Das Entwurfssystem PSL/PSA und seine Erweiterung für Prozeß-Automatisierung

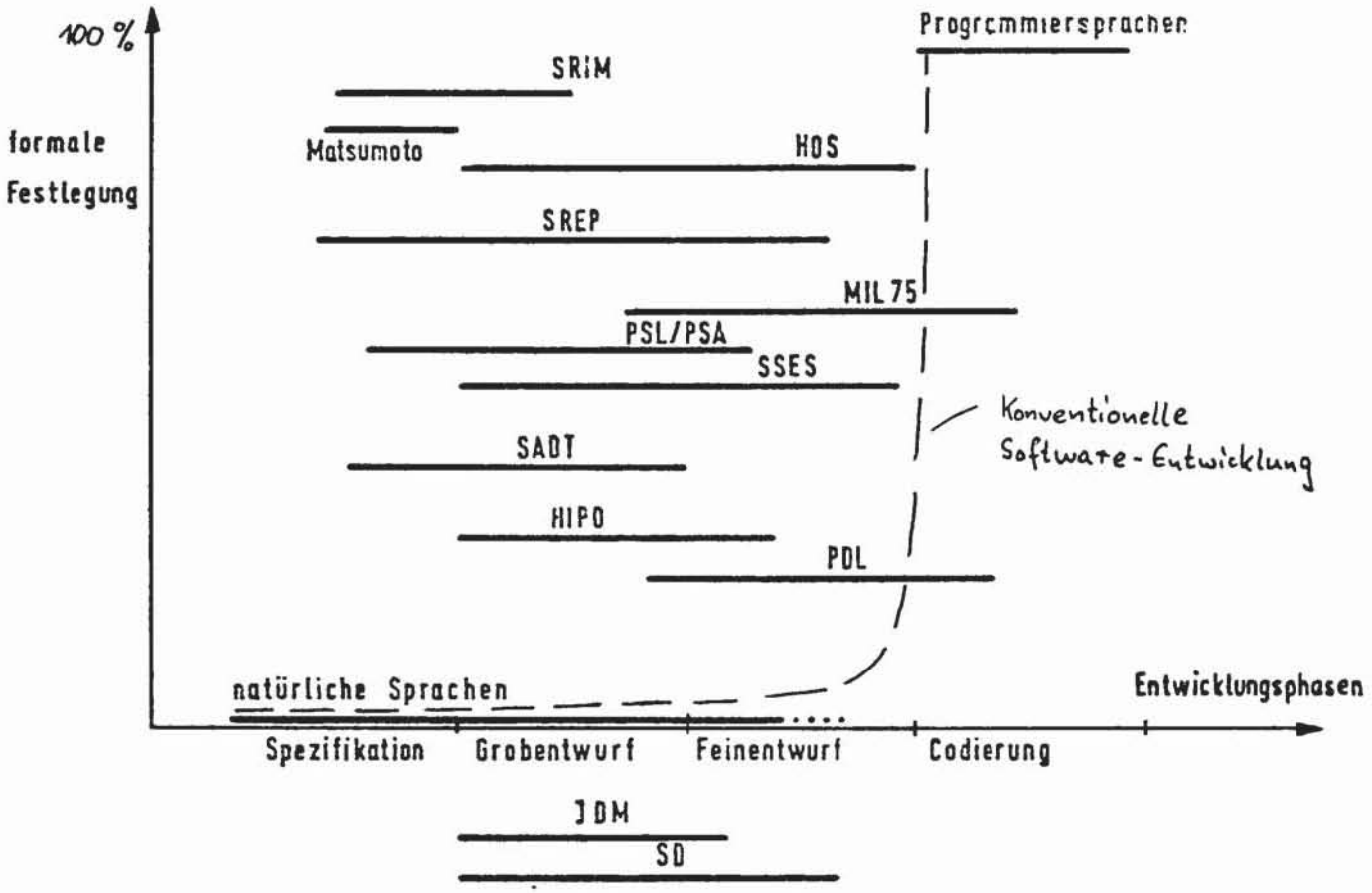
J. Ludewig, W. Streng

Kernforschungszentrum Karlsruhe GmbH
Institut für Datenverarbeitung in der Technik
Postfach 3640, 7500 Karlsruhe
Bundesrepublik Deutschland

Einleitung

Aufgrund von Fehleranalysen und Erfahrungen bei der Entwicklung großer Software-Systeme hat sich die Einsicht verbreitet, daß eine Erhöhung der Zuverlässigkeit vor allem durch eine Formalisierung und Methodisierung der Spezifikation und des Entwurfs zu erreichen ist. Erst durch formalisierte Beschreibungshilfsmittel wird eine rechnergestützte Software-Entwicklung und -Prüfung ermöglicht. Bisher ist der Anwender bei der Formulierung seiner Anforderungen auf natürliche Sprachen angewiesen. Dies führt häufig zu unvollständigen, mehrdeutigen und sogar widerspruchsvollen Spezifikationen, die dann die Basis für die Software-Entwicklung bilden. Enthält ein System systematische Fehler infolge einer falsch erkannten oder formulierten Aufgabenstellung, so können diese auch durch ausführliche Korrektheitsprüfungen nicht entdeckt werden.

Um zu prüfbaren Spezifikations- und Entwurfsbeschreibungen zu gelangen, wurden rechnergestützte Software-Entwicklungssysteme entworfen, die auf einem formalen Systembeschreibungsmodell beruhen. Angestrebt wird ein integriertes Werkzeug von aufeinander abgestimmten Hilfsmitteln zur Oberdeckung des gesamten Entwicklungszyklus. Dieser Zyklus wird als Transformationsprozeß zwischen Informationsräumen aufgefaßt, der verschiedene Phasen durchläuft. Abhängig von der Mächtigkeit des zugrundeliegenden Systembeschreibungsmodells und dessen Formalisierungsgrads lassen sich einige der bekannten Entwicklungshilfsmittel wie folgt den einzelnen Phasen zuordnen:



Der Formalisierungsgrad umfaßt hier neben der formalen Darstellbarkeit von Informationen auch das Maß an Vollständigkeit der in den betreffenden Phasen beschreibbaren Informationen.

Die Jackson Design Methodology (JDM) und Structured Design (SD) lassen sich nicht in dieses Diagramm einordnen, da sie nicht auf einem formalen Beschreibungsmodell aufbauen, sondern den methodischen Aspekt der Softwareentwicklung unterstützen.

Die Unterscheidung von Inhalt und Form der Informationsdarstellung (Beschreibungsmodell) und der Vorgehensweise zur Gewinnung dieser Information (Entwicklungsmethode) ermöglicht eine Charakterisierung der verfügbaren Entwicklungsmittel:

. Inhalt des Informationsraumes

  Unter Inhalt soll angegeben werden, was beschrieben werden soll, welche Informationen enthalten sein sollen, wie detailliert die Beschreibung sein muß, etc.

. Form der Informationsdarstellung

  Die Form bezieht sich auf die Art der Informationsdarstellung, z.B. graphisch oder linear.

. Entwicklungsmethode

  Unter Methode soll hier die Vorgehensweise bei der Systementwicklung verstanden werden, durch die man die zu beschreibenden Informationen über das System gewinnt. Eine Methode sollte somit den kreativen Prozeß der Systementwicklung unterstützen. Der Inhalt und die Form der Beschreibung sollten ihrerseits die Methode unterstützen.

Es zeigt sich, daß der überwiegende Teil der Entwicklungssysteme im wesentlichen Hilfsmittel zur Darstellung bereitstellt, während nur wenige Ansätze von einer Entwicklungsmethode ausgehen.

Ein Systembeschreibungsmodell definiert ein Schema, das einen Informationsraum für eine ganze Klasse von Systemen aufspannt. Umgekehrt kann ein System durch viele verschiedene Schemata beschrieben werden. Charakteristische Unterscheidungsmerkmale verschiedener Schemata sind ihr Abstraktionsgrad und die Systemaspekte, die beschrieben werden können. Ein Programm in einer Programmiersprache kann als Schema des Maschinencodes verschiedener Rechner betrachtet werden, da es von den Einzelheiten spezieller Maschineninstruktionen, Registerzuweisungen und Datenrepräsentationen abstrahiert. Ein Flußdiagrammschema erlaubt nur die Darstellung des Systemaspektes Kontrollfluß.

Bevor wir im folgenden einige Systemaspekte des Systembeschreibungsmodells von PSL ableiten, seien die Ziele unserer Arbeit kurz zusammengefaßt:

- Entwicklung formalisierter Beschreibungshilfsmittel für die Spezifikation und den Entwurf von Software für Prozeßrechneranwendungen (Spezifikations-/Entwurfssprache).

- Unterstützung der entwicklungsbegleitenden Dokumentation, d.h. alle in den einzelnen Entwicklungsschritten gewonnenen Informationen sollen in einer Datenbank abgespeichert werden.

- Rechnergestützte Prüfung der in den einzelnen Entwicklungsphasen gewonnenen Systembeschreibungen auf Vollständigkeit, Eindeutigkeit und Widerspruchsfreiheit (statische Analyse).

- Prüfung des dynamischen Systemverhaltens auf der Basis des Entwurfsmodells durch Simulation.

Das Systembeschreibungsmodell (SBM) von PSL

Das betrachtete SBM geht von der Annahme aus, daß der Inhalt des Informationsraums dargestellt werden kann durch

- eine Menge von Objekten und
- eine Menge von Relationen zwischen den Objekten.

Zur Identifizierung der Objekte und Relationen können diese Namen gegeben werden. Die Objekte können nach Objekttypen klassifiziert werden, und diesen können die Paare Attributname/Attributwertebereiche zur Beschreibung spezifischer Eigenschaften einer Objektklasse zugeordnet werden.

Eine bestimmte Klasse von Systemen (in PSL z.B. Informationssysteme) kann durch Spezialisierung dieses allgemeinen Modells charakterisiert werden, indem man die verfügbaren Objekttypen, Relationen und Attribute festlegt. Da noch keine allgemeine Theorie hierfür existiert, wird man sich dabei einerseits auf Erfahrungen stützen, die man bei der informellen Systemdefinition gemacht hat, andererseits auf Abstraktionen, die aus der Implementierung

von Systemen ableitbar sind. Zweckmäßigerweise wird man den komplexen Informationsraum, der von einem SBM aufgespannt wird, in Klassen (Systemaspekte) einteilen z.B. Datenfluß, Kontrollfluß, Systemstruktur, um die Modellentwicklung überschaubar zu machen.

Der Modellierungsprozeß zur Definition eines SBM zerfällt in folgende Schritte:

1. Erfassung des Informationsraums mit Hilfe der Grundelemente (Objekte und Relationen*)) und Benennung der Objekte
2. Bestimmung relevanter Merkmale der Grundelemente
3. Bildung disjunkter Typklassen der Grundelemente durch Abstraktionen
4. Weitere Klassifizierung der Objekttypen und Relationen nach Systemaspekten, wobei die Relationen disjunkten Systemaspekten zugeordnet werden.

Es sollte klar sein, daß ein so entwickeltes Modell nicht wahr oder falsch sein kann. Das Modell kann nur mehr oder weniger nützlich sein für den Zweck, für den es entwickelt wurde.

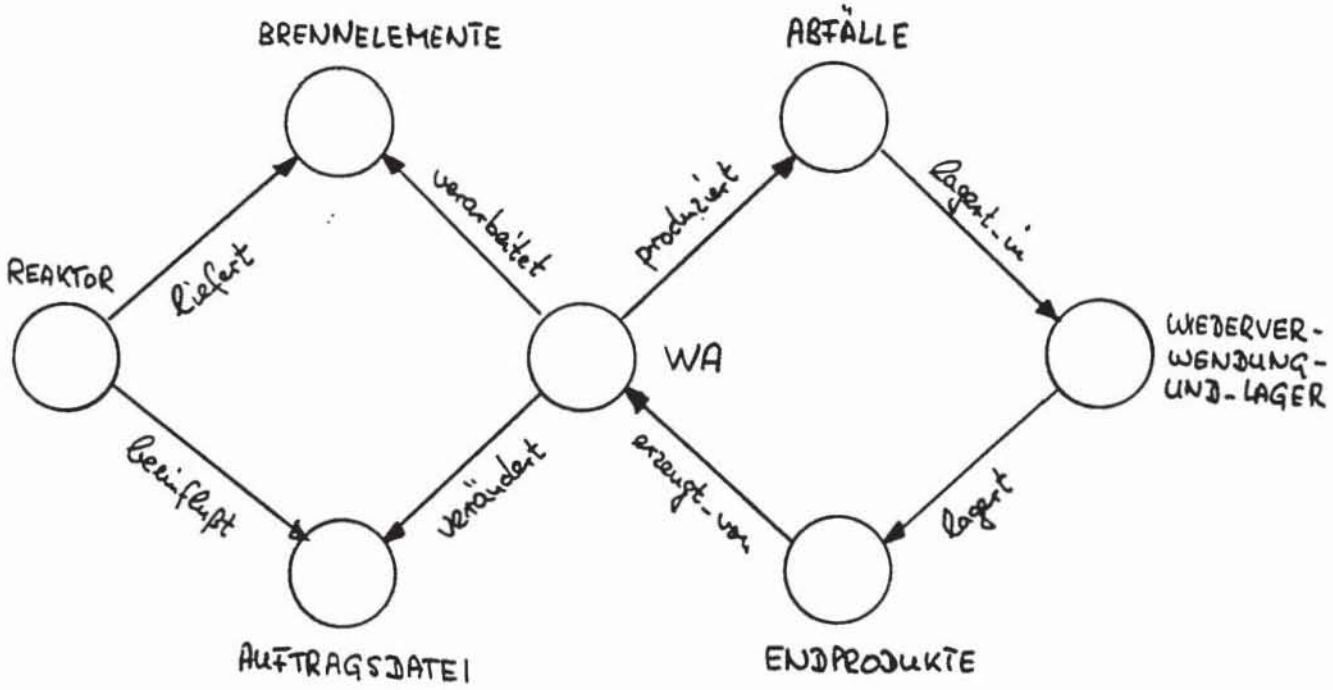
Nachfolgend soll gezeigt werden, wie ein PSL-ähnliches allgemeines Beschreibungsmodell aus der Darstellung eines speziellen Systems abgeleitet werden kann. Dabei wird zunächst nur ein bestimmter Aspekt betrachtet, die Verknüpfung mit der Umgebung.

