Alfred Holl

Structured design of process models, structured business process modeling

1.1 Motivation 1

BPM is a type of process (dynamic function) modeling, a subtype of behavior modeling, represented by
– event-driven process chain [A. W. Scheer, ARIS]
– UML activity diagram
– BPMN business process modeling notation

What other modeling approaches belong to this type?
Control flow modeling in program design and programming represented by
– block diagram (flow chart)
– Nassi-Shneiderman diagram
– UML activity diagram

Comparison of current diagrams:

– BPM	**unstructured: spaghetti [Scheer 1994]**
– control flow diagram	**structured**

Control flow modeling styles		BPM styles	
1950s 1960s	Spaghetti code programming and spaghetti design	late 1980s	Spaghetti BPM
early 1970s	Structured programming and structured design	2010 ?	Desire: Structured BPM (not only in WFM)

Historic comparison (Holl / Valentin 2004)

Why did BPM not realize the similarity and learn from structured program design?
– BPM ← business, information systems
– structured program design ← computer science

1.1 Motivation 2

The problem of structuring is independent of the notation used.

“There is nothing to prevent the systems analyst from creating an arbitrarily complex, unstructured flowchart.” [Yourdon 1989,222]

Not only
– mapping of spaghetti reality
but even
– higher complexity than the complexity of the reality

“Unless great care is taken, the flowchart can become incredibly complicated and difficult to read.” [Yourdon 1989, 290]

Only Nassi-Shneiderman is restrictive
with regard to structuring,
but it is not applied to BPM

“The Nassi-Shneiderman diagrams are generally more organized, more structured and more comprehensible than the typical flowchart.” [Yourdon 1989, 224]

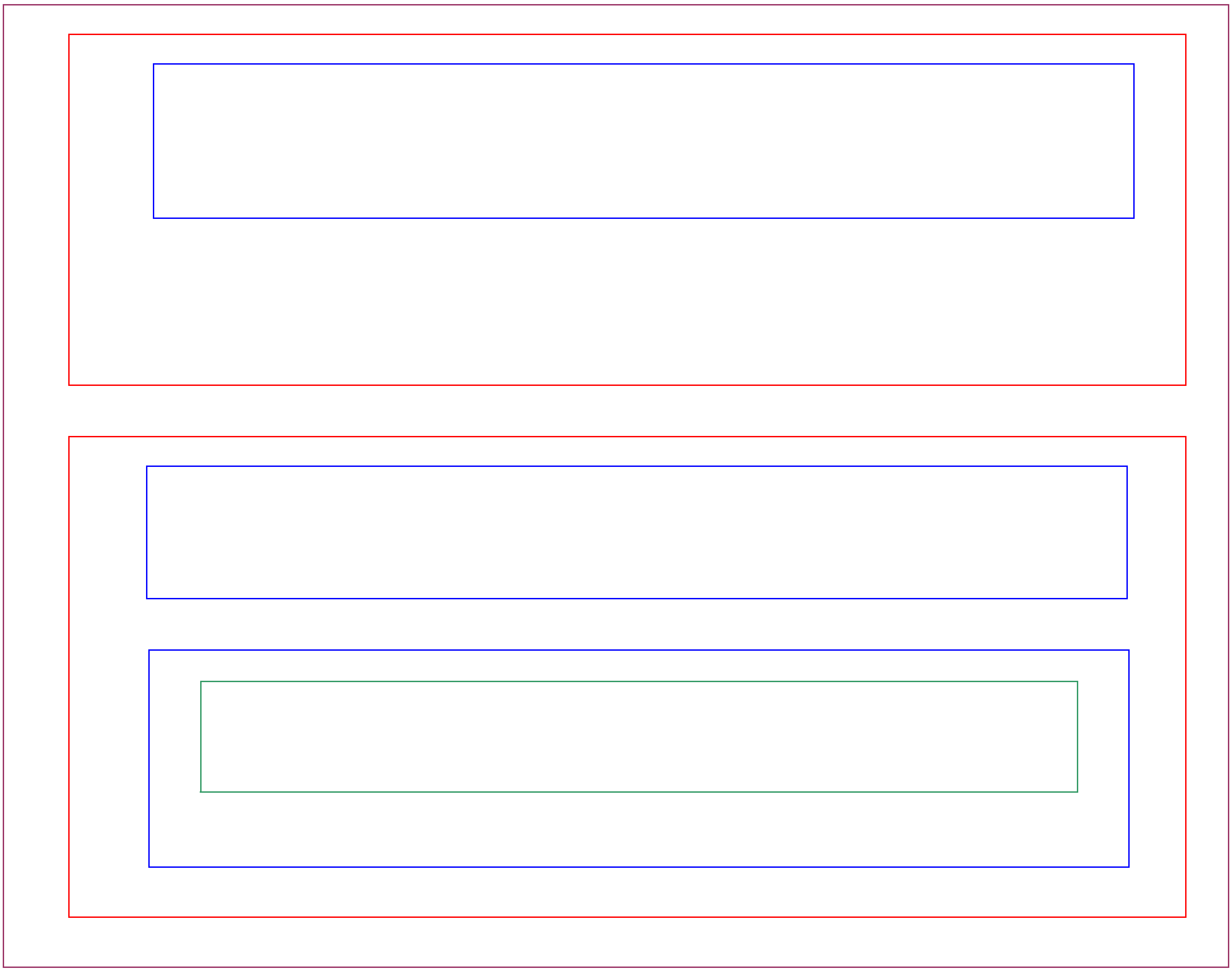
Improvement

“To create a structured flowchart, the systems analyst must organize his or her logic with nested combinations of the flowchart symbols (by Böhm-Jacopini).” [Yourdon 1989, 222]

Böhm-Jacopini proof 1966 shows the sufficiency of sequence, selection (alternative / test) and repetition (iteration) for every mathematically describable process.

