SVN42030
25. SVN11040	57. SVN30280	89. SVN31350	121. SVN42040
26. SVN12080	58. SVN30300	90. SVN31410	122. SVN42100
27. SVN12100	59. SVN30320	91. SVN44200	123. SVN42110
28. SVN13000	60. SVN30340	92. SVN40000	124. SVN42120
29. SVN13010	61. SVN30400	93. SVN40020	125. SVN42130
30. SVN13020	62. SVN30420	94. SVN40030	126. SVN42140
31. SVN13030	63. SVN30430	95. SVN40040	127. SVN42160
32. SVN13040	64. SVN30520	96. SVN40100	128. SVN42170

129. SVN42180
130. SVN42200
131. SVN43000
132. SVN43020
133. SVN43030
134. SVN43040
135. SVN43100
136. SVN43110
137. SVN43120
138. SVN43130
139. SVN43140
140. SVN43160
141. SVN43170
142. SVN43180
143. SVN43200
144. SVN44000
145. SVN44010
146. SVN44050
147. SVN44100
148. SVN50000
149. SVN50020
150. SVN50030
151. SVN50040
152. SVN50100
153. SVN50110
154. SVN50120
155. SVN50130
156. SVN50140
157. SVN50150
158. SVN50200
159. SVN50210
160. SVN50220
161. SVN50230
162. SVN50240
163. SVN50250
164. SVN50300
165. SVN50320
166. SVN50330
167. SVN50340
168. SVN50400
169. SVN50500
170. SVN50600
171. SVN50700
172. SVN50800
173. SVN50810
174. SVN50820
175. SVN50830
176. SVN50860
177. SVN50870
178. SVN50880
179. SVN51000
180. SVN51020
181. SVN51030
182. SVN51040
183. SVN51060
184. SVN51100
185. SVN51110
186. SVN51120
187. SVN51130
188. SVN51140
189. SVN51150
190. SVN51160
191. SVN51200
192. SVN51210
193. SVN51220
194. SVN51230
195. SVN51240
196. SVN51250
197. SVN51260
198. SVN51300
199. SVN51320
200. SVN51330
201. SVN51340
202. SVN51360
203. SVN51370
204. SVN51400
205. SVN51410
206. SVN51430
207. SVN51500
208. SVN51510
209. SVN51530
210. SVN51540
211. SVN51550
212. SVN51600
213. SVN51610
214. SVN51630
215. SVN51640
216. SVN51650
217. SVN51700
218. SVN51710
219. SVN51730
220. SVN51800
221. SVN51810
222. SVN51820
223. SVN51830
224. SVN51860
225. SVN51870
226. SVN51880
227. SVN52000
228. SVN52020
229. SVN52030
230. SVN52040
231. SVN52400
232. SVN53000
233. SVN53020
234. SVN53030
235. SVN53040
236. SVN53060
237. SVN53400
238. SVN53440
239. SVN53450
240. SVN54000
241. SVN62150

Für eine Konformitätsprüfung der PCP-Version 2.0 bei Prüflingen mit Remote Management und keiner Unterstützung des Service load_kbl_rem müssen folgende 293 Testcases durchlaufen werden:

1. CTPMSM
2. SVN00010
3. SVN00020
4. SVN01000
5. SVN01010
6. SVN01100
7. SVN01110
8. SVN02000
9. SVN02010
10. SVN02020
11. SVN02030
12. SVN03000
13. SVN03010
14. SVN03020
15. SVN03030
16. SVN03040
17. SVN06000
18. SVN07000
19. SVN10000
20. SVN10020
21. SVN10030
22. SVN10050
23. SVN11000
24. SVN11010
25. SVN11040
26. SVN12080
27. SVN12100
28. SVN13000
29. SVN13010
30. SVN13020
31. SVN13030
32. SVN13040
33. SVN13120
34. SVN13140
35. SVN13220
36. SVN13230
37. SVN13240
38. SVN14000
39. SVN14010
40. SVN14020
41. SVN14030
42. SVN20000
43. SVN21000
44. SVN30000
45. SVN30020
46. SVN30040
47. SVN30060
48. SVN30070
49. SVN30080
50. SVN30100
51. SVN30120
52. SVN30140