Aufgabenstellung: Als Teil einer Wiederaufarbeitungsanlage (WA) soll ein Laborautomatisierungssystem (LA) entwickelt werden, das den Aufarbeitungsprozeß abgebrannter Brennelemente überwacht. Die WA verarbeitet die von einem Reaktor gelieferten Brennelemente. Dabei werden einerseits Endprodukte erzeugt, die zur Wiederverwendung gelagert werden, andererseits entstehen Abfälle, die ebenfalls sicher gelagert werden müssen. Das LA überwacht den Aufarbeitungsprozeß, indem an geeigneten Stellen Proben gezogen und analysiert werden. Abhängig von den Analyseergebnissen werden Steueraufträge für den Betrieb des Prozesses ausgegeben. Über die bearbeiteten Brennelemente soll in Form von Auftragslisten Buch geführt werden.

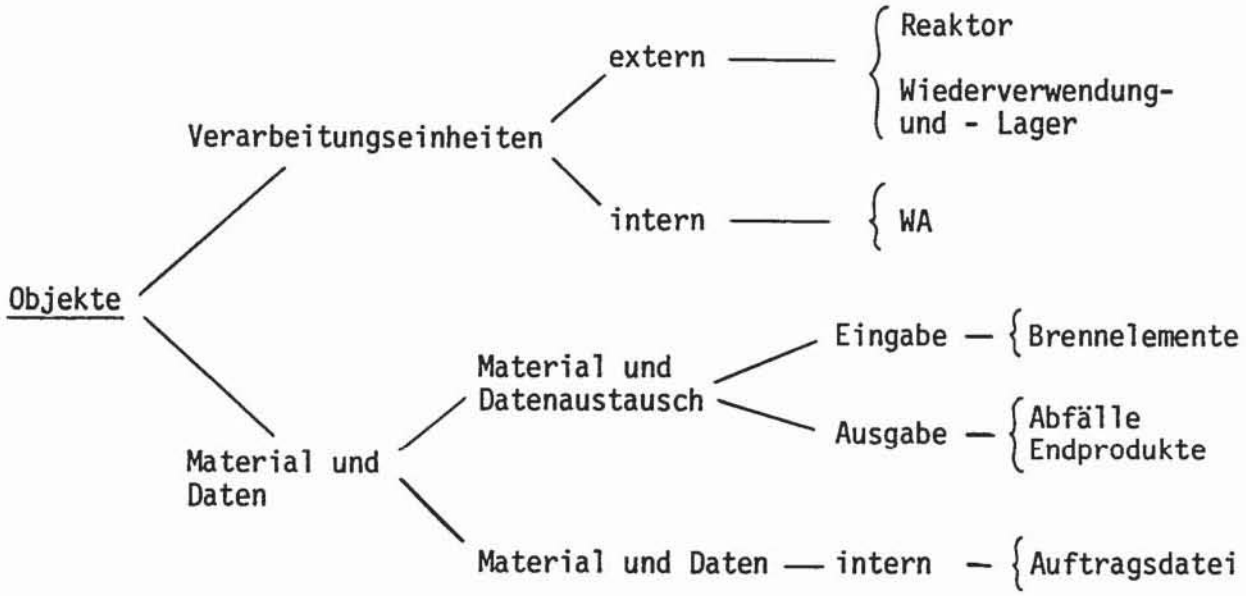
---

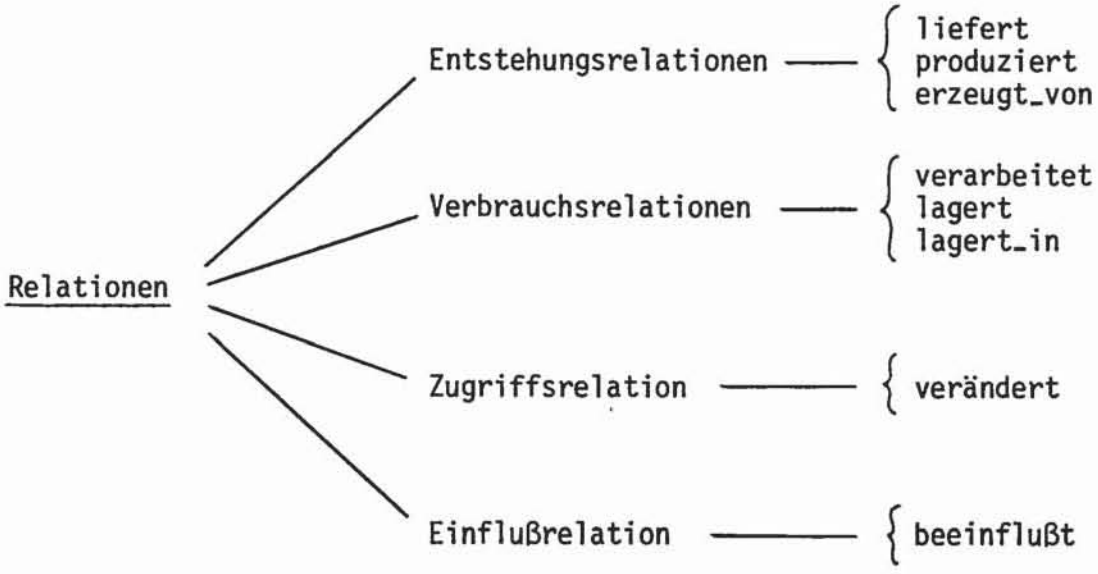
*) Wir beschränken uns im weiteren auf 2-stellige (binäre) Relationen, da mehrstellige Relationen durch Kombination von binären Relationen ausgedrückt werden können. PSL enthält auch 3-stellige Relationen.

Schritt 1: Die oberste Beschreibungsebene (Systemumgebung) läßt sich folgendermaßen darstellen:

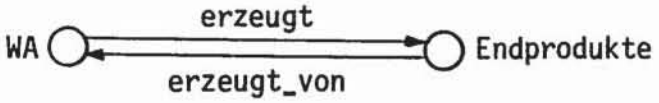


Schritt 2: Bestimmung relevanter Merkmale der Objekte und Relationen





Man erkennt, daß die Richtung der Relationspfeile im Beispiel beliebig gewählt wurde: dreht man z.B. den Pfeil "erzeugt_von" um, so ist die Beschreibung wieder korrekt, wenn man die Bezeichnung ersetzt durch "erzeugt". Damit ist eine gewisse Freiheit bei der Formulierung gegeben (eine Relation läßt sich aus zwei Perspektiven beschreiben).



Wo es sinnvoll erscheint, sollen deshalb auch sog. komplementäre Relationen definiert werden.

Schritt 3: Bildung von Typklassen (Syntax von PSL)

Einzelnen Unterbäumen (Pfaden) der obigen Entscheidungsbäume werden Typen zugeordnet.

Objekttypen:

REAL-WORLD-ENTITY	=	{Reaktor, Wiederverwendung-und-Lager}
PROCESS	=	{WA}
INPUT	=	{Brennelemente}
OUTPUT	=	{Abfälle, Endprodukte}
SET	=	{Auftragsdatei}

Relationstypen:			komplem. Relationstypen:	
GENERATES	=	{liefert, produziert}	GENERATED BY	= {erzeugt_von}
RECEIVES	=	{verarbeitet, lagert}	REVEICED BY	= {lagert_in}
UPDATES	=	{verändert}	UPDATED BY	
RESPONSIBLE FOR	=	{beeinflußt}	RESPONSIBLE-INTERFACE	

Die Relationstypen sind nur definiert zwischen bestimmten Objektklassen, z.B.

GENERATES $\subset$ REAL-WORLD-ENTITY x INPUT $\cup$ PROCESS x OUTPUT

(siehe Tabelle auf Seite 9) .

Schritt 4: Die eingeführten Objekte und Relationen beschreiben die Schnittstelle eines Systems in seiner Umgebung. Wir nennen diesen Systemaspekt SYSTEMUMGEBUNG .

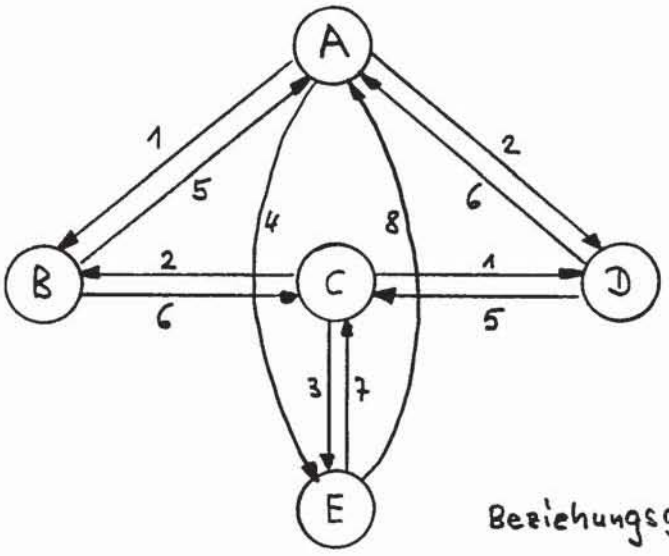
In PSL wird der Systemaspekt SYSTEMUMGEBUNG durch folgende Mengen der Objekt- und Relationstypen beschrieben.

Objekttypen

A - REAL-WORLD-ENTITY
B - INPUT
C - PROCESS
D - OUTPUT
E - SET

Relationen

1 - GENERATES
2 - RECEIVES
3 - UPDATES
4 - RESPONSIBLE FOR
5 - GENERATED BY
6 - RECEIVED BY
7 - UPDATED BY
8 - RESPONSIBLE-INTERFACE



Beziehungsgraph

	A	B	C	D	E
A		1		2	4
B	5		6		
C		2		1	3
D	6		5		
E	8		7		

Objektmatrix

Der Beziehungsgraph und die Objektmatrix definieren die erlaubten Relationen zwischen Objekten.

Formulierung der SYSTEM UMGEBUNG unseres Beispiels in PSL:

```
      PSA Version A4.2R1                                          78.151  09.58
                                BRLUST  IDT/KFK   PSA V4.2

                                     Input Source Listing

LINE  S T M T

  94 >                                                                      <
  95 >                                                                      <
  96 >/*--------------------------*/                                        <
  97 >/* EIN-AUSGABE SCHNITTSTELLE */                                       <
  98 >/*--------------------------*/                                        <
  99 >                                                                      <
 100 >                                                                      <
 101 >                                                                      <
 102 >/* OBERSTE (GLOBALE) BESCHREIBUNGSEBENE */                           <
 103 >                                                                      <
 104 >PROCESS : WIEDERAUFARBEITUNGSANLAGE;                                  <
 105 >      SYNONYMS ARE WA, AO;                                           <
 106 >      DESC;                                                          <
 107 >      DIE WA BESTEHT AUS DEM TECHNISCHEN PROZESS                     <
 108 >      UND DER PROZESSFUEHRUNG, DIE NACHFOLGEND                       <
 109 >      NAEHER BESCHRIEBEN WIRD. DIE WA IST EINE STATION               <
 110 >      IM KERNBRENNSTOFF-KREISLAUF.                                   <
 111 >      ;                                                              <
 112 >                                                                      <
 113 >INPUT :  BRENNELEMENTE;                                              <
 114 >      GENERATED BY REAKTOR;                                          <
 115 >      RECEIVED  BY WA;                                               <
 116 >      DESC;                                                          <
 117 >      MATERIALFLUSS IN DIE WA;                                       <
 118 >                                                                     <
 119 >OUTPUTS : ENDPRODUKTE, ABFAELLE;                                     <
 120 >      GENERATED BY WA;                                               <
 121 >      RECEIVED  BY WIEDERVERWENDUNG-UND-LAGER;                       <
 122 >      DESC;                                                          <
 123 >      MATERIALFLUSS AUS DER WA;                                      <
 124 >                                                                     <
 125 >REAL-WORLD-ENTITY : REAKTOR,                                         <
 126 >                    WIEDERVERWENDUNG-UND-LAGER;                      <
 127 >                                                                     <
 128 >SET : AUFTRAGSDATEI;                                                 <
 129 >    UPDATED BY WA;                                                   <
 130 >    RESPONSIBLE-INTERFACE IS REAKTOR;                                <
 131 >                                                                     <
 132 >                                                                     <
 133 >                                                                     <
```