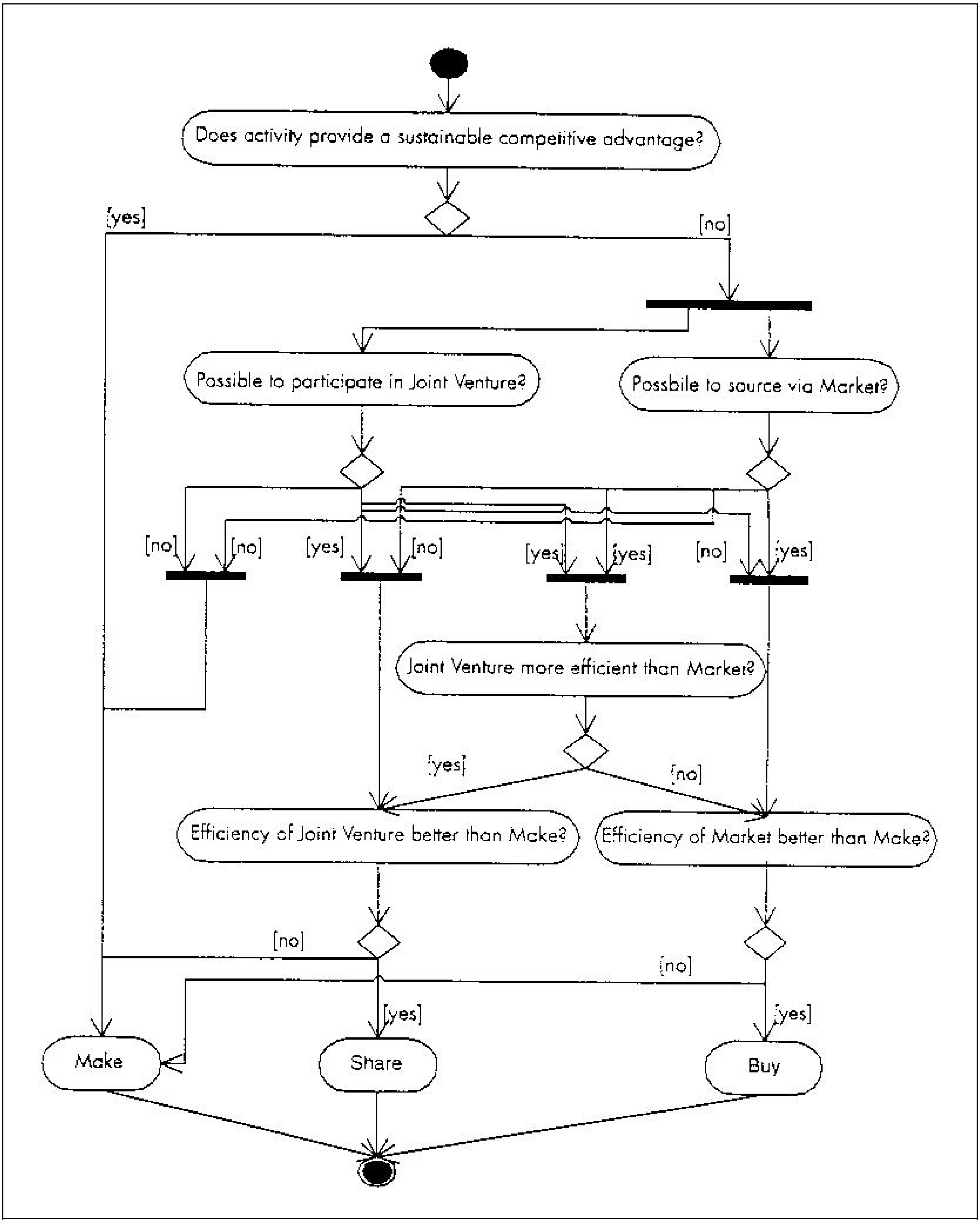
1.1 Motivation 3

Nested structure components



cf. latest version of UML sequence diagrams

1.2.1 Unstructured examples: current literature 1



(Wirtschaftsinformatik 46(2004) 207)

1.2.1 Unstructured examples: current literature 2

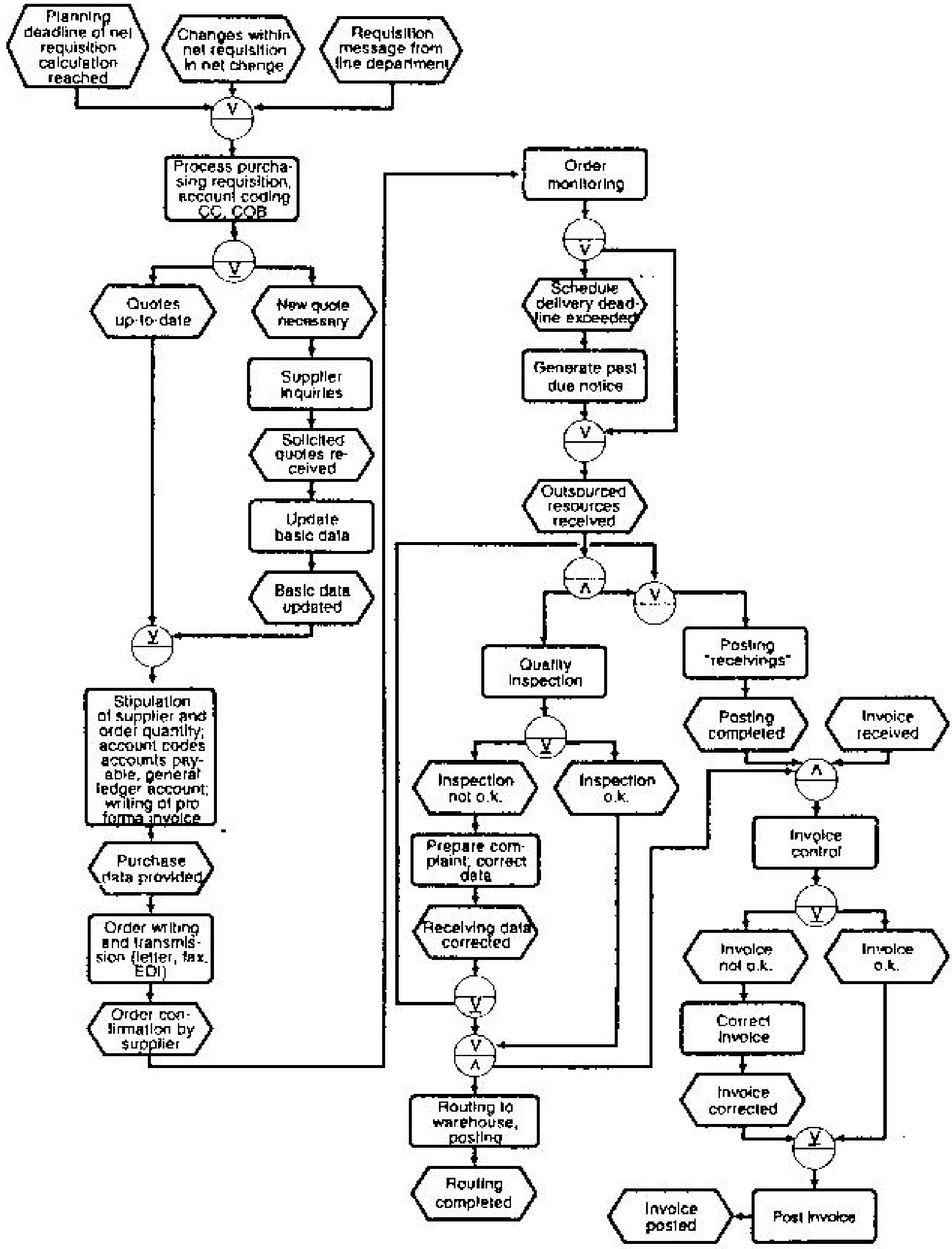


Fig. B.II.09: Event-driven process chain for inbound logistics

(Scheer, Business Process Engineering, 1994: 404)

1.2.1 Unstructured examples: current literature 3

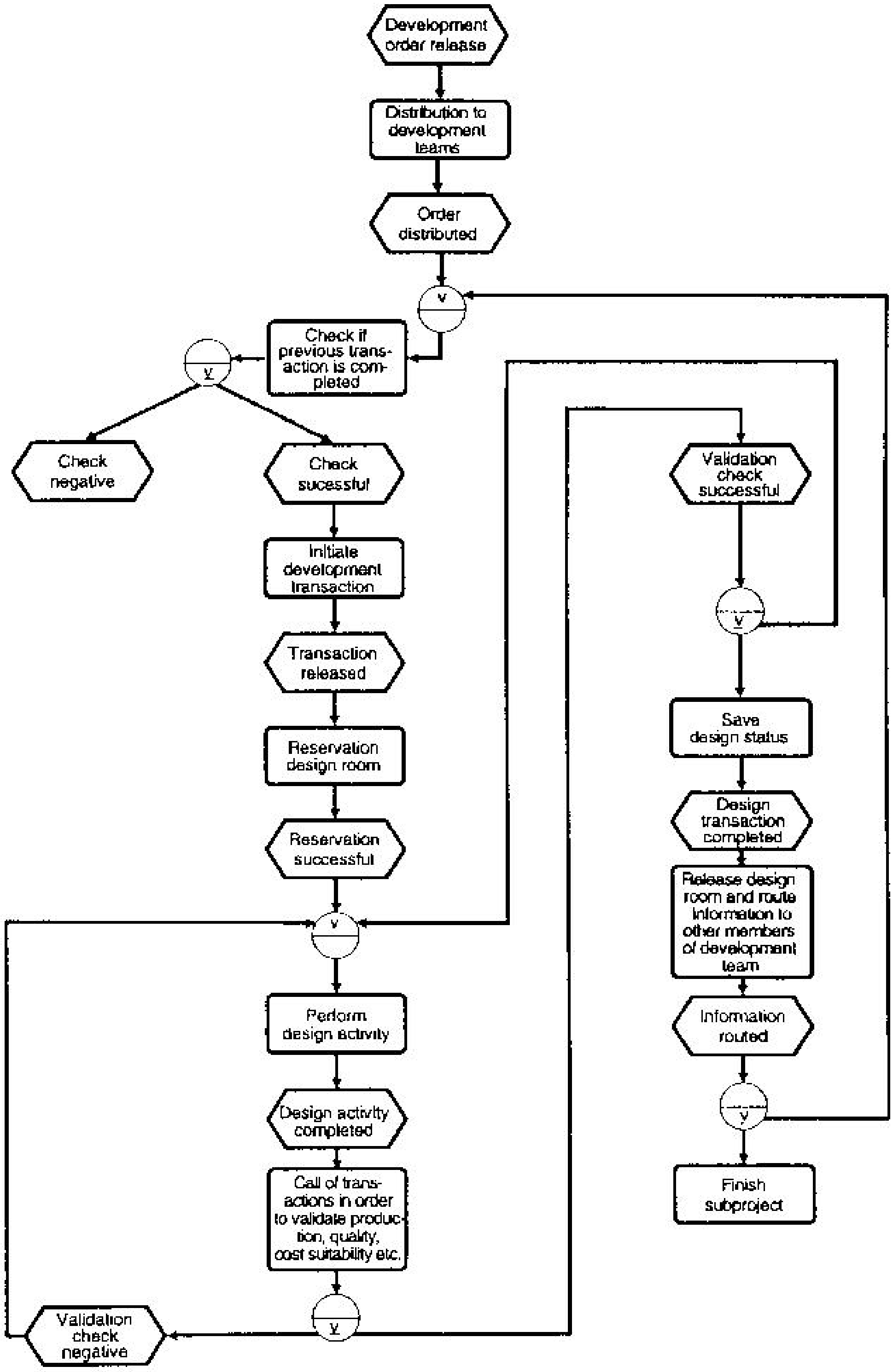


Fig. C.II.56: EPC for the design process

(Scheer, Business Process Engineering, 1994: 589)