53. SVN30160
54. SVN30180
55. SVN30260
56. SVN30270
57. SVN30280
58. SVN30300
59. SVN30320
60. SVN30340
61. SVN30400
62. SVN30420
63. SVN30430
64. SVN30520
65. SVN30530
66. SVN30540
67. SVN30550
68. SVN30560
69. SVN30570
70. SVN30580
71. SVN30590
72. SVN30600
73. SVN31010
74. SVN31030
75. SVN31050
76. SVN31070
77. SVN31080
78. SVN31090
79. SVN31110
80. SVN31130
81. SVN31150
82. SVN31170
83. SVN31190
84. SVN31270
85. SVN31280
86. SVN31290
87. SVN31310
88. SVN31330
89. SVN31350
90. SVN31410
91. SVN44200
92. SVN40000
93. SVN40020
94. SVN40030
95. SVN40040
96. SVN40100
97. SVN40110
98. SVN40120
99. SVN40130
100. SVN40140
101. SVN40160
102. SVN40170
103. SVN40180
104. SVN40200
105. SVN41000
106. SVN41020
107. SVN41030
108. SVN41040
109. SVN41100
110. SVN41110
111. SVN41120
112. SVN41130
113. SVN41140
114. SVN41160
115. SVN41170
116. SVN41180
117. SVN41200
118. SVN42000
119. SVN42020
120. SVN42030
121. SVN42040
122. SVN42100
123. SVN42110
124. SVN42120
125. SVN42130
126. SVN42140
127. SVN42160
128. SVN42170
129. SVN42180
130. SVN42200
131. SVN43000
132. SVN43020
133. SVN43030
134. SVN43040
135. SVN43100
136. SVN43110
137. SVN43120
138. SVN43130
139. SVN43140
140. SVN43160
141. SVN43170
142. SVN43180
143. SVN43200
144. SVN44000
145. SVN44010
146. SVN44050
147. SVN44100
148. SVN50000
149. SVN50020
150. SVN50030
151. SVN50040
152. SVN50100
153. SVN50110
154. SVN50120
155. SVN50130
156. SVN50140
157. SVN50150
158. SVN50200
159. SVN50210
160. SVN50220
161. SVN50230
162. SVN50240
163. SVN50250
164. SVN50300
165. SVN50320
166. SVN50330
167. SVN50340
168. SVN50400
169. SVN50500
170. SVN50600
171. SVN50700
172. SVN50800
173. SVN50810
174. SVN50820
175. SVN50830
176. SVN50860
177. SVN50870
178. SVN50880
179. SVN51000
180. SVN51020
181. SVN51030
182. SVN51040
183. SVN51060
184. SVN51100
185. SVN51110
186. SVN51120
187. SVN51130
188. SVN51140
189. SVN51150
190. SVN51160
191. SVN51200
192. SVN51210
193. SVN51220
194. SVN51230
195. SVN51240
196. SVN51250
197. SVN51260
198. SVN51300
199. SVN51320
200. SVN51330
201. SVN51340
202. SVN51360
203. SVN51370
204. SVN51400
205. SVN51410
206. SVN51430
207. SVN51500
208. SVN51510
209. SVN51530
210. SVN51540
211. SVN51550
212. SVN51600
213. SVN51610
214. SVN51630
215. SVN51640
216. SVN51650
217. SVN51700
218. SVN51710
219. SVN51730
220. SVN51800
221. SVN51810
222. SVN51820
223. SVN51830
224. SVN51860
225. SVN51870
226. SVN51880
227. SVN52000
228. SVN52020
229. SVN52030
230. SVN52040
231. SVN52400
232. SVN53000
233. SVN53020
234. SVN53030
235. SVN53040
236. SVN53060