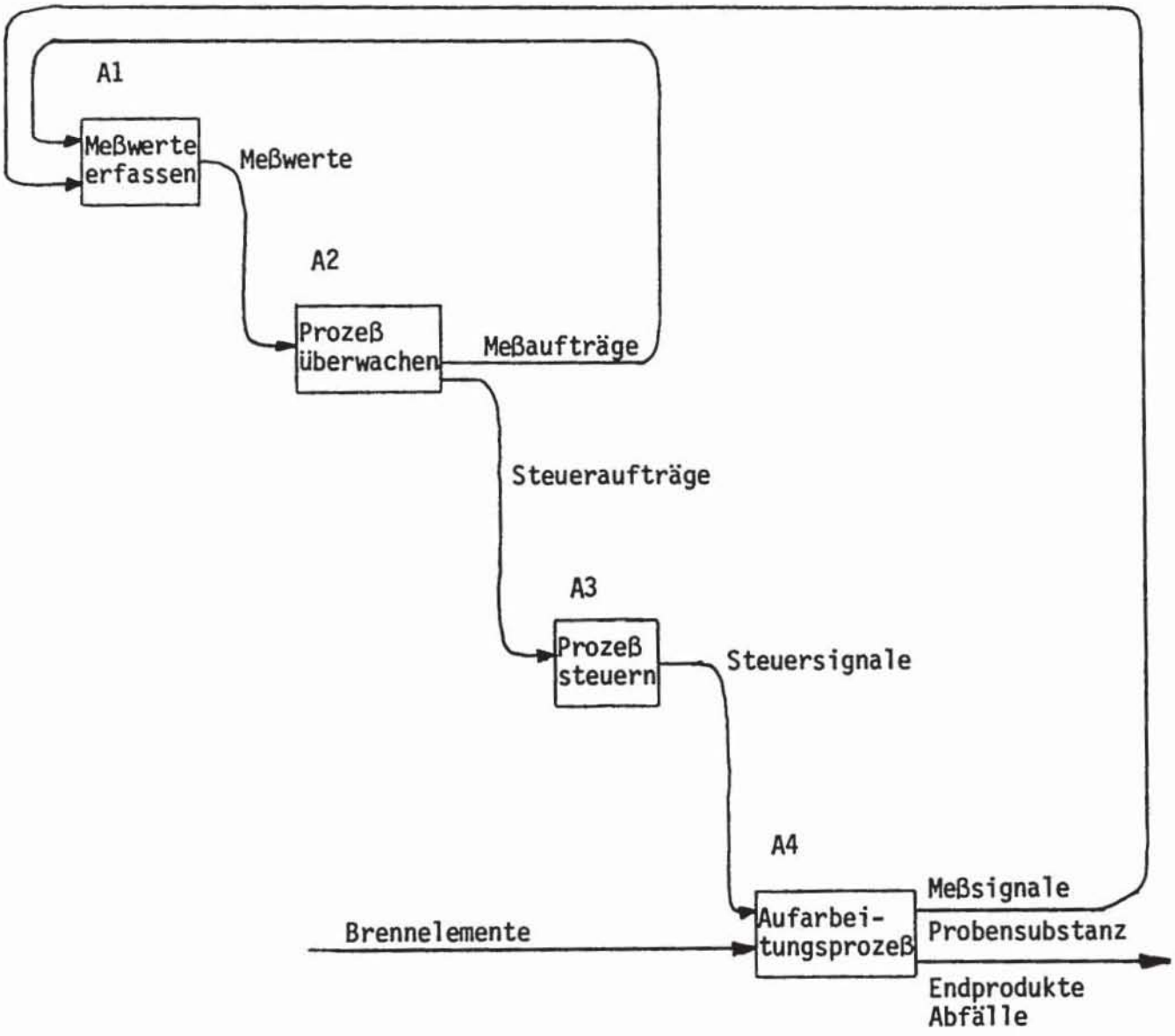
Nach der Beschreibung der Grobstruktur unseres Entwurfsbeispiels wäre die Verfeinerung des PROCESS "Wiederaufarbeitungsanlage" notwendig. Ebenso könnten die Daten verfeinert und die Flüsse näher beschrieben werden. PSL stellt dafür weitere Sprachmittel zur Verfügung, die nach folgenden Systemaspekten gegliedert werden können:

Systemaspekte:

SYSTEM STRUKTUR	:	beschreibt die statische Zerlegungsstruktur der Objekte im System. Baumstrukturen und gerichtete Netzwerke können beschrieben werden.
DATEN STRUKTUR	:	legt die Beziehungen zwischen den Daten fest (Netzwerke, Bäume).
DATEN TRANSFORMATION	:	beschreibt die Datentransformationen durch die Prozesse auf verschiedenen Abstraktionsebenen.
SYSTEM UMFANG	:	legt quantitative Aspekte des Systems wie Laufhäufigkeit und Datenumfang fest.
DYNAMISCHES VERHALTEN	:	führt Ereignisse und auslösende Bedingungen für die Prozesse als dynamischen Aspekt des Systems ein.
SYSTEM EIGENSCHAFTEN	:	erlaubt die Vergabe beliebiger Eigenschaftsattribute für Objekte, z.B. Synonyma, informelle Textbeschreibungen.
PROJEKT MANAGEMENT	:	umfaßt Informationen die für den Projektverlauf von Bedeutung sind, z.B. die Verantwortlichkeit bestimmter Personen für bestimmte Teilaufgaben.

Beispielhaft sollen jetzt Ausschnitte aus der weiteren Verfeinerung unseres Beispiels angegeben werden, um einen Eindruck von den oben erläuterten Systemaspekten zu bekommen.

Die Wiederaufbereitungsanlage bestehe aus den folgenden Teilprozessen und ihren in SADT-Diagrammform notierten Beziehungen:



In PSL lassen sich diese Informationen durch die Systemaspekte SYSTEM STRUKTUR und DATEN TRANSFORMATION beschreiben:

```
510 >
511 >/*-----------------------*/
512 >/* DATEN TRANSFORMATIONEN */
513 >/*-----------------------*/
514 >
515 >
516 >
517 >
518 >/* BESCHREIBUNGSEBENE A.0, STUFE 1 */
519 >
520 >/* PROZESS FUEHREN */
521 >
522 >PROCESS : WA;                                    } SYSTEM STRUKTUR
523 >     SUBPARTS ARE :
524 >          MESSWERTE-ERFASSEN,
525 >          PROZESS-UEBERWACHEN,
526 >          PROZESS-STEUERN,
527 >          AUFARBEITUNGSPROZESS;
528 >     KEYWORD : STUFE-1;
529 >
530 >PROCESS : MESSWERTE-ERFASSEN;
531 >     SYNONYM : A1;
532 >     USES    : MESSAUFTRAEGE, MESS-SIGNALE, PROBENSUBSTANZ;
533 >     DERIVES : MESSWERTE;
534 >
535 >PROCESS : PROZESS-UEBERWACHEN;
536 >     SYNONYM : A2;
537 >     USES    : MESSWERTE;
538 >     DERIVES : STEUERAUFTRAEGE;
539 >     UPDATES : MESSAUFTRAEGE;
540 >
541 >PROCESS : PROZESS-STEUERN;
542 >     SYNONYM : A3;
543 >     USES    : STEUERAUFTRAEGE;
544 >     DERIVES : STEUERSIGNALE;
545 >     DESCRIPTION;
546 >          DIE PROZESS-STEUERUNG WIRD DV-MAESSIG NICHT UNTERSTUETZT.
547 >          NACHFOLGEND WIRD DIESE NICHT WEITER VERFEINERT.;
548 >
549 >PROCESS : AUFARBEITUNGSPROZESS;
550 >     SYNONYMS ARE : PROZESS, A4;
551 >     USES         : STEUERSIGNALE, BRENNELEMENTE;
552 >     DERIVES      : MESS-SIGNALE, PROBENSUBSTANZ,
553 >                    ENDPRODUKTE, ABFAELLE;
554 >     DESCRIPTION;
555 >          DER AUFARBEITUNGSPROZESS IST DER TECHNISCHE TEIL DER ANLAGE.
556 >          NACHFOLGEND WIRD DIESER NICHT WEITER VERFEINERT.;
```

Ausschnitt aus der Beschreibung des dynamischen Verhaltens unseres Beispielsystems:

```
1463 >
1464 >EVENT : MESSWERTEBESTIMMUNG ;
1465 >    TRIGGERS : A1-41;
1466 >    WHEN  MESS-SIGNALE-BEREIT  BECOMES TRUE ;
1467 >
1468 >CONDITION : MESS-SIGNALE-BEREIT ;
1469 >    FALSE WHILE;
1470 >        DIESE BEDINGUNG IST FALSCH BIS ALLE MESS-SIGNALE AUS
1471 >        DEM PROZESS ZUR VERFUEGUNG STEHEN ;
1472 >    TRUE WHILE;
1473 >        MESS-SIGNALE STEHEN ZUR VERARBEITUNG BEREIT ;
1474 >
1475 >EVENT : MAGAZIN-LEER ;
1476 >    DESCRIPTION;
1477 >        DIESES EREIGNIS STEUERT DAS NACHFUELLEN DES MAGAZINS,
1478 >        DAS FASSUNGSVERMOEGEN DES MAGAZINS IST 30 PROBENFLASCHEN ;
1479 >    TRIGGERS : A1-2232 ;
1480 >    ON TERMINATION OF : A1-2231 ;
1481 >    WHEN  MAGAZINBELEGUNGSTEST  BECOMES FALSE ;
1482 >    HAPPENS  MAGAZIN-LEER-HAEUFIGKEIT  TIMES-PER  TAG ;
1483 >
1484 >CONDITION : MAGAZINBELEGUNGSTEST ;
1485 >
1486 >    TRUE WHILE;
1487 >        MAGAZINBELEGUNG  GROESSER  MAGAZIN-MINIMUM ;
1488 >    FALSE WHILE;
1489 >        MAGAZINBELEGUNG  KLEINER-GLEICH  MAGAZIN-MINIMUM ;
1490 >
1491 >CONDITION : AUFTRAGSART ;
1492 >    DESC;
1493 >        AUSWAHL DER PROBENNAHMEAUFTRAGS-BEARBEITUNG ;
1494 >    BECOMING TRUE IS CALLED : AUTOMATISCH ;
1495 >    BECOMING FALSE IS CALLED  MANUELL ;
1496 >
1497 >EVENT : AUTOMATISCH ;
1498 >    TRIGGERS : A1-223 ;
1499 >
1500 >EVENT : MANUELL ;
1501 >    TRIGGERS : A1-225 ;
1519 >EVENT : PROTOKOLLANFORDERUNGEN ;
1520 >    TRIGGERS : A2-43, A2-431 ;
1521 >    ON TERMINATION OF : A2-41, A2-42 ;
1522 >    ATTRIBUTES ARE : ANFORDERUNG1 PERIODISCH,
1523 >                     ANFORDERUNG2 EXTERN ;
1569 >PROCESS : A1-2 ;
1570 >    HAPPENS :  200  TIMES-PER  TAG ;
1605 >PROCESS : A2-33 ;
1606 >    HAPPENS :  1  TIMES-PER  STATUS-PRUEFUNGS-ZYKLUS  ;
```

Ausschnitt aus der Beschreibung der SYSTEM EIGENSCHAFTEN :

```
1671 >/*----------------------*/
1672 >/* SYSTEM-EIGENSCHAFTEN */
1673 >/*----------------------*/
1674 >
```

```
1691 >PROCESS : A1, A2, A3 ;
1692 >    ATTRIBUTES ARE : ABLAUF ZYKLISCH ;
1693 >
1694 >ENTITY : A311-AUFTRAG ;
1695 >    ATTRIBUTE IS : BEARBEITUNGSZUSTAND  Z1-311 ;
1696 >
```

```
1706 >MEMO : PROBENNAHME-KOMMENTAR ;
1707 >    DESCRIPTION;
1708 >        ZU JEDER IM LABOR IN BEARBEITUNG ODER ZUR AUFBEWAHRUNG
1709 >        BEFINDLICHEN PROBE ENTHAELT EIN EINTRAG DIE KENNUNG,
1710 >        DEN BEARBEITUNGSSTATUS, ROHERGEBNISSE U.A.  ;
1711 >    APPLIES TO : PROBENNAHMEAUFTRAG,
1712 >                 P-221, P-223, P-231, P-232, P-2231,
1713 >                 AUT-P,  AND  MAN-P ;
```

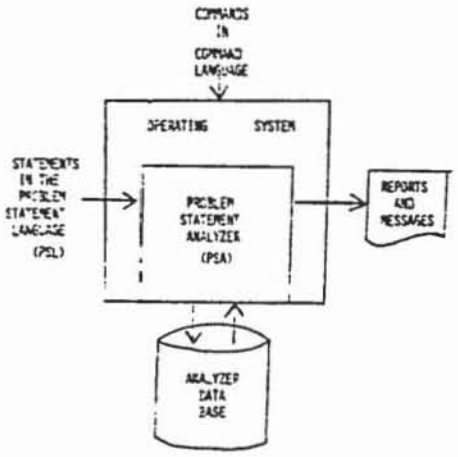
```
1731 >GROUP : DIREKT-ERFASSTE-MESSWERTE ;
1732 >    THE ATTRIBUTES ARE  FUELLSTAENDE ANZEIGEN,
1733 >                        TEMPERATUR ANZEIGEN,
1734 >                        DRUCK ANZEIGEN,  AND
1735 >                        DURCHFLUSS MENGEN    ;
1736 >
1737 >GROUP : BEWERTETER-PROZESSZUSTAND ;
1738 >    ATTRIBUTE IS : BEARBEITUNGSZUSTAND Z2-21 ;
1739 >
1740 >ELEMENT : DATUM ;
1741 >    ATTRIBUTE IS : FORMAT  TAG-MONAT-JAHR  ;
```

Ausschnitt aus der Beschreibung des Systemaspektes SYSTEM UMFANG :

```
1610 >/*--------------------------------*/
1611 >/* SYSTEMGROESSE-AUFWANDSPARAMETER */
1612 >/*--------------------------------*/
1613 >
1614 >
1615 >
1616 >
1617 >SET : MESSAUFTRAEGE ;
1618 >    CARDINALITY IS : ANZAHL-WARTENDER-MESSAUFTRAEGE ;
1619 >
1620 >ELEMENT : PROBENNEHMER-NR ;
1621 >    VALUES ARE : 1  THRU  18  ;
1622 >
1623 >ELEMENT : LAUFENDE-NUMMER ;
1624 >    VALUES ARE : 1  THRU  99999  ;
1625 >
1626 >RELATION : ANALYSE-PROBEN-RELATION ;
1627 >    CONNECTIVITY : EINS  TO  MEHRERE  ;
1628 >    CARDINALITY  : PROBENZAHL  ;
1629 >
1630 >DEFINE : METHODENANZAHL  SYSTEM-PARAMETER ;
1631 >    DESCRIPTION;
1632 >        ANZAHL DER ANALYSENMETHODEN ;
1633 >    VALUE IS : 30 ;
1634 >
1635 >INTERVAL : STATUS-PRUEFUNGS-ZYKLUS ;
1636 >    SYNONYM : ZEIT ;
1637 >    CONSISTS OF : MEHRERE  MINUTEN ;
1638 >
1639 >ELEMENT : DRINGLICHKEIT ;
1640 >    VALUES ARE : 1  THRU  3  ;
1641 >
1642 >ELEMENT : PROBENFLASCHE ;
1643 >    ATTRIBUTE IS : VOLUMEN MILLILITER-3-4 ;
```

Rechnerunterstützte Prüfungen

Das System PSL/PSA besteht, wie der Name sagt, aus der formalen Sprache PSL, die oben skizziert wurde, und dem Analysator PSA . Der in PSL beschriebene Systementwurf wird von PSA verarbeitet, in einer Datenbank abgespeichert und analysiert.