1.2.1 Unstructured examples: current literature 4

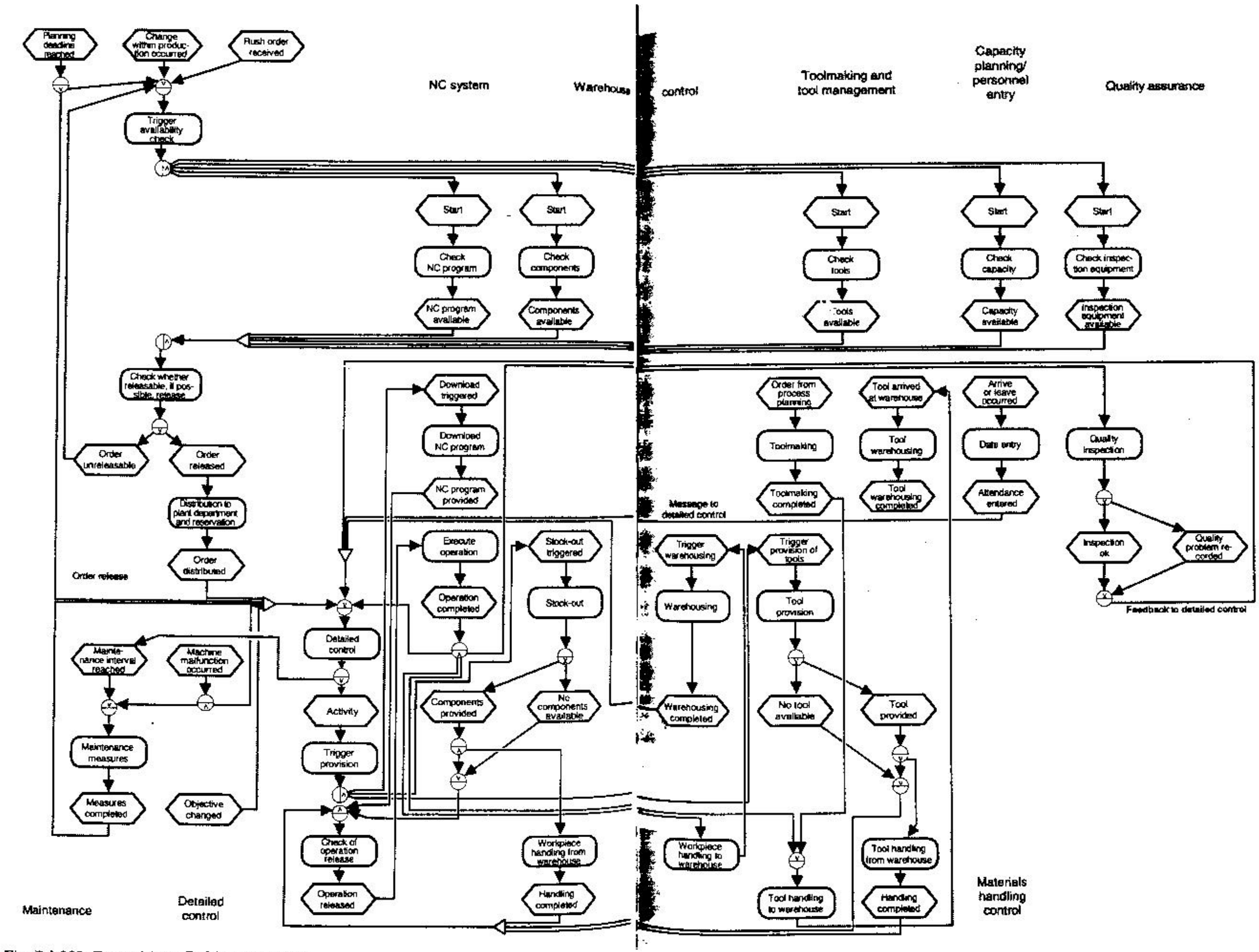
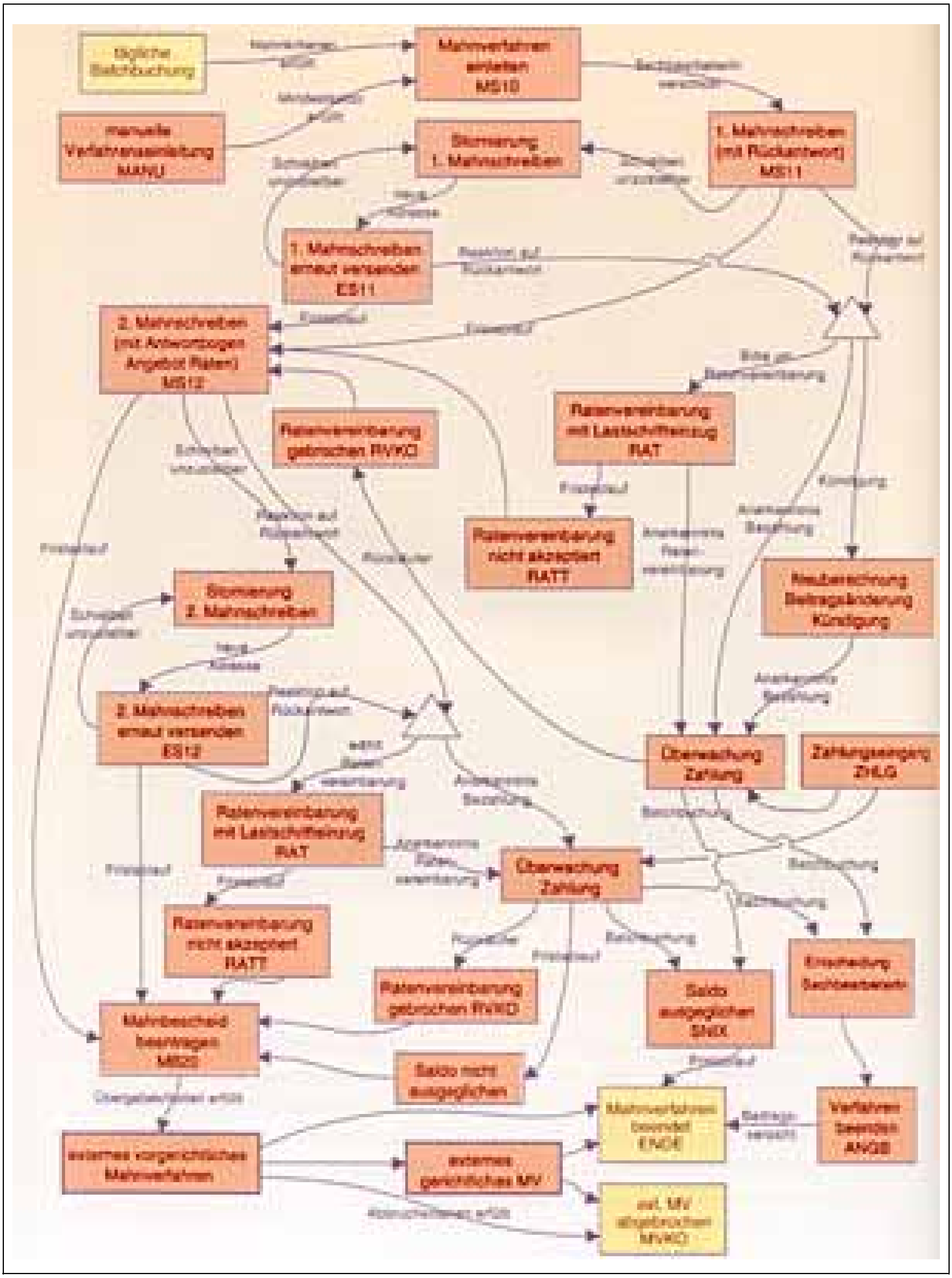