237. SVN53400
238. SVN53440
239. SVN53450
240. SVN54000
241. SVN62150
242. SVN60000
243. SVN60020
244. SVN60050
245. SVN61000
246. SVN61040
247. SVN62080
248. SVN62100
249. SVN64000
250. SVN64010
251. SVN64020
252. SVN64030
253. SVN70000
254. SVN70010
255. SVN71000
256. SVN71100
257. SVN71110
258. SVN71120
259. SVN70020
260. SVN72000
261. SVN72010
262. SVN72030
263. SVN72100
264. SVN72110
265. SVN72120
266. SVN72020
267. SVN72130
268. SVN72140
269. SVN72150
270. SVN72160
271. SVN72170
272. SVN72180
273. SVN72190
274. SVN72200
275. SVN72210
276. SVN72220
277. SVN72240
278. SVN72250
279. SVN72260
280. SVN72270
281. SVN72280
282. SVN72290
283. SVN72310
284. SVN72320
285. SVN80000
286. SVN80010
287. SVN80020
288. SVN80030
289. SVN80070
290. SVN80080
291. SVN80100
292. SVN80110
293. SVN80300

Für eine Konformitätsprüfung der PCP-Version 2.0 bei Prüflingen mit Remote Management und Unterstützung des Service load_kbl_rem müssen folgende 323 Testcases durchlaufen werden:

1. CTPMSML
2. SVN00010
3. SVN00020
4. SVN01000
5. SVN01010
6. SVN01100
7. SVN01110
8. SVN02000
9. SVN02010
10. SVN02020
11. SVN02030
12. SVN03000
13. SVN03010
14. SVN03020
15. SVN03030
16. SVN03040
17. SVN06000
18. SVN07000
19. SVN60000
20. SVN60020
21. SVN60050
22. SVN61000
23. SVN61040
24. SVN62080
25. SVN62100
26. SVN64000
27. SVN64010
28. SVN64020
29. SVN64030
30. SVN70000
31. SVN70010
32. SVN71000
33. SVN71100
34. SVN71110
35. SVN71120
36. SVN70020
37. SVN72000
38. SVN72010
39. SVN72030
40. SVN72100
41. SVN72110
42. SVN72120
43. SVN72020
44. SVN72130
45. SVN72140
46. SVN72150
47. SVN72160
48. SVN72170
49. SVN72180
50. SVN72190
51. SVN72200
52. SVN72210
53. SVN72220
54. SVN72240
55. SVN72250
56. SVN72260
57. SVN72270
58. SVN72280
59. SVN72290
60. SVN72310
61. SVN72320
62. SVN80000
63. SVN80010
64. SVN80020
65. SVN80030
66. SVN80070
67. SVN80080
68. SVN80100
69. SVN80110
70. SVN80300
71. KBL0
72. SVN10000
73. SVN10020
74. SVN10030
75. SVN10050
76. SVN11000
77. SVN11010
78. SVN11040
79. SVN12080
80. SVN12100
81. SVN13000
82. SVN13010
83. SVN13020
84. SVN13030
85. SVN13040
86. SVN13120
87. SVN13140
88. SVN13220
89. SVN13230
90. SVN13240
91. SVN14000
92. SVN14010
93. SVN14020
94. SVN14030
95. SVN20000
96. SVN21000
97. SVN30000
98. SVN30020
99. SVN30040
100. SVN30060
101. SVN30070
102. SVN30080
103. SVN30100
104. SVN30120
105. SVN30140
106. SVN30160
107. SVN30180
108. SVN30260
109. SVN30270
110. SVN30280
111. SVN30300
112. SVN30320
113. SVN30340
114. SVN30400
115. SVN30420
116. SVN30430
117. SVN30520
118. SVN30530
119. SVN30540
120. SVN30550
121. SVN30560
122. SVN30570
123. SVN30580
124. SVN30590
125. SVN30600
126. SVN31010
127. SVN31030
128. SVN31050
129. SVN31070
130. SVN31080
131. SVN31090
132. SVN31110
133. SVN31130
134. SVN31150
135. SVN31170
136. SVN31190
137. SVN31270
138. SVN31280
139. SVN31290
140. SVN31310
141. SVN31330
142. SVN31350
143. SVN31410
144. SVN44200
145. SVN40000
146. SVN40020
147. SVN40030
148. SVN40040
149. SVN40100
150. SVN40110
151. SVN40120
152. SVN40130
153. SVN40140
154. SVN40160
155. SVN40170
156. SVN40180
157. SVN40200
158. SVN41000
159. SVN41020
160. SVN41030
161. SVN41040
162. SVN41100
163. SVN41110
164. SVN41120
165. SVN41130
166. SVN41140
167. SVN41160
168. SVN41170