Die mit dem Analysator durchführbaren Prüfungen lassen sich in drei Klassen einordnen:

Syntaktische Prüfungen

Diese Analysen sollen die korrekte Verwendung von PSL sicherstellen:

Namenskonflikte und nichterlaubte Relationen zwischen bestimmten Objekttypen werden aufgedeckt. Implizit ist jeder Relation in PSL eine Korrespondenz zugeordnet. Diese gibt an, wieviele Elemente des Definitionsbereichs mit wievielen Elementen des Wertebereichs zu einem Zeitpunkt verknüpft sein dürfen. Man unterscheidet (1:1), (1:N), (N:1) und (N:N)-Ver-

knüpfungen. Z.B. muß die PART OF Relation zwischen zwei PROCESSes vom Verknüpfungstyp (N:1), also eine Abbildung im mathematischen Sinne sein, weil die definierte Systemstruktur eine Baumhierarchie sein soll. Diese impliziten Korrespondenzen werden ebenfalls geprüft.

Konsistenz- und Vollständigkeitsprüfungen

Die Konsistenzanalysen beruhen auf Grundhypothesen, die in Form von Konsistenzregeln den einzelnen Systemaspekten zugeordnet sind. In dem Beziehungsgraph eines Systemaspektes wurden alle erlaubten Relationen dargestellt; dies sind alle Relationen, die in dem SBM existieren können. Durch Konsistenzregeln werden nun für die einzelnen Systemaspekte die Relationen definiert, die existieren sollten, um die Vollständigkeit einer Systembeschreibung zu gewährleisten.

Konsistenzregeln können folgende Form haben:

<rule> ::= <each-statement>/<if-statement>

<each-statement> ::= FOR EACH <object-name> THERE

{MUST / CAN / CANNOT} EXIST A <relation-list> RELATION

<if-statement> ::= IF A <object-name> HAS A <relation-list> ,
THE CORRESPONDING <object-name>

{MUST / CAN / CANNOT} HAVE A <relation-list> RELATION

<relation-list> ::= <relation-name>/<relation-list> OR <relation-name>

Die Konsistenzregeln für den Systemaspekt

SYSTEM UMGEBUNG sehen in PSL folgendermaßen aus:

```
FOR EACH      OUTPUT          THERE MUST EXIST A RECEIVED BY              RELATION
FOR EACH      OUTPUT          THERE MUST EXIST A GENERATED BY             RELATION
FOR EACH      INPUT           THERE MUST EXIST A GENERATED BY             RELATION
FOR EACH      INPUT           THERE MUST EXIST A RECEIVED BY              RELATION
FOR EACH REAL-WORLD-ENTITY    THERE MUST EXIST A GENERATES OR RECIEVES
                                                  OR RESPONSIBLE-INT      RELATION
```

Auch weitere komplexere Restriktionen sind denkbar:

IF A PROCESS HAS A USES RELATION, THE CORRESPONDING SET MUST HAVE A
UPDATES RELATION

Ein Teil dieser Regeln wird z.B. durch den DATA PROCESS Report ausgewertet.

Reports zur Unterstützung der manuellen Prüfung

Die Speicherung der Systembeschreibung in einer Datenbank bietet die Möglichkeit, die vorhandenen Informationen nach mehreren Aspekten auszuwerten.

Formatierte Systembeschreibung

Alle in der Datenbank verfügbaren Informationen werden in PSL-Syntax ausgegeben, und zwar in gut lesbarer Formatierung und mit aller Redundanz (u.a. werden alle komplementären Relationen ergänzt) FORMATED PROBLEM STATEMENT .

Beschreibung von Teilmengen

Einzelne Systemaspekte oder auch Vereinigungen von Systemaspekten lassen sich als Reports erzeugen. Die isolierte Betrachtung von Teilaspekten verbessert die Obersicht.

Beispiele dafür sind die Teilmengen

- Objekte
- Daten
- Eingaben und Ausgaben
- bestimmte Relationen
- Teilsysteme, die aus bestimmten Prozessen bestehen

NAME-SELECTION, STRUCTURE, CONTENTS, SUBSET ANALYSIS

Graphische Ausgaben

Der Datenfluß durch das System kann graphisch ausgegeben werden. Das dynamische Verhalten kann in Form der sogenannten process chain, die durch Analyse der verschiedenen TRIGGER-Relationen zwischen Prozessen erzeugt wird graphisch gezeigt werden.

PICTURE, EXTENDED PICTURE, PROCESS CHAIN

Statistiken und Kataloge

Statistiken und Kataloge von Objekten können unter verschiedenen Gesichtspunkten erzeugt werden und verbessern damit die Obersichtlichkeit der Dokumentation.

Ein Beispiel dafür ist ein KWIC-Index über alle Objektnamen.

KWIC, NAME LIST, DATA BASE SUMMARY

Ausschnitte aus den für unser Entwurfsbeispiel erzeugten Reports werden nachfolgend wiedergegeben.

PSA Version A4.2R1 78.157 10.21

BELUST IDT/KFK PSA V4.2

Formatted Problem Statement

```
Parameters:   DB=PSADB.DBF  NAME=ANALYSENERGEBNISSE-BERECHNEN  NOINDEX  NOPUNCHED
  EMPTY  NOPUNCH  SMARG=5  NMARG=20  AMARG=10  BMARG=25  RNMARG=70  CMARG=1  HMAR
  NODESIGNATE  ONE-PER-LINE  DEFINE  COMMENT  NONEW-PAGE  NONEW-LINE  NOALL-STATE
  COMPLEMENTARY-STATEMENTS  LINE-NUMBERS  PRINTEOF  DLC-COMMENT

  1 PROCESS                                   ANALYSENERGEBNISSE-BERECHNEN;
  2      /*   DATE OF LAST CHANGE -        78.151, 09.58.33 */
  3     SYNONYMS ARE:  A1-341;
  4     PART OF:       ANALYSENERGEBNISSE-BESTIMMEN;
  5     USES:          ANALYSENAUFTRAEGE-2,
  6                    PROBENERGEBNISSE,
  7                    METHODENDATEN;
  8     DERIVES:       ANALYSENERGEBNISSE-341;
  9     TRIGGERED BY:  LETZTES-PROBENERGEBNIS;
 10     TERMINATION-CAUSES:
 11                    STANDARDANALYSE;
 12
 13 EOF EOF EOF EOF EOF
```

PSA Version A4.2R1 78.157 10.21

BELUST IDT/KFK PSA V4.2

Formatted Problem Statement

```
Parameters:   DB=PSADB.DBF  NAME=ARBEITSPLATZAUFTRAG-AUSWAEHLEN  NOINDEX  NOPUNCH
  EMPTY  NOPUNCH  SMARG=5  NMARG=20  AMARG=10  BMARG=25  RNMARG=70  CMARG=1  HMAR
  NODESIGNATE  ONE-PER-LINE  DEFINE  COMMENT  NONEW-PAGE  NONEW-LINE  NOALL-STATE
  COMPLEMENTARY-STATEMENTS  LINE-NUMBERS  PRINTEOF  DLC-COMMENT

  1 PROCESS                                   ARBEITSPLATZAUFTRAG-AUSWAEHLEN;
  2      /*   DATE OF LAST CHANGE -        78.151, 09.58.33 */
  3     SYNONYMS ARE:  A1-331;
  4     SEE-MEMO:      DIALOG-KOMMENTAR;
  5     PART OF:       PROBEN-ANALYSIEREN;
  6     UTILIZES:      DIALOG-FUEHREN;
  7     USES:          ANALYSENAUFTRAEGE-32,
  8                    PROBEN-32;
  9     DERIVES:       ARBEITSPLATZAUFTRAG-331;
 10     DERIVES:       PROBEN-331;
 11     PROCEDURE;
 12         1. PRIORITAET DES AUFTRAGS
 13         2. AUSLASTUNG DER ANALYSENGERAETE
 14         3. BESTIMMUNG DES ARBEITSPLATZES
 15         4. MITTEILUNG AN DAS PROBENVERTEILUNGSSYSTEM
 16            DURCH ARBEITSPLATZAUFTRAG;
 17
 18 EOF EOF EOF EOF EOF
```

PSA Version A4.2R1 78.151

BELUST IDT/KFK PSA V4.2

Formatted Problem Statement

```
Parameters:   DB=PSADB.DBP  NAME=MESSAUFTRAEGE  NOINDEX  NOPUNCHED-NAMES  PR
  SMARG=5  NMARG=20  AMARG=10  BMARG=25  RNMARG=70  CMARG=1  HMARG=40  NODES
  DEFINE  COMMENT  NONEW-PAGE  NONEW-LINE  NOALL-STATEMENTS  COMPLEMENTARY-S
  LINE-NUMBERS  PRINTEOP  DLC-COMMENT

  1 SET                                          MESSAUFTRAEGE;
  2      /*   DATE OF LAST CHANGE -       78.151, 09.58.33 */
  3     SUBSETS ARE:   ANALYSENAUFTRAEGE-1,
  4                    MESSWERTERFASSUNGSAUFTRAEGE;
  5     CONSISTS OF:
  6                         ANALYSENAUFTRAG,
  7                         MESSWERTERFASSUNGSAUFTRAG;
  8     SUBSETTING-CRITERIA ARE:
  9                    BESTIMMUNGSGROESSEN;
 10     USED BY:       MESSWERTE-ERFASSEN,
 11                    MESSUNGEN-AUSWAEHLEN,
 12                    MESSAUFTRAG-AUSWAEHLEN,
 13                    AUFTRAEGE-UEBERWACHEN;
 14     USED BY:
 15          IN-AUFTRAGSLISTE-EINREIHEN
 16           TO DERIVE     AUFTRAGSLISTEN-31;
 17     UPDATED BY:    PROZESS-UEBERWACHEN;
 18     DERIVED BY:    PROZESSZUSTAND-BEWERTEN;
 19     DERIVED BY:    MESSAUFTRAG-ERTEILEN;
 20     DERIVED BY:    MESSWERTERF-AUFTRAG-ERTEILEN;
 21     DERIVED BY:    ANALYSENAUFTRAG-BESTAETIGEN;
 22     DERIVATION;
 23         DIE PROZESSUEBERWACHUNG ERZEUGT MESSAUFTRAEGE DIE IN
 24         DIESEM SET GESAMMELT WERDEN UND ENTWEDER ALS ANALYSEN
 25         AUFTRAEGE ODER MESSWERTERFASSUNGSAUFTRAEGE WEITERVER
 26         ARBEITET WERDEN.;
 27     OCCURRENCES:   ANZAHL-WARTENDER-MESSAUFTRAEGE;
 28     VOLATILITY-SET;
 29         DIESER FILE VERAENDERT SICH IMMER WENN EIN
 30         NEUER MESSAUFTRAG ANGEFORDERT WIRD.
 31         DIES GESCHIEHT MEHRMALS AM TAG.;
 32
 33 EOF EOF, EOF EOF EOF
```

PSA Version A4.2R1 78.151

BELUST IDT/KFK PSA V4.2

Structure Report

Parameters: DB=PSADB.DBF NAME=WA INDENT=3 NOINDEX NOPUNCHED-NAMES LE
LINE-NUMBERS LEVEL-NUMBERS STATISTICS NONEW-PAGE PRINT

```
COUNT LEVEL NAME

    1     1 WIEDERAUFARBEITUNGSANLAGE                       PROCESS
    2      2   MESSWERTE-ERFASSEN
    3       3     MESSUNGEN-AUSWAEHLEN
    4        4       MESSAUFTRAEGE-SAMMELN
    5        4       MESSAUFTRAG-AUSWAEHLEN
    6        4       MESSVERFAHREN-AUSWAEHLEN
    7       3     PROBEN-ZIEHEN
    8        4       PROBENNAHMEAUFTRAEGE-ABLEITEN
    9         5         PROBENNAHMESTELLE-ERMITTELN
   10         5         ANALYSENMETHODE-VORGEBEN
   11         5         PROBENANZAHL-VORGEBEN
   12        4       PROBENNAHME-EINLEITEN
   13         5         PROBENNAHMEAUFTRAG-AUSWAEHLEN
   14         5         PROBENANZAHL-KORRIGIEREN
   15         5         PROBENNEHMERMAGAZINE-VERWALTEN
   16          6           MAGAZINBELEGUNG-PRUEFEN
   17          6           NACHFUELLAUFTRAG-AUSGEBEN
   18          6           MAGAZIN-NACHFUELLEN
   19          6           FLASCHENNUMMERN-ERFASSEN
   20          6           FLASCHENNUMMER-ZUORDNEN
   21         5         PROBENNAHMEVORGANG-AUSLOESEN
   22          6           BEFEHL-AUSGEBEN
   23          6           PROBENNAHMEPULT-BEDIENEN
   24          6           BEDIENUNG-BESTAETIGEN
   25          6           FLASCHE-FUELLEN
   26         5         HANDPROBEN-ZIEHEN
   27        4       PROBENKENNUNG-ERFASSEN
   28         5         PROBE-AUSWAEHLEN
   29         5         PROBEN-NR-ERFASSEN
   30         5         PROBEN-NR-VORGABEWERTVERGLEICH
   31       3     LABORANALYSEN-DURCHFUEHREN
   32        4       PROBEN-VORVERARBEITEN
   33         5         VORVERARBEITUNGSAUFTRAG-AUSG
   34         5         VORVERARBEITUNG-DURCHFUEHREN
   35         5         MESS-UND-KENNDATEN-ERFASSEN
```