Fig. B.1.222: Event driven CAM process chain

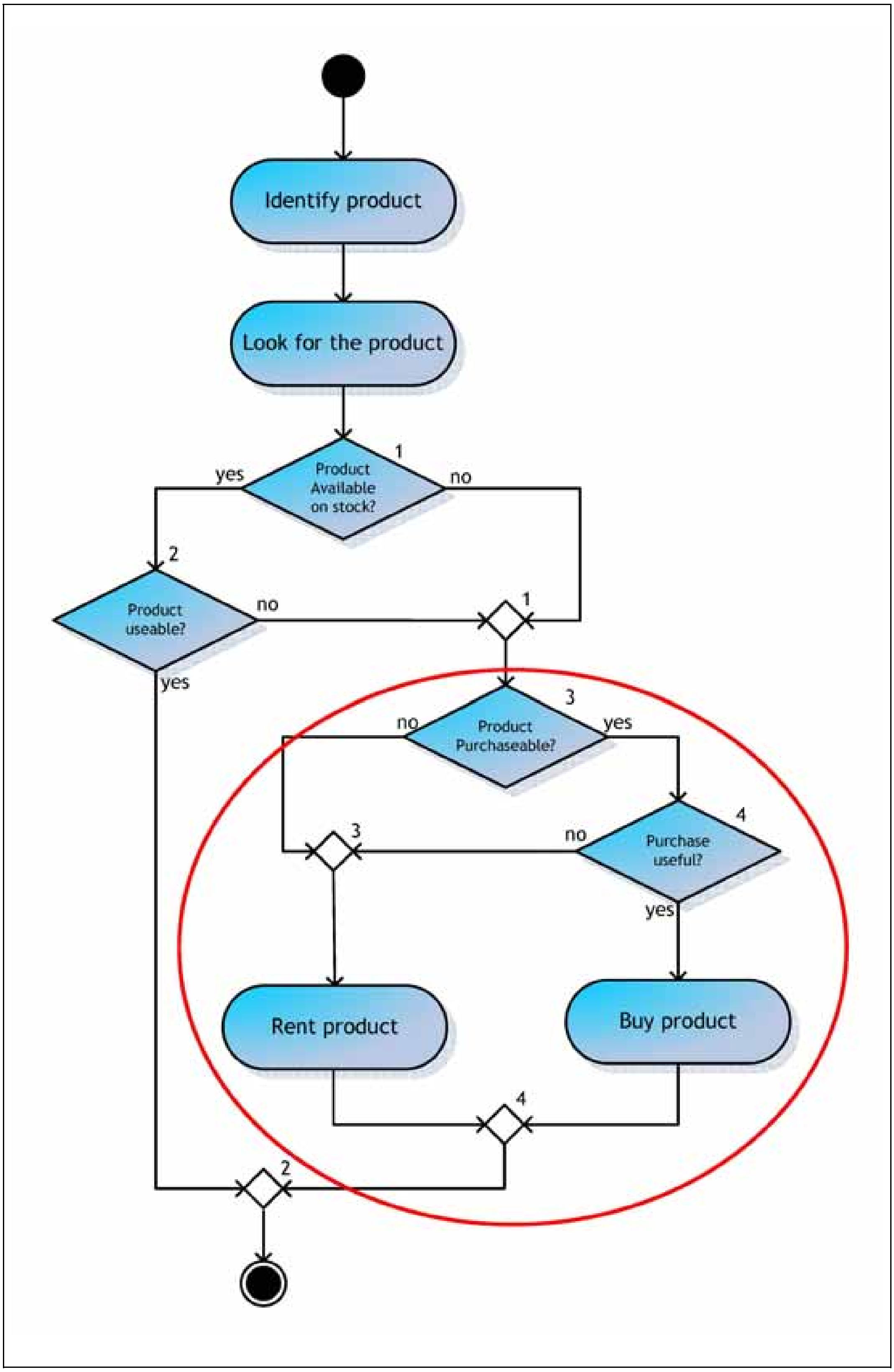
(Scheer, Business Process Engineering, 1994: 350-351)

1.2.1 Unstructured examples: current literature 5



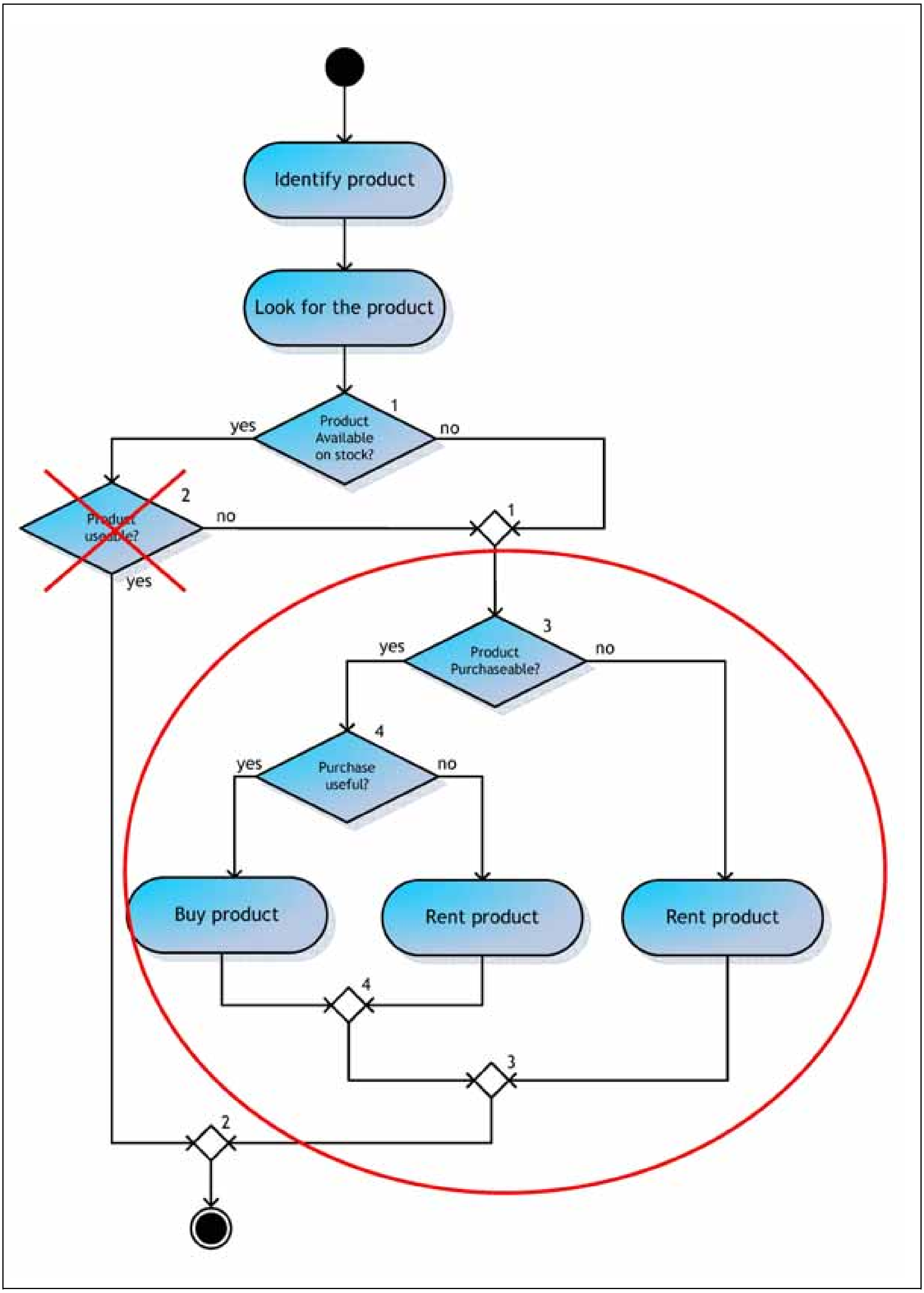
(Süddeutsche Zeitung 14.04.2008)