169. SVN41180
170. SVN41200
171. SVN42000
172. SVN42020
173. SVN42030
174. SVN42040
175. SVN42100
176. SVN42110
177. SVN42120
178. SVN42130
179. SVN42140
180. SVN42160
181. SVN42170
182. SVN42180
183. SVN42200
184. SVN43000
185. SVN43020
186. SVN43030
187. SVN43040
188. SVN43100
189. SVN43110
190. SVN43120
191. SVN43130
192. SVN43140
193. SVN43160
194. SVN43170
195. SVN43180
196. SVN43200
197. SVN44000
198. SVN44010
199. SVN44050
200. SVN44100
201. SVN50000
202. SVN50020
203. SVN50030
204. SVN50040
205. SVN50100
206. SVN50110
207. SVN50120
208. SVN50130
209. SVN50140
210. SVN50150
211. SVN50200
212. SVN50210
213. SVN50220
214. SVN50230
215. SVN50240
216. SVN50250
217. SVN50300
218. SVN50320
219. SVN50330
220. SVN50340
221. SVN50400
222. SVN50500
223. SVN50600
224. SVN50700
225. SVN50800
226. SVN50810
227. SVN50820
228. SVN50830
229. SVN50860
230. SVN50870
231. SVN50880
232. SVN51000
233. SVN51020
234. SVN51030
235. SVN51040
236. SVN51060
237. SVN51100
238. SVN51110
239. SVN51120
240. SVN51130
241. SVN51140
242. SVN51150
243. SVN51160
244. SVN51200
245. SVN51210
246. SVN51220
247. SVN51230
248. SVN51240
249. SVN51250
250. SVN51260
251. SVN51300
252. SVN51320
253. SVN51330
254. SVN51340
255. SVN51360
256. SVN51370
257. SVN51400
258. SVN51410
259. SVN51430
260. SVN51500
261. SVN51510
262. SVN51530
263. SVN51540
264. SVN51550
265. SVN51600
266. SVN51610
267. SVN51630
268. SVN51640
269. SVN51650
270. SVN51700
271. SVN51710
272. SVN51730
273. SVN51800
274. SVN51810
275. SVN51820
276. SVN51830
277. SVN51860
278. SVN51870
279. SVN51880
280. SVN52000
281. SVN52020
282. SVN52030
283. SVN52040
284. SVN52400
285. SVN53000
286. SVN53020
287. SVN53030
288. SVN53040
289. SVN53060
290. SVN53400
291. SVN53440
292. SVN53450
293. SVN54000
294. SVN62150
295. KBL1
296. SVN10001
297. SVN10051
298. SVN11041
299. SVN13001
300. SVN13011
301. SVN13021
302. SVN13031
303. SVN13041
304. SVN13121
305. SVN13141
306. SVN13221
307. SVN13231
308. SVN13241
309. SVN14011
310. SVN20001
311. SVN21001
312. SVN30001
313. SVN30401
314. SVN31011
315. SVN31411
316. SVN50151
317. SVN50251
318. SVN51151
319. SVN51251
320. KBL2
321. SVN10002
322. SVN10032
323. SVN11002

8 Notizen