PSA Version A4.2R1 78.151

BELUST IDT/KFK PSA V4.2

Contents Report

```
Parameters:   DB=PSADB.DBF  NAME=MAN-PROBENNAHMEAUFTRAG  NOCOMPLETENESS-CHE(
  NOPUNCHED-NAMES  LEVELS=ALL  LINE-NUMBERS  LEVEL-NUMBERS  NAME-TYPES  PRI

  1* (ENTITY)     1 MAN-PROBENNAHMEAUFTRAG
   1 (GROUP)       2   PROBENKENNUNG
   2 (GROUP)         3     PROBENNUMMER
   3 (ELEMENT)         4       PROBENNEHMER-NR
   4 (ELEMENT)         4       LAUFENDE-NUMMER
   5 (GROUP)         3     PROBENNAHMESTELLE
   6 (ELEMENT)         4       PROBENNEHMER
   7 (ELEMENT)         4       POSITION
   8 (ELEMENT)         4       BEHAELTERNUMMER
   9 (ELEMENT)       3     ANALYSENAUFTRAGSNUMMER
  10 (ELEMENT)       3     ALIQUOT
  11 (GROUP)         3     PARALLELPROBE
  12 (ELEMENT)       3     INTERNE-IDENTIFIZIERUNG
  13 (GROUP)       2   TECHNISCHE-DATEN
  14 (ELEMENT)       3     PROBENROHWERTE
  15 (ELEMENT)       3     PROBENERGEBNIS
  16 (ELEMENT)       3     VORGABEWERT
  17 (GROUP)       2   UEBERWACHUNGSINFORMATION
  18 (ELEMENT)       3     BEARBEITUNGSSTATUS
  19 (ELEMENT)       3     FERTIGSTELLUNGSTERMIN
  20 (ELEMENT)       3     ALTERNATIVMETHODE
```

PSA Version A4.2R1 78.151

BELUST IDT/KFK PSA V4.2

Frequency Report

Parameters: DB=PSADB.DBF FILE=PSATEMP.PSANAME INTERVAL NOINDEX NOPUNCH
ORDER=BYTYPE

INTERVAL: STATUS-PRUEFUNGS-ZYKLUS

APPLIES TO	TYPE	FREQUENCY
BEARBEITUNGSSTATUS-PRUEFEN	PROCESS	1

INTERVAL: TAG

APPLIES TO	TYPE	FREQUENCY
MAGAZIN-LEER	EVENT	MAGAZIN-LEER-HAEUFIGKEIT
PROBEN-ZIEHEN	PROCESS	200
PROTOKOLL-AUSGEBEN	PROCESS	MEHRERE

BELUST IDT/KFK PSA V4.2

Contents Analysis Report

Parameters: DB=PSADB.DBF FILE=PSATEMP.PSANAME PRINT-MATRIX CONSISTS EXPLANATION

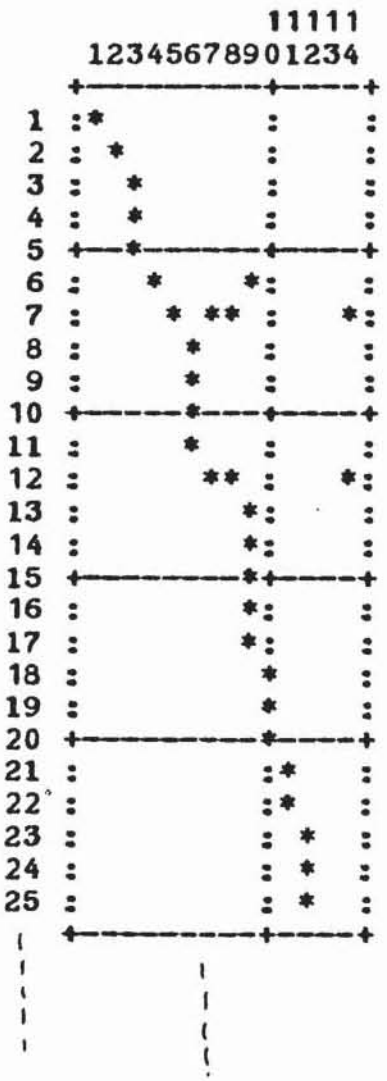
PSA315:NAME DOES NOT CONSIST OF ANYTHING - ABFAELLE-1-2
PSA315:NAME DOES NOT CONSIST OF ANYTHING - ANALYSENERGEBNISSE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - ANALYSENERGEBNISSE-343
PSA315:NAME DOES NOT CONSIST OF ANYTHING - AUFTRAGSLISTEN-31
PSA315:NAME DOES NOT CONSIST OF ANYTHING - AUFTRAGSLISTEN-32
PSA315:NAME DOES NOT CONSIST OF ANYTHING - AUSWERTEALGORITHMUS
PSA315:NAME DOES NOT CONSIST OF ANYTHING - BEDIENUNGSANWEISUNG
PSA315:NAME DOES NOT CONSIST OF ANYTHING - BEHAELTER-EICHWERTE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - BEHAELTERWERTE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - BESTIMMUNGSGROESSEN
PSA315:NAME DOES NOT CONSIST OF ANYTHING - BEWERTETER-PROZESSZUSTAND
PSA315:NAME DOES NOT CONSIST OF ANYTHING - BEWERTUNGSKRITERIEN
PSA315:NAME DOES NOT CONSIST OF ANYTHING - DATENAUSWAHLKRITERIEN
PSA315:NAME DOES NOT CONSIST OF ANYTHING - DIREKT-ERFASSTE-MESSWERTE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - ENDPRODUKTE-2
PSA315:NAME DOES NOT CONSIST OF ANYTHING - MESSWERTE-332
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PARALLELPROBE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROBENERGEBNISSE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROTOKOLLDATEN
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROTOKOLLE
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROTOKOLLRAHMEN
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROZESSDATEN-2
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROZESSDATEN-4
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROZESSDATEN-41
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PROZESSDATEN-42
PSA315:NAME DOES NOT CONSIST OF ANYTHING - PRUEF-EICHDATEN
PSA315:NAME DOES NOT CONSIST OF ANYTHING - ROHWERTE-SCHEMA
PSA315:NAME DOES NOT CONSIST OF ANYTHING - UEBERWACHUNGSDATEN

	Row Names		Column Names	
1	PROZESSDATEN-2	GROUP	1 AKTUELLER-PROZESSZUSTAND	GROUP
2	STANDARDABWEICHUNG	ELEMENT	2 ANALYSENERGEBNISSE-341	GROUP
3	UMRECHNUNGSFAKTOREN	ELEMENT	3 AUSWERTEDATEN	GROUP
4	ROHWERTE-SCHEMA	GROUP	4 FLASCHENNUMMERN	GROUP
5	AUSWERTEALGORITHMUS	GROUP	5 LEERE-PROBENFLASCHEN	GROUP
6	PROBENNUMMER	GROUP	6 METHODENKENNUNG	GROUP