1.2.2 Unstructured examples: structuring 1



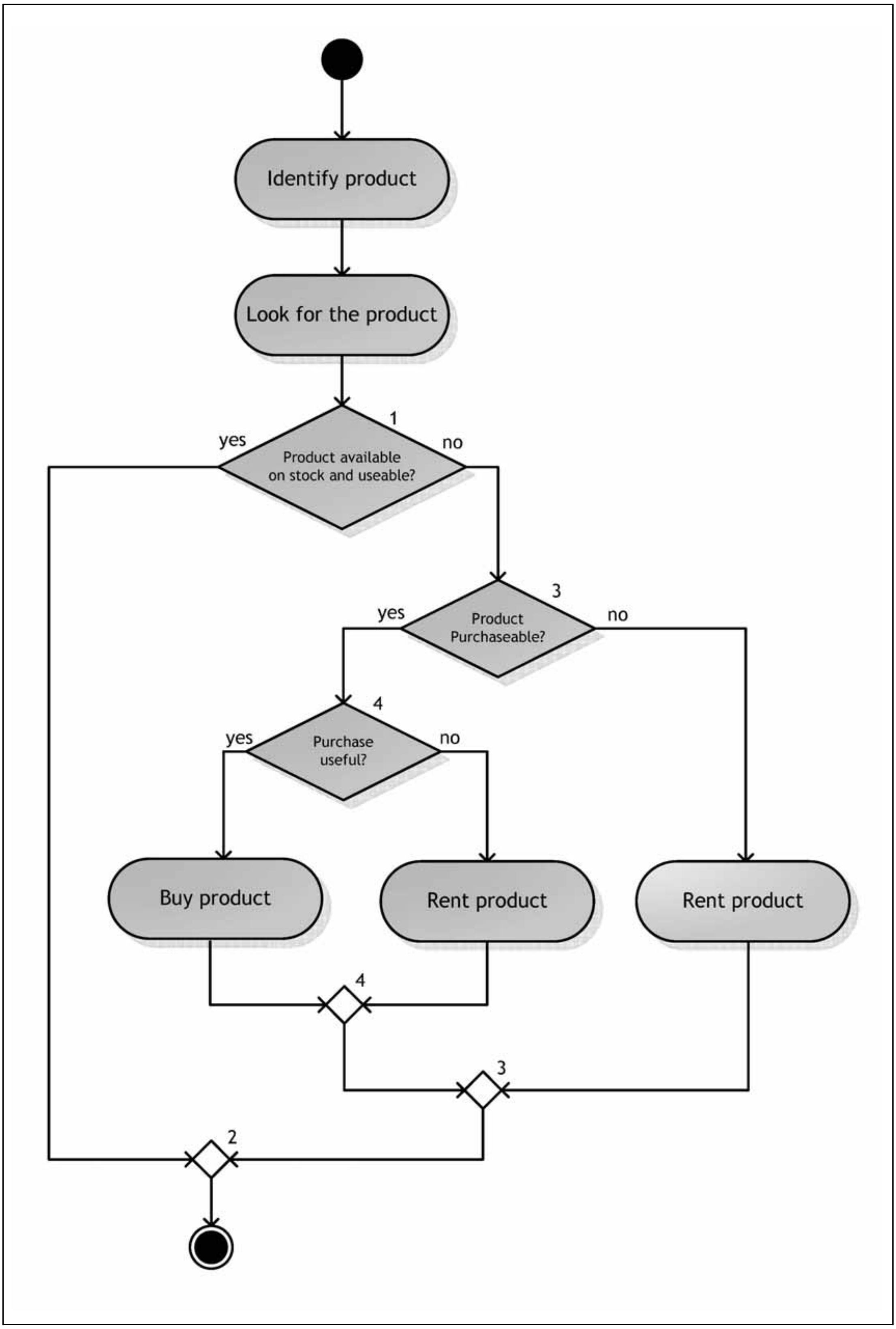
Typical example of the current BPM style
in the form of a UML activity diagram;
example only covers unstructured tests (Holl / Valentin 2004)

1.2.2 Unstructured examples: structuring 2



Improved business process model
(Holl / Valentin 2004)

1.2.2 Unstructured examples: structuring 3



Well-structured business process model
(Holl / Valentin 2004)

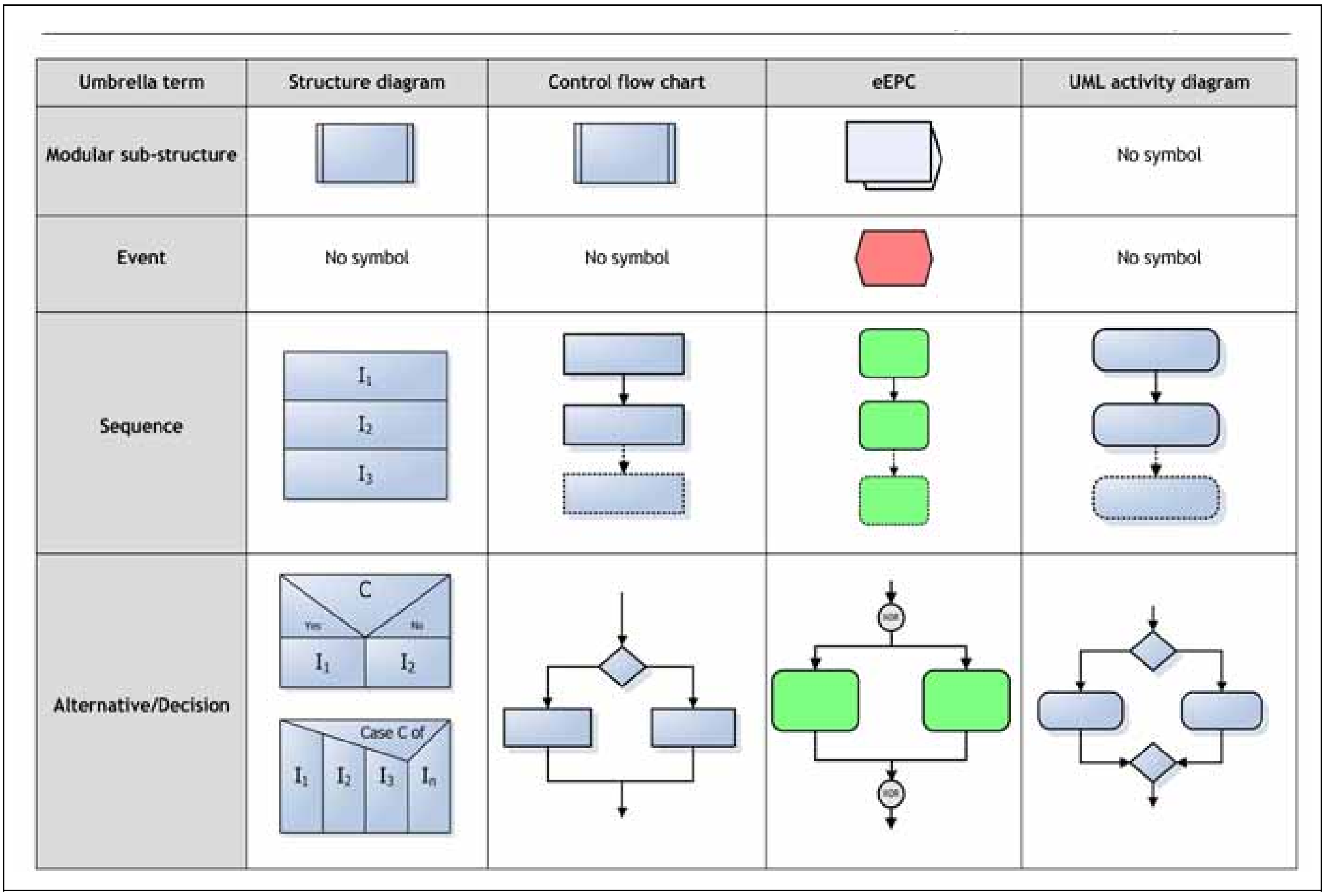
1.3 Basic components of process models 1

→ Aim: to convince the BPM community with the presentation of a detailed analogy

Umbrella term	BPM	Control flow modeling
Modular substructure	partial process	subprogram, subroutine
Event	business event	operating system event, interrupt
Sequence	sequence	sequence
Test, alternative, decision	XOR	IF
Iteration	cycle	loop
Simultaneity, parallelism	AND	parallel functions
Process unit	business activity	instruction or block of instructions

Analogy (umbrella terms) of the basic components of BPM and control flow modeling (Holl / Valentin 2004)

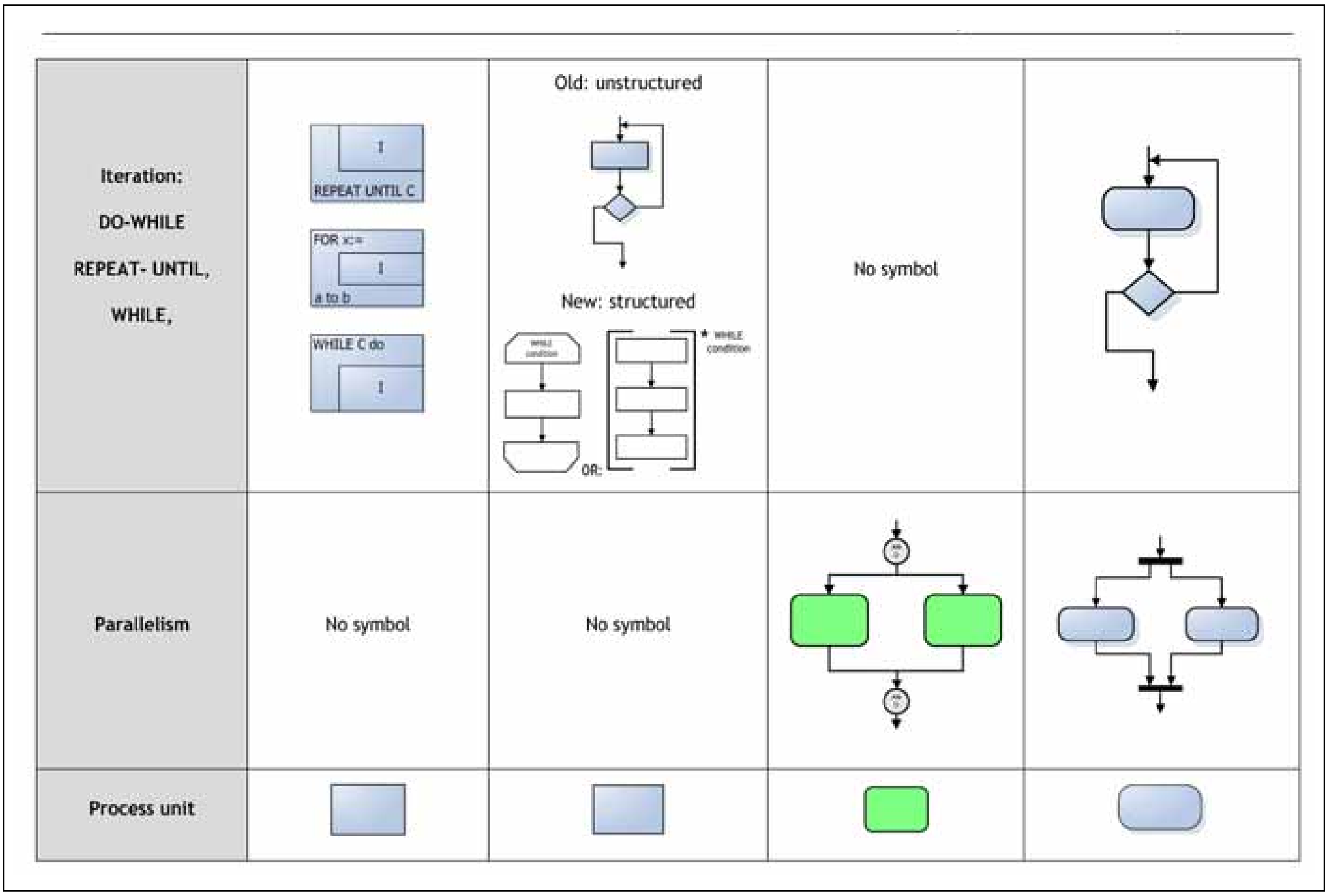
1.3 Basic components of process models 2



Analogy of the notations of BPM and control flow modeling (Grünauer 2008: 102 according to Holl / Valentin 2004)

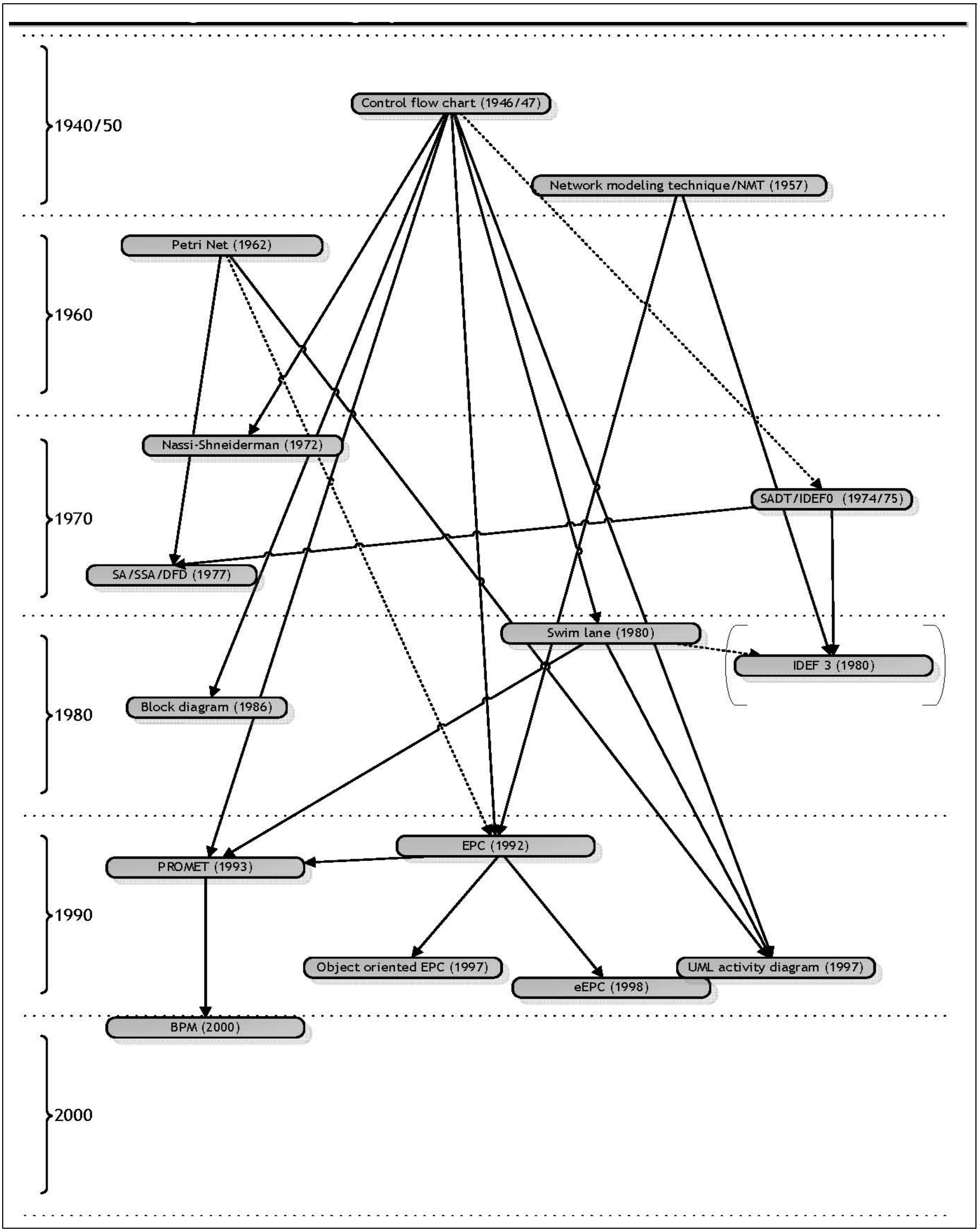
Structure diagram: DIN 66 261, according to Nassi-Shneiderman
Control flow chart: DIN 66001

1.3 Basic components of process models 3



Analogy of the notations of BPM and control flow modeling (Grünauer 2008: 102 according to Holl / Valentin 2004)

1.3 Genealogical tree of process notations



(Grünauer 2008: 30)

1.4 Process meta-model: elements 1

In the following, process meta-models will be examined from the point of view of information systems.

That is, there will be a focus on the activity-on-node variant.

The activity-on-arc variant (state transition networks, Petri nets), which is important for theoretical computer science approaches, will be excluded.