BELUST IDT/KFK PSA V4.2

Contents Analysis Report

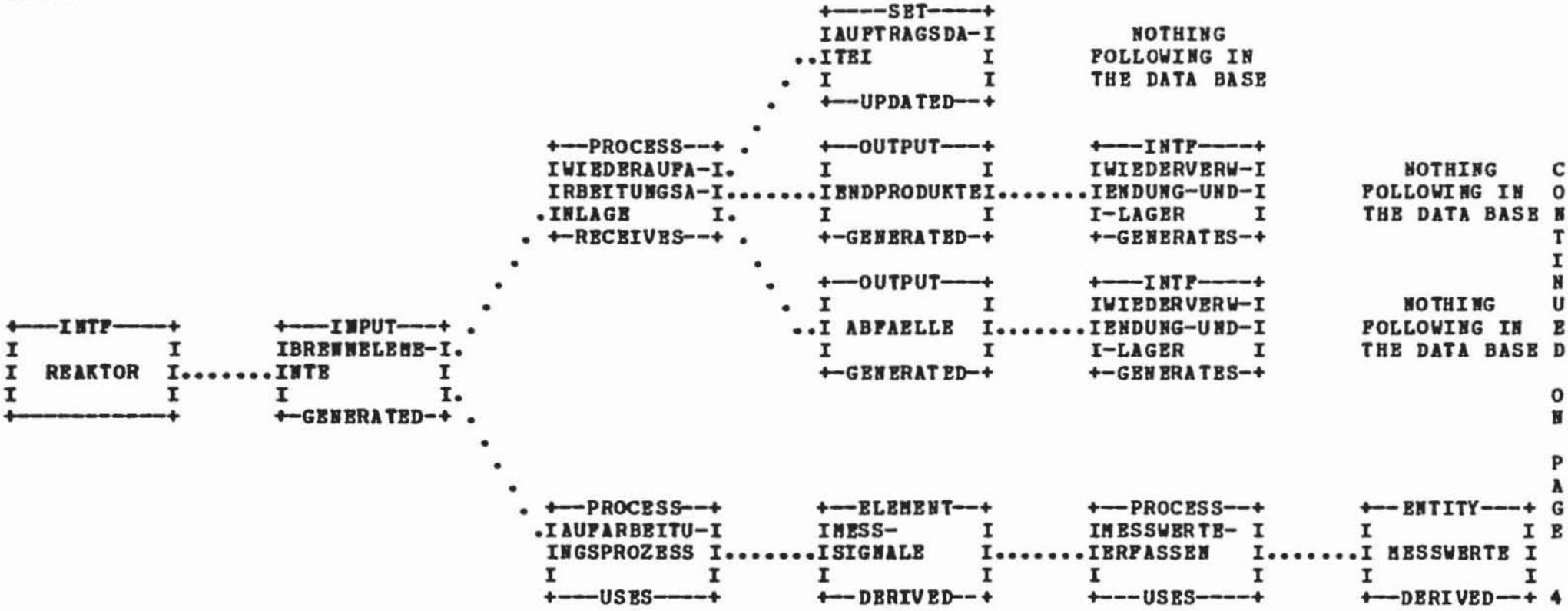
THE ROWS ARE CONTAINED IN THE COLUMNS WITH *S



BELUST IDT/KFK PSA V4.2

EXTENDED PICTURE

INITIAL NAME = REAKTOR



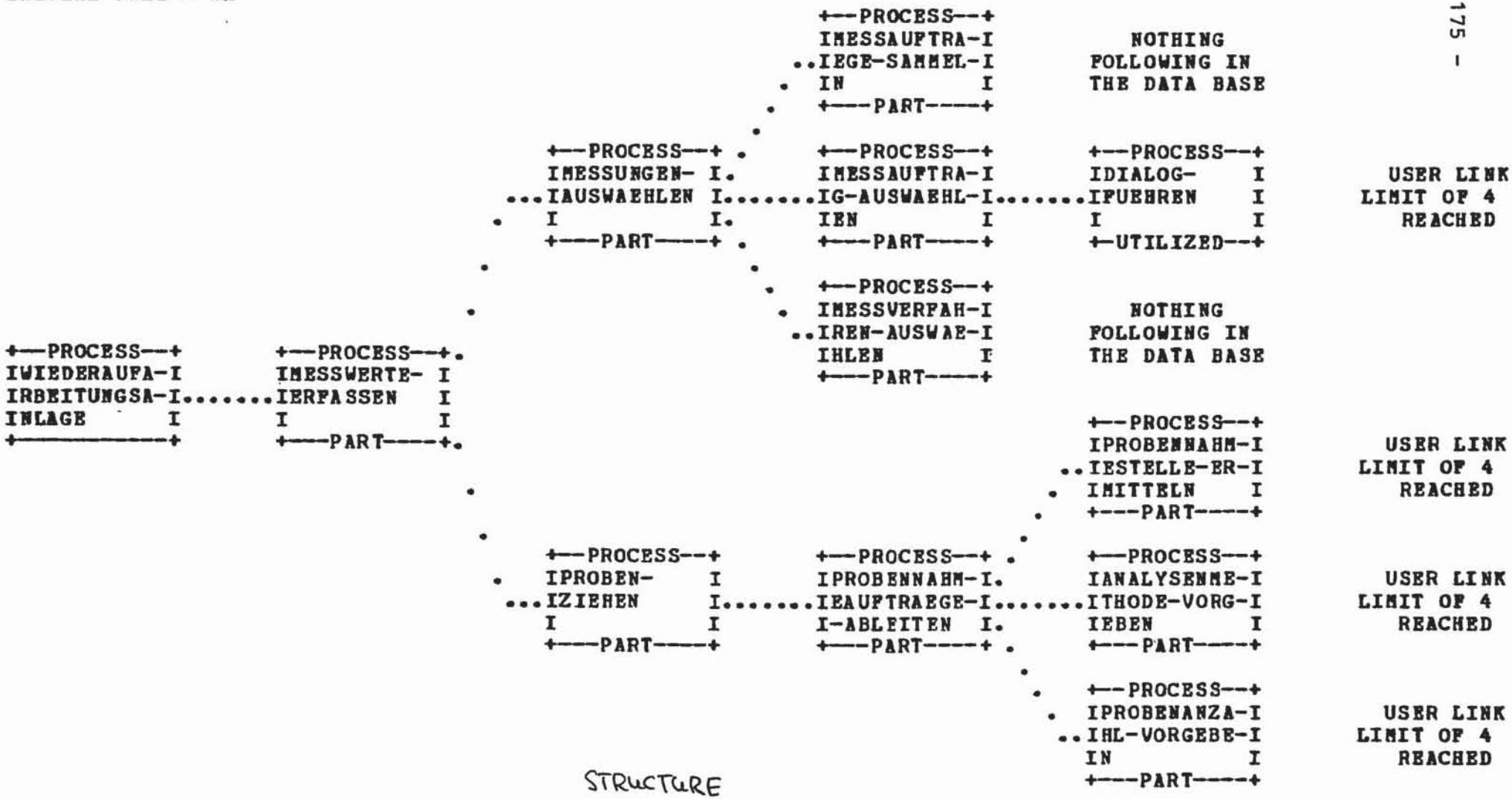
DATA FLOW

CONTINUED ON PAGE 6

BELUST IDT/KFK PSA V4.2

EXTENDED PICTURE

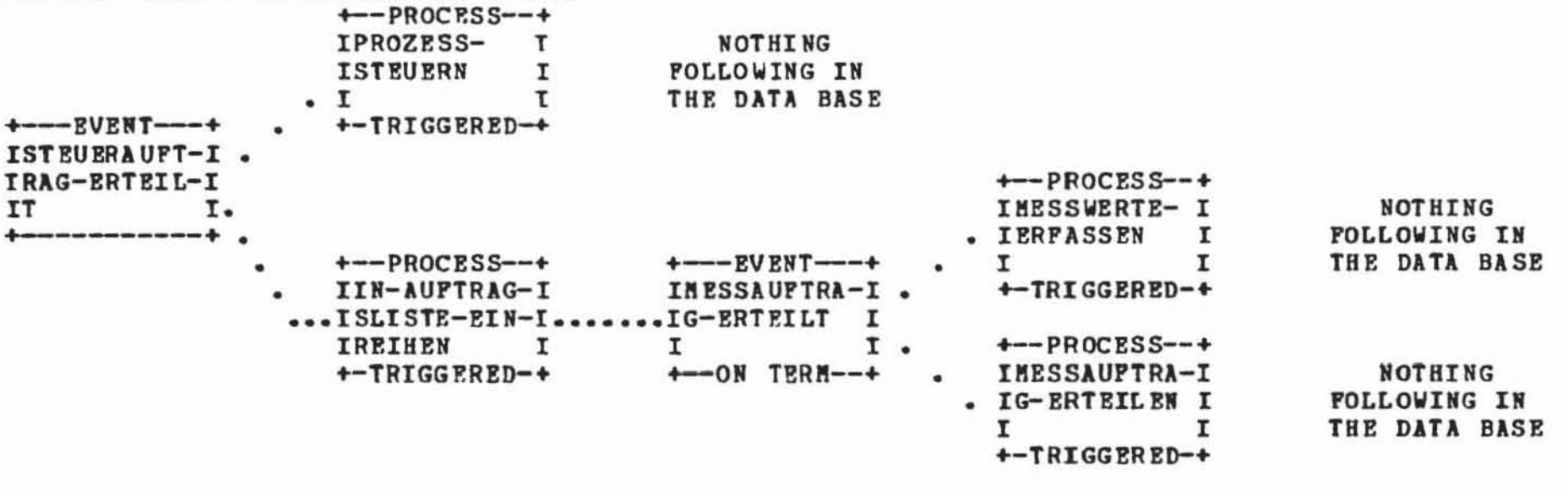
INITIAL NAME = WA



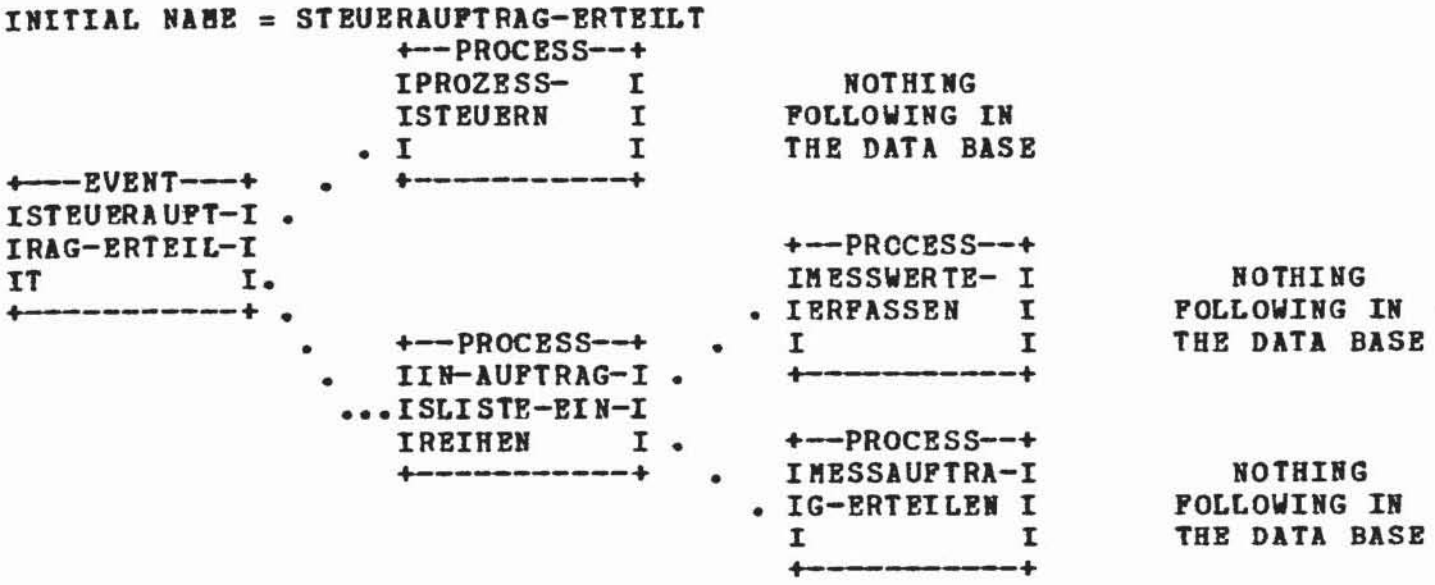
STRUCTURE

CONTINUED ON PAGE 23

Process Chain



Process Chain



Erweiterungen von PSL/PSA für Prozeßrechneranwendungen

Viele Prozeßrechneranwendungen sind Systeme, deren interne Abläufe durch Ereignisse in der Umgebung (techn. Prozeß) gesteuert werden und deren Aktionen (Verarbeitungsschritte) innerhalb einer begrenzten Zeit eine Antwort des Systems gewährleisten müssen. Die Einhaltung dieser Zeitbedingungen und die Eigenschaften der Prozeßführungsaufgaben (Ereignisse im technischen Prozeß sind meist zeitlich nicht einplanbar, asynchron) machen es notwendig, daß diese Systeme Prozesse enthalten, die (quasi-)simultan ablaufen können. Im Vergleich zu anderen Softwaresystemen spielt die Kommunikation mit der Umgebung und somit auch deren Beschreibung eine besonders wichtige Rolle.

Aus dem oben Gesagten leiten wir für unsere Erweiterungen zwei Voraussetzungen ab:

- In den ersten Phasen der Softwareentwicklung (Spezifikation, Grobentwurf) sollte bei der Beschreibung keine Unterscheidung zwischen Software, Hardware und anderen Objekten der Umgebung (Mensch als Operator), die mit dem System eine Schnittstelle haben, vorgenommen werden. Dies bedeutet, daß keine gesonderten Sprachelemente zur Verfügung gestellt werden. Bei der weiteren Verfeinerung des Entwurfs können dann Entwurfsentscheidungen wie Hardware/Software durch geeignete Attributzuordnungen festgelegt werden.

- Das System kann als ein Netzwerk kooperierender paralleler Prozesse beschrieben werden, die über Datenbereiche miteinander kommunizieren. Diese Konzeption maximal möglicher Parallelität (aus logischer Sicht) bedeutet nicht, daß das System so implementiert werden muß. Aus Effizienzgründen (Entwurfsentscheidung) können später bestimmte parallele Ablauffolgen sequentialisiert und zu Tasks zusammengefaßt werden.

Prozesse laufen parallel ab, wenn sie sich zeitlich überlappen, d.h. wenn sich in einem endlichen Zeitintervall mehr als ein Prozeß zwischen seinem Anfangs- und Endzeitpunkt befindet. Dabei können keine Voraussagen über den relativen Fortschritt der parallel ablaufenden Prozesse im Vergleich zueinander gemacht werden. Diese Unbestimmtheit bzgl. des relativen Fortschritts führt zu den nicht trivialen Problemen der parallelen Verarbeitung,

da die auftretenden Fehler in einem System zeitabhängig werden (unerwünschtes Oberschreiben gemeinsamer Daten oder gegenseitige Blockierung mehrerer Prozesse). Da es aber in komplexen Realzeitsystemen praktisch unmöglich ist, alle zeitabhängigen Ereignisse exakt zu wiederholen, sind diese Fehler im allgemeinen nicht reproduzierbar. Hier zeigen sich die Grenzen des Testens solcher Systeme; damit erhebt sich die Forderung, daß durch konstruktive Maßnahmen bereits in der Entwurfsphase die unbeabsichtigte gegenseitige Beeinflussung paralleler Prozesse ausgeschlossen werde. Um diese Forderung zu erfüllen, muß das SBM von PSL ein Modell für den dynamischen Ablauf enthalten. Restriktionen des parallelen Prozeßablaufs sollten durch geeignete höhere Synchronisationsmechanismen ausdrückbar sein. Das Modell für die Dynamik des Systems sollte einerseits für statische Fehleranalysen wie Deadlockerkennungs-Algorithmen, andererseits auch als Basis für eine Simulation verwendbar sein.

Wir werden uns im folgenden auf die Erweiterung des Systemaspekts DYNAMISCHES VERHALTEN von PSL und der damit verbundenen Erweiterung der Analysefähigkeiten von PSA beschränken.

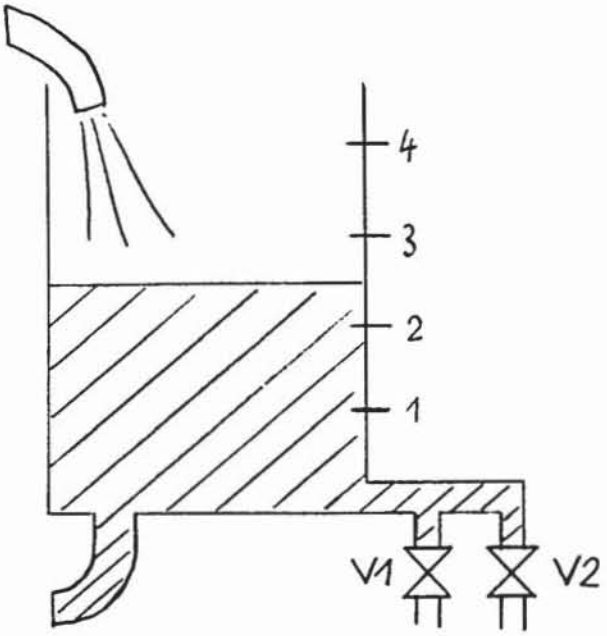
Geplante Erweiterung des Systemaspekts DYNAMISCHES VERHALTEN

. Endliche Automaten

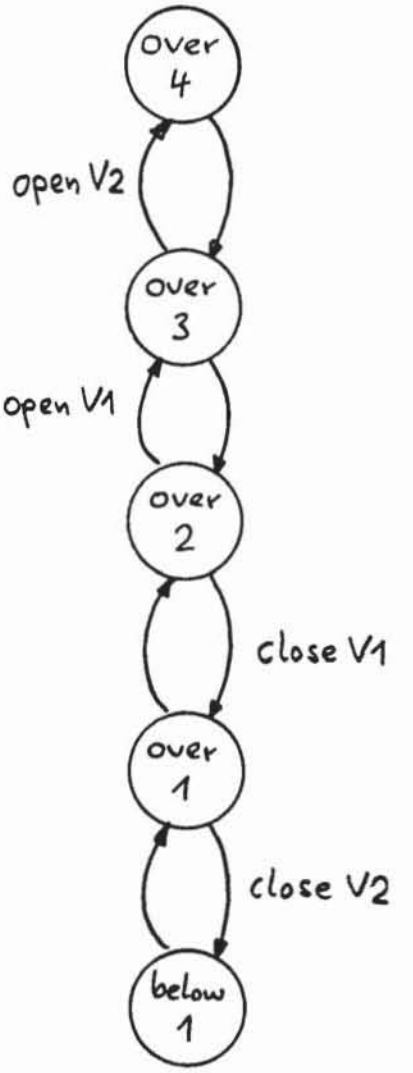
Bisher hat man in PSL die Möglichkeit, den sequentiellen Ablauf von Aktionen durch die TRIGGER-Relation zu beschreiben, was der Beschreibung durch Flußdiagramme entspricht. Es gibt eine Klasse von Prozeßrechneranwendungen, die sog. Folgeprozesse, die im wesentlichen durch ihren sequentiellen Ablauf beschrieben werden können. Folgeprozesse bewegen sich in einem endlichen Zustandsraum, ihre Ablauffolgen lassen sich deshalb durch einen endlichen Automaten (EA) modellieren. EA haben gegenüber Flußdiagrammen den Vorteil, daß der Ablauf nicht durch eine Folge von Aktionen dargestellt wird, sondern daß der Effekt dieser Aktionen in Form von Zustandstransformationen beschrieben werden kann.