1.4 Process meta-model: elements 2

Nodes of a semantic network:

1. function, action (computer-aided or not)
function unit, function module

– **name from the view of the organization**
– **decomposition-marker: reference to subprocesses**
– **algorithm, internal logic in a note**
– **duration, start time, end time**
– **features, feature values (→ theory of gestalt)**
– **IT support: computer-aided or manual**

2. initiating and resulting events

3. actor: person/role/department responsible for the action
partly connected with data flow

4. external (business/communication) partners
connected with data flow

5. data stores accessed: input data and output data
connected with data flow

6. resources used (machines etc.)

	World 1 (reality)	World 3 (model)
single object, "instance"	one individual course of events in an organization	business process instance
set - type of similar objects	set of homogeneous courses of events	business process type

1.4 Process meta-model: elements 3

Arcs of a semantic network:

1. control flow: temporal interrelation of functions (cf. structured programming)

– temporal succession: sequence (predecessors and successors)

– condition: alternative, selection (IF, XOR)
case discrimination (CASE)
or complex rule (decision table)
disjoint and complete

– repetition: iteration, loop (WHILE or REPEAT)
test-first loop and test-last loop

– recursion

– simultaneousness: parallel processing (AND)

– coroutine: mutual call

CAUTION:
all control flow elements without the mere sequence must have
a divergent delimitor (begin) and
a convergent delimitor (end, synchronization);
the delimitors have to be arranged symmetrically in a diagram:
IF – ENDIF, CASE – ENDCASE, LOOP – ENDLOOP etc.

2. data flow (only partly)

3. mere connectors to actors and resources used

1.4 Process meta-model: special notations 1

1. Classical notations

1.1 Traditional notations for structured programming

flow chart, block diagram ('Programm-Ablauf-Plan')
structure diagram, structogram (Nassi-Shneiderman diagram)
Jackson tree
- Jackson structured design (JSD)
- Jackson structured programming (JSP)

functions and control flow

1.2 Decision table

complex conditions and functions: rules

1.3 Network model(ing technique)

functions, sequence, parallel processing,
duration, start time, end time
→ critical path

1.4 Control flow plus data flow

HIPO: hierarchy plus input-process-output (Mills 1972, IBM)
functions, control flow, data stores, data flow

1.4 Process meta-model: special notations 2

1.5 Swim lane diagram

functions, control flow, responsible departments
predecessor of UML activity diagram
Arbeitsablaufdiagramm: Arbeitsschritte – Abteilungen
Organisationsprozessdarstellung (H. F. Binner)

2. Business process models

Event-driven process chain

functions, control flow (ridiculous: no iterations!)
events
actors, partners, data stores, resources, data flow

3. Dynamic object models

UML activity diagram

functions, control flow
events
actors, partners, data stores, resources, data flow
swim lanes (responsible departments)

UML sequence diagram

classes, elementary functions called by messages, control flow

1.5 Conclusion

Changes to be made in BPM

– block structures:
BEGIN – END, LOOP – ENDLOOP,
IF(XOR) – ENDIF, CASE – ENDCASE
BEGIN OR – END OR, BEGIN AND – END AND

– corresponding notations for block structures:
divergent and convergent delimiters
symbol for iterations

– hierarchically nested structures (LIFO principle)

– vertical decomposition with motivated cuts
hierarchic modular structure

– transparent diagrams

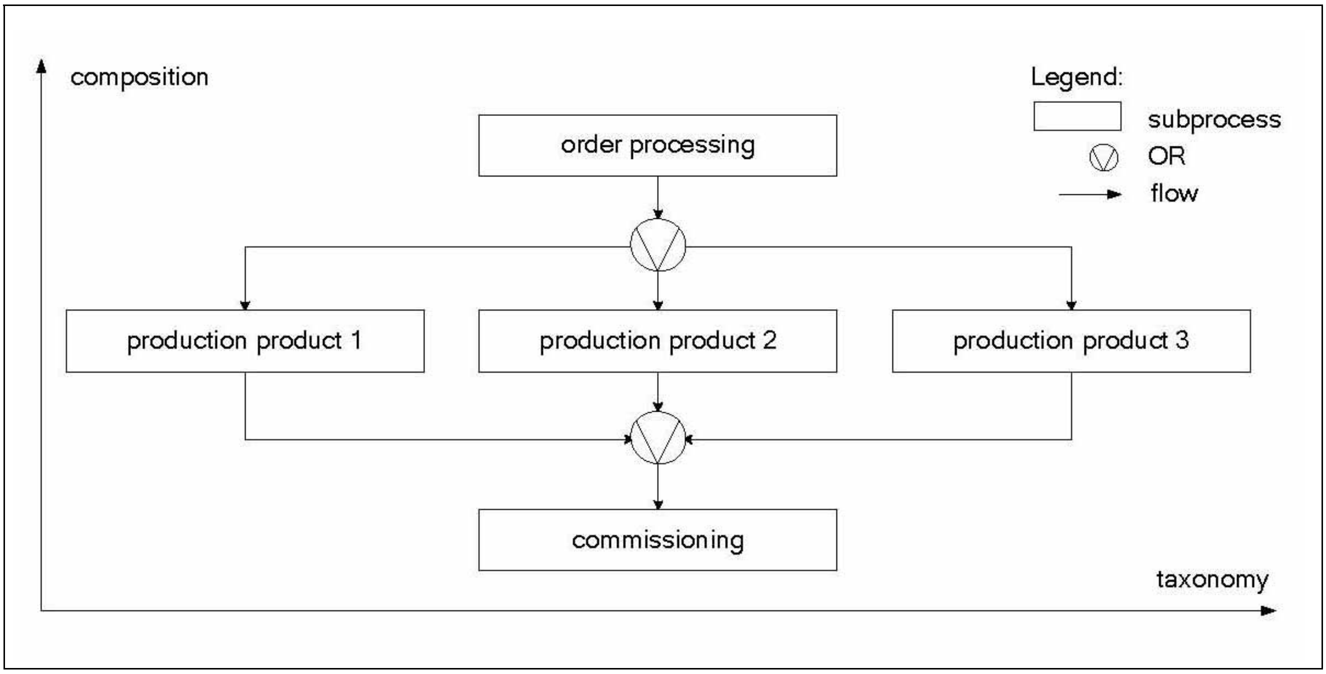
Advantages

– more transparent description of the reality
– easier optimization of BP models (BP reengineering)
– easier modification and adaptation of BP models
– more effective mapping to workflow management tools

– better, transparent basis of communication
– more effective requirements engineering
– better usable reference models

2 Structured business process decomposition

2.1 Motivation 1: teaser

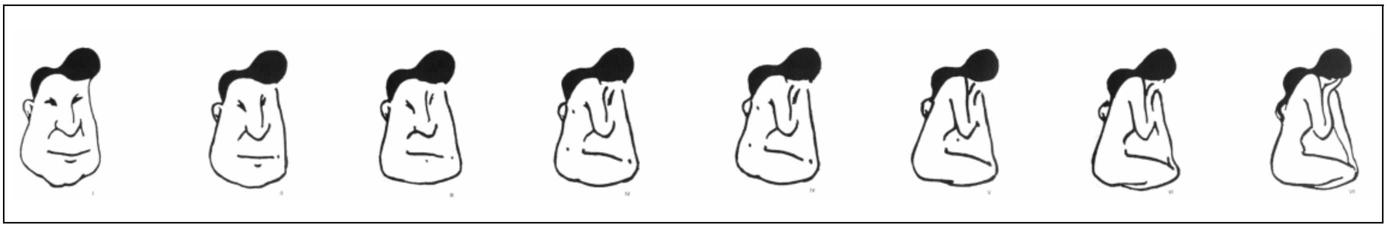


Decomposition of processes in sub-processes
(Holl / Krach / Mnich 2000, 198)

1 Decomposition in sequential sub-process (compositional)
2 Decomposition in parallel sub-processes (taxonomic)

The former is the subject of the following considerations.