Zur Definition eines EA benötigen wir als Sprachmittel die Objekttypen PROCESS, STATE und EVENT. Ereignisse (EVENT) werden als Zustandsübergänge definiert.

Beispiel: In einem Wasserbehälter werden fünf Füllstände unterschieden: unter Pegel 1, über Pegel 1 bis über Pegel 4 . Zwei Ventile V1 und V2 sollen geschlossen oder geöffnet werden, um einen konstanten Wasserpegel zu halten.
V1 (V2) soll geöffnet werden, wenn der Pegel über Pegel 3 (Pegel 4) steigt, geschlossen, wenn er unter Pegel 2 (1) fällt.



WATERCONTAINER



TRANSITION-DIAGRAM

```
PROCESS watercontainer;
    STATES  below-1,over-1,over-2,over-3,over-4;
    TRANSITIONS  below-1 TO over-1,
                 over-1  TO over-2,
                 over-2  TO over-3  NAMED up-1,
                 over-3  TO over-4  NAMED up-2,
                 over-4  TO over-3,
                 over-3  TO over-2,
                 over-2  TO over-1  NAMED down-1,
                 over-1  TO below-1 NAMED down-2;

EVENTS up-1,down-1,up-2,down-2;
    up-1   STARTS  open-V1;
    down-1 STARTS  close-V1;
    up-2   STARTS  open-V2;
    down-2 STARTS  close-V2;

PROCESS open-V1;
    PRODUCES open-message FOR valve-control;
```

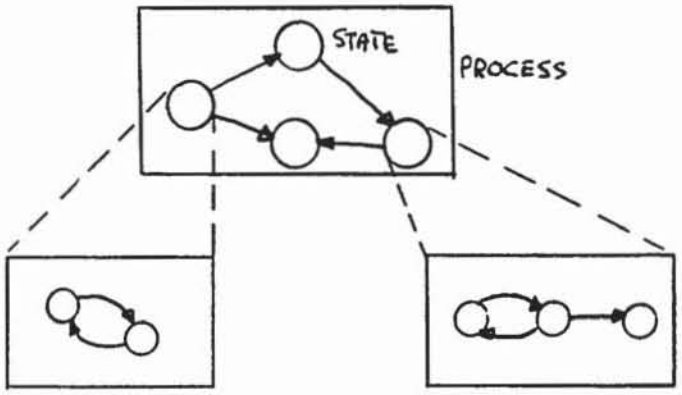
Für komplexere Systeme ist es notwendig, hierarchische Automaten zu verwenden, d.h. ein Zustand kann wiederum einen Automaten enthalten. Wir erreichen dies dadurch, daß einem Zustand ein PROCESS zugeordnet oder umgekehrt ein PROCESS nur dann aktiviert werden kann, wenn ein anderer sich in einem bestimmten Zustand befindet.

```
STATE A                                PROCESS B
   ASSOCIATED [PROCESS] B                  WHILE PROCESS C IN [STATE] A
```

Die Aktivierung kann auch periodisch durch eine Uhr (TIMER) gesteuert werden:

```
TIMER   Wassertemperaturzyklus ;
   WHILE  Heizanlage = arbeitet
   CYCLE  60 SECONDS
   STARTS Wassertemperatur-Regelung
```

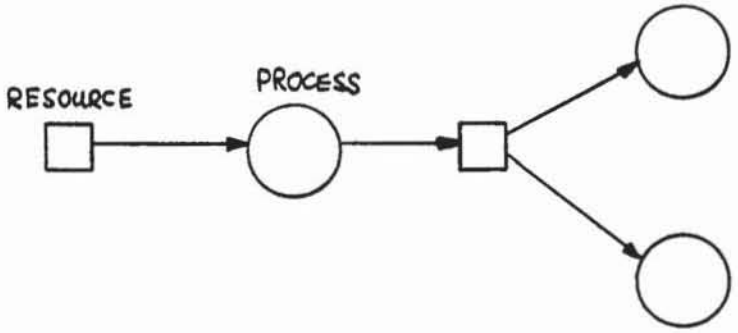
Da ein PROCESS wiederum durch einen Automaten definiert werden kann, erhält man so einen hierarchischen Automaten.



Flußgraphen

Zur Beschreibung paralleler Abläufe führen wir Flußgraphen ein. Sie können als eine Vereinigung von erweiterten Petri-Netzen und Datenflußgraphen angesehen werden.

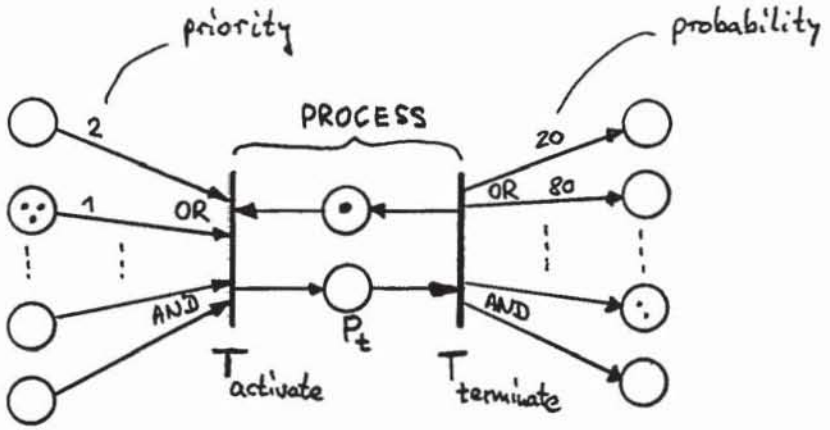
Ein Flußgraph ist ein gerichteter Graph mit zwei Arten von Knoten: Prozessen, dargestellt als Kreise, und Resourcen, dargestellt als Kästen. Die einzelnen Prozesse konkurrieren um Resourcen und beeinflussen sich so gegenseitig. Im Prinzip kann ein Prozeß aktiviert werden, sobald seine benötigten Resourcen verfügbar sind.



Abhängig von der Art der benötigten Resourcen können bestimmte Beziehungsrelationen (Kantenmarkierungen) zwischen Prozessen und Resourcen definiert werden, die Bedingungen für die Aktivierung der Prozesse beschreiben und somit die möglichen Ablauffolgen festlegen.

Jedem Knoten vom Typ PROCESS wird in Anlehnung an erweiterte Petri-Netze eine Eingabe- und Ausgabelogik zugeordnet. Diese Logik kann entweder eine UND- oder ODER-Verknüpfung sein. Jeder eingehenden und ausgehenden Kante kann eine Gewichtung zugeordnet werden. Bei eingehenden Kanten gibt diese die Anzahl der für eine Aktivierung notwendigen Resourceelemente an. Bei ausgehenden Kanten gibt diese die Anzahl der nach einer Aktivierung produzierten Resourceelemente an. Die Resourceelemente können von einem bestimmten Typ sein.

In erweiterter Petri-Netz Notation kann der Kontrollaspekt eines PROCESS durch zwei Transitionen dargestellt werden, eine zur Aktivierung (Belegung) und die andere zur Termination (Freigabe) des PROCESS .



Definition der Ein- und Ausgangslogik durch erweiterte Petri-Netze

Zur Auflösung von Konflikten bei OR-Logik in der Eingabe können den Kanten Prioritäten zugeordnet werden, bei OR-Logik in der Ausgabe können den Kanten Wahrscheinlichkeiten beigegeben werden. Für Simulationszwecke kann einem PROCESS eine Zeitdauer zugeordnet werden, die im Simulationsmodell als Verweildauer der Marke P_t dargestellt wird.

Resourcen können Datenbereiche, Geräte etc. sein. Wir unterteilen die Resourcen in zwei Teilklassen und führen die beiden Objekttypen DATA und BUFFER ein. Als Merkmal definieren wir, ob eine Resource permanent wie z.B.eine Tabelle oder verbrauchbar wie z.B. eine Botschaft oder das von einer Maschine verarbeitete Material ist.

Objekttyp: DATA permanente Resource
BUFFER verbrauchbare Resource

Zur Beschreibung der erlaubten Prozeßinteraktion führen wir nun Relationen als höhere Synchronisationsmechanismen ein. Diese Synchronisationsmechanismen korrespondieren mit der Lösung der drei klassischen Synchronisationsprobleme: Erzeuger-Verbraucher, Leser-Schreiber und wechselseitiger Ausschluß.

Der Inhalt eines Objekts vom Typ BUFFER kann nur erzeugt (PRODUCES) und verbraucht (CONSUMES) werden, ein DATA dagegen kann gelesen (READS) und überschrieben (WRITES) sowie belegt (OCCUPIES) werden.

Für die Aktivierung von PROCESSes, die durch diese Relationen verknüpft sind, gelten folgende Synchronisationsregeln:

produces/consumes: Ein PROCESS, der einen vollen BUFFER weiter füllen oder aus einem leeren weiter verbrauchen will, muß warten, wird also suspendiert.

Bei der Beschreibung des Puffers kann aber auch ein anderes Verhalten festgelegt werden, z.B. daß aller Input in den vollen Puffer verlorengeht.

reads/writes : Mehrere PROCESSes können parallel von einem gemeinsamen DATA lesen.
Beschreibt ein PROCESS ein gemeinsames DATA, so werden alle anderen PROCESSES suspendiert.

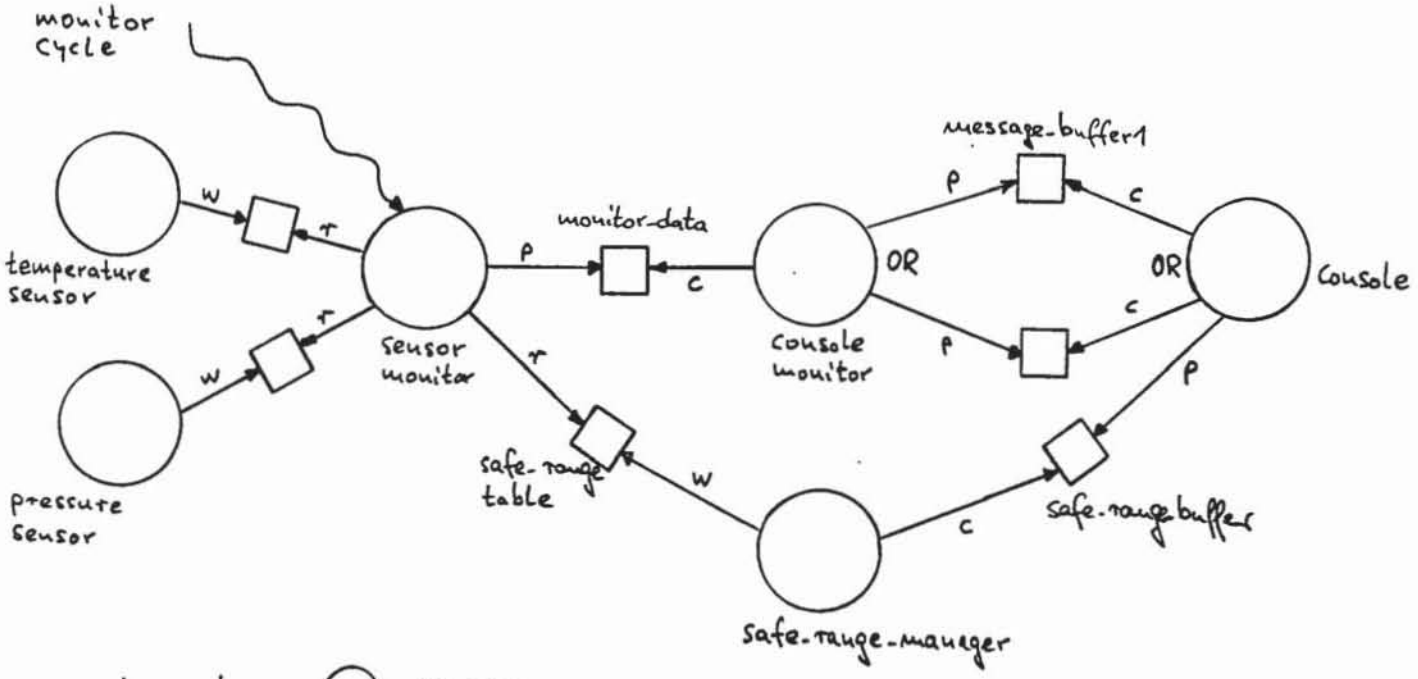
occupies : Ist ein gemeinsamer DATA von einem PROCESS belegt, so hat kein anderer PROCESS Zugriff und wird suspendiert.

Neben der Kommunikation von Prozessen durch gemeinsame Resourcen ist auch eine direkte Koordination von Prozessen ohne Datenaustausch möglich. Dies entspricht der Koordination durch Marken in einem Petri-Netz und kann als Trigger oder Interrupt interpretiert werden. Zu diesem Zweck haben wir zwei Ereignistypen EVENT und TIMER eingeführt, die einen PROCESS aktivieren können.

```
EVENT enter
  STARTS command-interpreter
```

```
TIMER  start-temp-control
  CYCLE   10 SECONDS
  STARTS  temperature-control
```

BEISPIEL: Teil eines Patientenüberwachungssystems
(Die nachfolgende Beschreibung in erweitertem PSL ist nicht vollständig)



Legende :
- ○ PROCESS
- □ BUFFER, DATA
- r, w reads, writes
- p, c produces, consumes

```
PROCESS sensor-monitoring-system;
    DESCRIPTION;
        As part of a patient-monitoring-system sensor-stations
        are monitored periodically which measure the temperature
        and pressure of a patient. These factors are read in and
        checked with specified safe-ranges. If a factor falls
        outside of the sensor's safe-range a warning-message
        should be produced. Otherwise these factors are output
        to a console. The console is able to set up and change
        the safe-ranges.

    SUBPARTS ARE  pressure-sensor,temperature-sensor,
                  sensor-monitor,console-monitor,
                  console,safe-range-manager;

PROCESS pressure-sensor;
    CONSUMES pressure;
    WRITES   pressure-signal;

PROCESS temperature-sensor;
    CONSUMES temperature;
    WRITES   temperature-signal;

PROCESS sensor-monitor;
    READS pressure-signal,temperature-signal,
          safe-range-table;
    PRODUCES factor-info FOR monitor-data;

BUFFER monitor-data;
    SIZE 5 OF TYPE factor-info;
    ORDER FIFO;

TYPE factor-info;
    STRUCTURE temperature-value,
              pressure-value,
              safe-ranges;

TYPE temperature-value;
    VALUES ARE 30 TO 45;

PROCESS console-monitor;
    CONSUMES 1 OF TYPE factor-info FROM monitor-data;
    PRODUCES warning FOR message-buffer1 WITH PROBABILITY 10
          OR factor-info FOR message-buffer2 WITH PROB 90;

PROCESS console;
    CONSUMES 1 OF TYPE warning FROM message-buffer1
          OR 5 OF TYPE factor-info FROM message-buffer2
    OR PRODUCES safe-ranges FOR safe-range-buffer;

PROCESS safe-range-manager;
    CONSUMES safe-ranges FROM safe-range-buffer;
    WRITES   safe-range-table;

TIMER monitor-cycle;
    CYCLE 60 SECONDS;
    STARTS sensor-monitor;

DATA safe-range-table;
    CONSISTS OF no-of-patients OF TYPE safe-ranges;

TYPE safe-ranges;
    STRUCTURE temperature-up,temperature-down,
              pressure-up, pressure-down;
```

Erweiterungen der dynamischen Analysefähigkeiten

Da sich unsere Untersuchungen zu diesem Themenkreis erst in einem Anfangsstadium befinden, sind die hier skizzierten Oberlegungen nur als potentielle Lösungsmöglichkeiten zu verstehen. Dies gilt insbesondere für die Ausführungen im Zusammenhang mit Petri-Netzen.

Graphische Ausgaben

Für die neu eingeführten Objekte und Relationen lassen sich graphische Ausgaben wie z.B. das Transitionsdiagramm eines Automaten und die zugeordnete Verbindungsmatrix ausgeben.

Konsistenzprüfungen:

Nach Festlegung geeigneter Konsistenzregeln lassen sich statische Prüfungen durchführen:

. Jeder Automat, dem ein PROCESS zugeordnet ist, muß genau einen Anfangs- und mindestens einen Endzustand besitzen.

. Jeder PROCESS muß eine Aktivierungsbedingung enthalten, entweder explizit durch einen EVENT oder TIMER, oder implizit durch mindestens eine CONSUMES- oder WHILE-Relation.

Statisch/analytische Prüfungen

Die Theorie der Petri-Netze stellt Analysealgorithmen zur Verfügung, mit denen man viele Eigenschaften von Petri-Netzen aufgrund der statischen Netzstruktur berechnen kann.

Es ist bekannt, daß endliche Automaten durch eine Teilklasse der Petri-Netze repräsentiert werden können. Abstrahiert man in den von uns gewählten Flußgraphen von den speziellen Resourcen und repräsentiert diese allein durch Plätze mit Marken, so entsteht eine erweiterte Form von Petri-Netzen, deren Mächtigkeit nicht größer ist als die von normalen Petri-Netzen. Sie lassen

sich also in normale Petri-Netze abbilden und sind so für die Analysealgorithmen zugänglich. Dies gilt allerdings nur unter der Voraussetzung, daß das erweiterte Petri-Netz K-beschränkt ist, d.h. daß die BUFFER-Längen (CAPACITY) beschränkt sind. Durch die Analysealgorithmen lassen sich Eigenschaften wie Erreichbarkeit (ist ein bestimmter Systemzustand (Markenverteilung) von einem gegebenen Zustand her erreichbar), Sicherheit (wird die Kapazität eines BUFFER nie überschritten), Lebendigkeit (Systemverklemmung) prüfen.

Simulation

Für große Systeme lassen sich bestimmte Eigenschaften selbst dann, wenn man ein formales Ablaufmodell wie Petri-Netze zugrundelegt, nur schwer analytisch ermitteln.

Im IDT existiert ein Simulationssystem, das auf erweiterten Petri-Netzen (S-Netzen) basiert. Voraussetzung für die Benutzung dieses Systems ist, daß alle zur Steuerung dieses Simulationsmodells notwendigen Informationen in der Datenbank vorhanden sind.

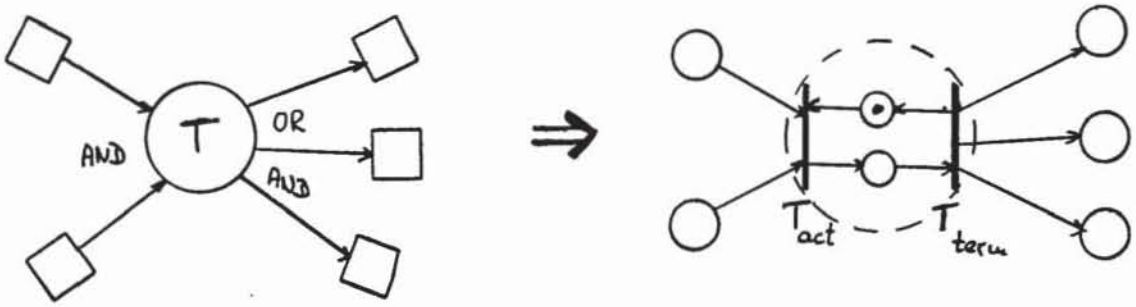
Wesentliche Erweiterungen der S-Netze gegenüber Petri-Netzen /5/:

- Jeder Platz hat eine sogenannte Platzzeit $T_p \geq 0$, welche die Verweildauer einer Marke auf diesem Platz definiert (die Platzzeit kann auch eine Zufallsvariable sein).
- Marken können Attribute zugeordnet werden.
- Jeder Transition ist ein sogenanntes Aktivierungs- und Feuerungsschema zugeordnet.
- Besitzt eine Transition mehrere Ausgangsplätze, so kann durch ein Zufallsgesteuertes Auswahlverfahren (Placeselectionmode) der dynamisch nachfolgende Platz ausgewählt werden (ODER-Logik am Ausgang).
- Besitzt der Eingangsplatz einer Transition T mehr Marken, als zur Feuerung von T notwendig sind, so kann eine Ordnung für die Reihenfolge der Auswahl definiert werden (Tokenselection-mode).
- Den Transitionen können Prioritäten zugeordnet werden.

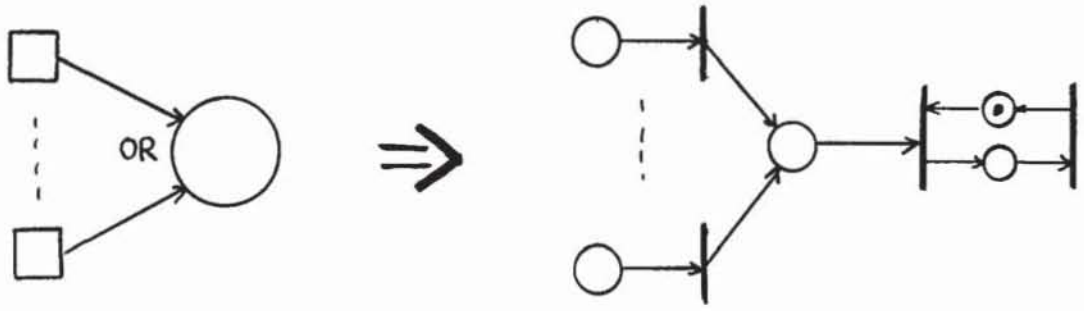
Wenn die Platzzeit irgendeiner Marke abgelaufen ist, werden die Aktivierungsschemata der Nachfolgetransitionen aufgerufen. Das Aktivierungsschema einer Transition gibt für jeden Eingabeplatz die Zahl der zur Feuerung notwendigen Marken an. Das Aktivierungsschema ruft das Feuerungsschema auf, falls die Markenanzahl jedes Eingabeplatzes größer oder gleich der im Aktivierungsschema definierten Markenanzahl ist.

Zur Abbildung der Flußgraphen auf S-Netze müssen folgende Schritte durchgeführt werden:

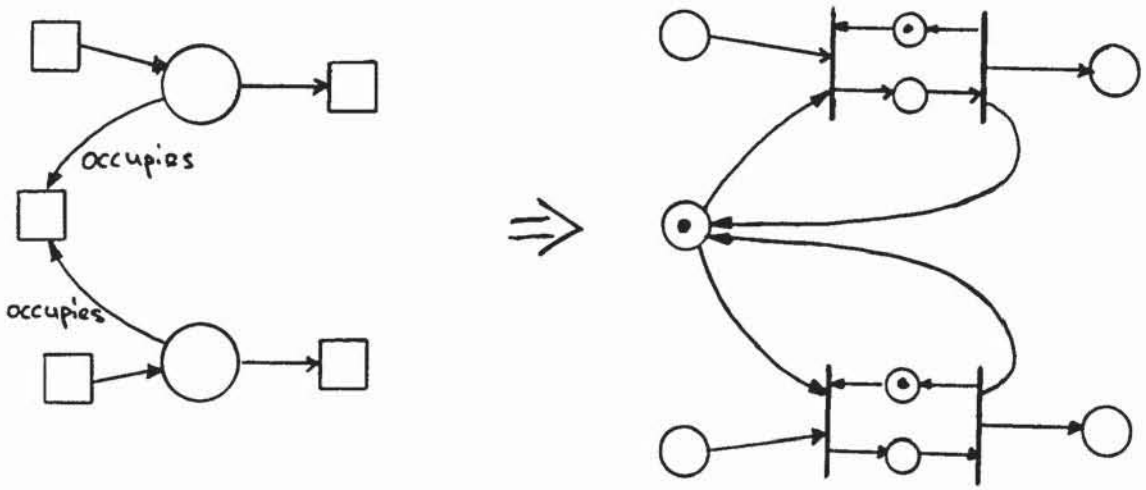
1. Alle Resourcen werden zu Plätzen, alle Prozesse werden wie folgt transformiert:



2. Die ODER-Eingangslogiken werden transformiert:



3. Die Synchronisationsbedingungen der Relationen werden transformiert z.B. für die OCCUPIES-Relation wie folgt:



Die Informationen zur Steuerung des Simulationssystems können dann aus der Datenbank entnommen werden.

Beispiel: Folgende Informationen werden für eine Transition angefordert:

```
TRANSITIONSNO:      1
     PRIORITY:   1.000
     ACTIVATION-SCHEME:
          INPUTPLACE  :                    1        2
          NO-OF-TOKENS-NEEDED              1        3
     FIRING-SCHEME :
          INPUTPLACE:                      1        2
          NO-OF-TOKENS-TO-BE-DISPLACED     1        3
          NO-OF-RESULTING-TOKENS           1        2
          TOKENSELECTION-MODE              FIFO     FIFO
          PLACESELECTION-MODE              RANDOM
          POSSIBLE OUTPUT-PLACES           3        4
```

Sie können aus der Flußgraph-Beschreibung abgeleitet werden:

```
PROCESS T1
     PRIORITY    1
     CONSUMES    1  OF TYPE  input 1  FROM  P1
             AND 3  OF TYPE  input 2  FROM  P2
     PRODUCES    1  OF TYPE  message  FOR   P3  WITH PROBABILITY  20
             OR  2  OF TYPE  output   FOR   P4  WITH PROBABILITY  80
     DURATION    7  DELTA  3

BUFFER  P1, P2
     ORDER FIFO
```

Durch Zuordnung einer Transitionsprozedur kann die Wahrscheinlichkeitsverteilung 20/80 des PLACESELECTION-MODE spezifiziert werden.

Das Simulationssystem berechnet bestimmte Meßgrößen, die bei geeigneter Interpretation Auskunft über das dynamische Verhalten des Systems geben. Für jeden Platz können z.B. Minimum, Maximum, Durchschnitt und Varianz der Anzahl der Marken und der Zeit ermittelt werden. Die durchschnittliche Markenplatzzeit kann z.B. als durchschnittliche Wartezeit eines Auftrags oder als mittlere Ausführungsdauer eines Prozesses interpretiert werden. Die durchschnittliche Anzahl von Marken auf einem Platz kann als Warteschlangenlänge oder als durchschnittliche Auslastung aufgefaßt werden. Ferner enthält das System Tracemöglichkeiten und erzeugt Histogramme.

Literatur

/1/ ISDOS-Project
Problem Statement Language (PSL)
Language Reference Manual
ISDOS Ref. # 7242-0142-0 , 1977

/2/ ISDOS-Project
Problem Statement Analyzer (PSA)
Reports and Report Commands
ISDOS Ref. # 7742-0144-1 , 1977

/3/ Ludewig, J., Streng, W.
Oberblick und Vergleich verschiedener Mittel für
Spezifikation und den Entwurf von Software
KfK-Bericht 2506 (1978)

/4/ Ludewig, J., Streng, W.
Prüfmittel für die rechnergestützte Software-Entwicklung
KfK-Bericht 2507 (1978)

/5/ Schumacher, F.
Beschreibung und Auswertung diskreter dynamischer Systeme
KfK-Bericht 2635 (1978)