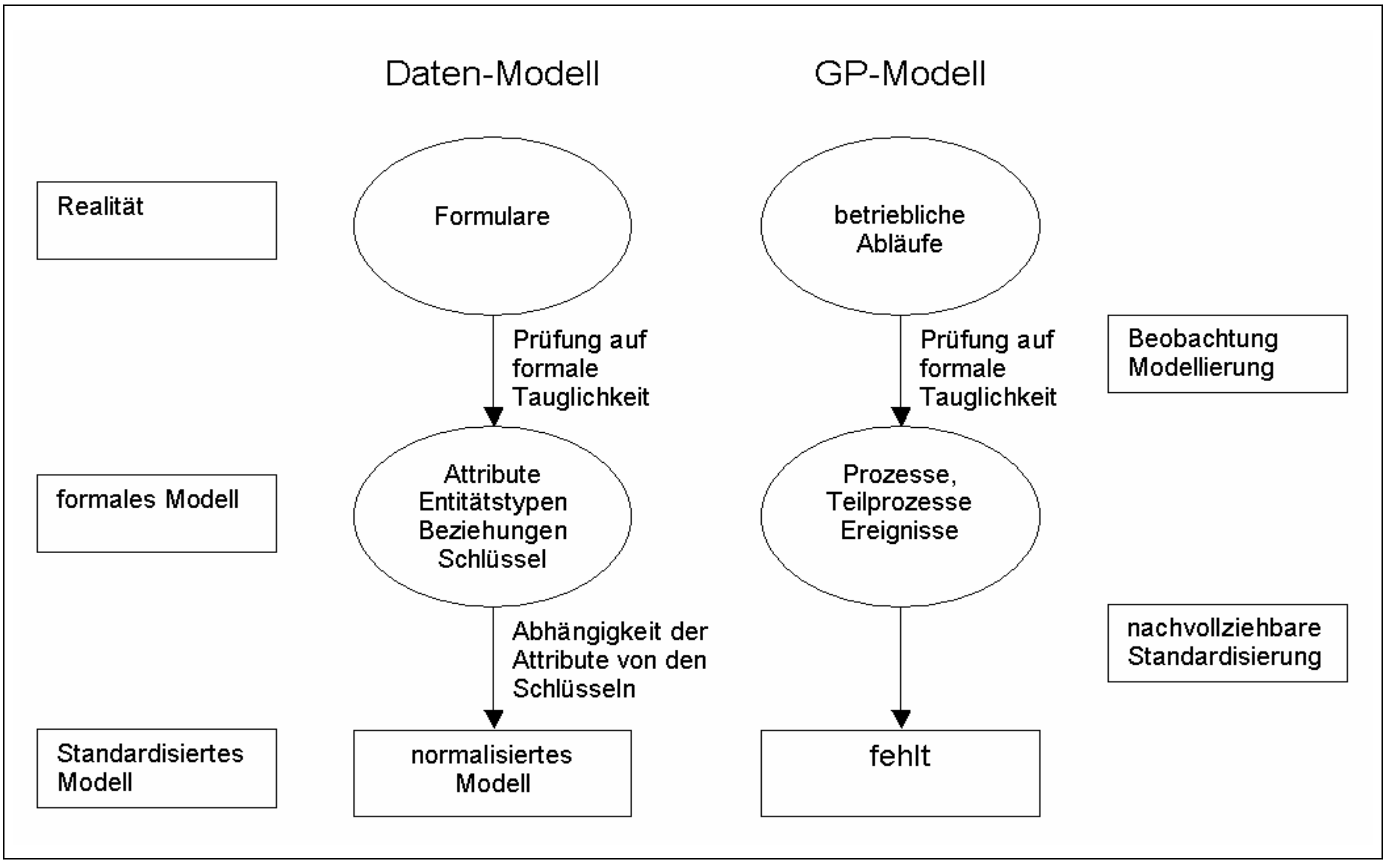
Where can the following process be divided into sub-processes?



A man's face to a woman's body (Riedl 1987: 74-77)

2.1 Motivation 2: two starting points and their synthesis

1 Different model designers construct different BP models vs. data and static OO models are more independent of designers



**Comparison between data and BP modeling:
a method analogous to normalization is missing
(Holl / Krach / Mnich 2000, 203)**

2 Examination of similarity and features as cognitive principles in evolutionary epistemology and theory of gestalt: becoming aware of decomposition features changes hypotheses of decomposition, of splitting points

**3 Aim / synthesis of the two starting points:
gestalt-theoretical business process decomposition:
processes are split up where a feature changes its value.**

2.2 Theory of gestalt 1

The theory of gestalt dates back to considerations of
- Johann Wolfgang von Goethe
- Christian von Ehrenfels
- Max Wertheimer

It is an interdisciplinary theory with applications in
- epistemology, psychology of perception
- biology
- pedagogic
- architecture, arts

The whole (semantics) is more than the sum of its parts (syntax).
'Forms' (German "Gestalten") can be
– static: physical objects
– dynamic: melody, ritual, process

What is this?

2.2 Theory of gestalt 2

Decomposition of static and dynamic ‘forms’ (“Gestalten”)

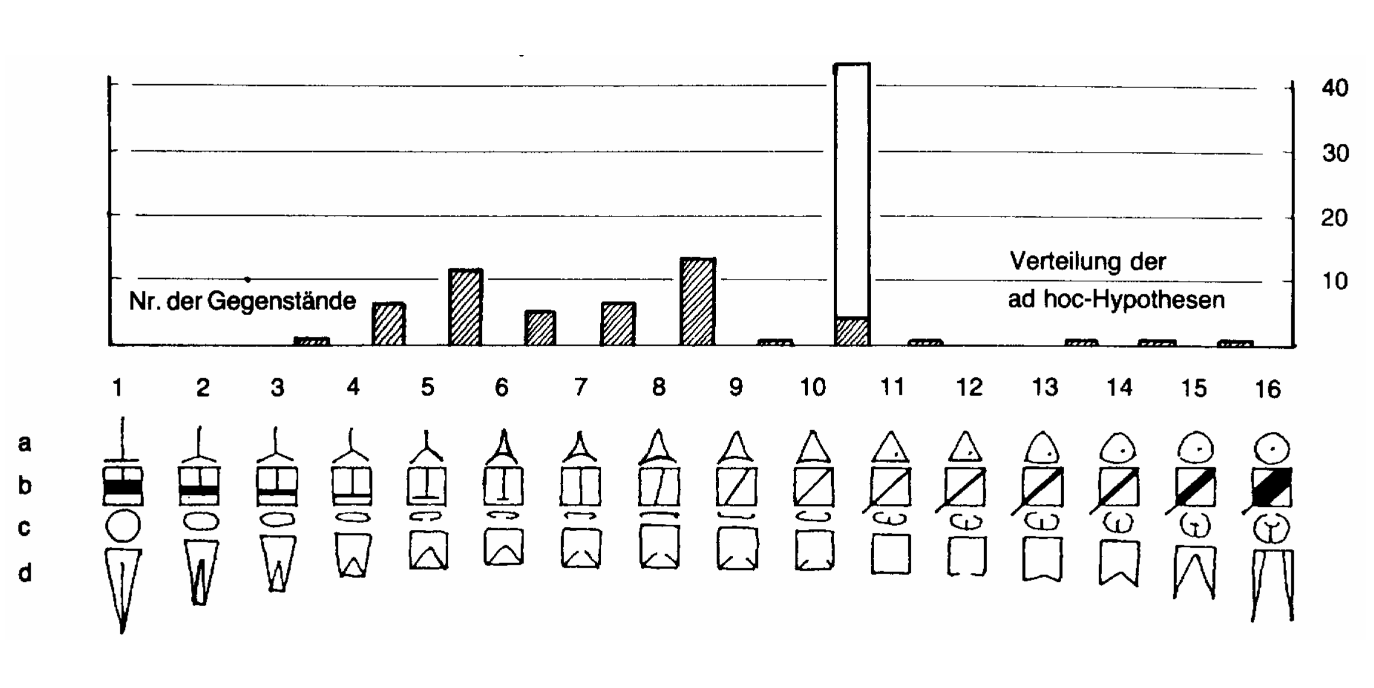
For humans, it is easy to decompose static ‘forms’ (pictures), difficult to decompose dynamic ‘forms’ (courses of events, business processes, morphing processes).

Features

A particularity or a property of a ‘form’ can be called a feature.

Rupert Riedl has systematically examined the idea of a feature in his book “Begriff und Welt” (‘Concept and reality’) 1987.

Riedl shows that features cannot only be used to find similarities between different static ‘forms’ but also to decompose / subdivide dynamic ‘forms’.



Splitting of a process according to changes of features (Riedl 1987: 195)

2.3 Business process decomposition and gestalt-theoretical features 1

BP decomposition is done using features.

**The model designer has to be aware of these features,
has to lift them from the unconscious to the conscious level
and has to make them explicit.**

**Thus, we obtain BP models which can be followed and,
therefore, be discussed and motivated.**

Possible features in business processes:
- responsible person
- order status
- machine

Processes are split up where a feature changes its value.

Relation between features and events

When a feature changes its value, an event happens.

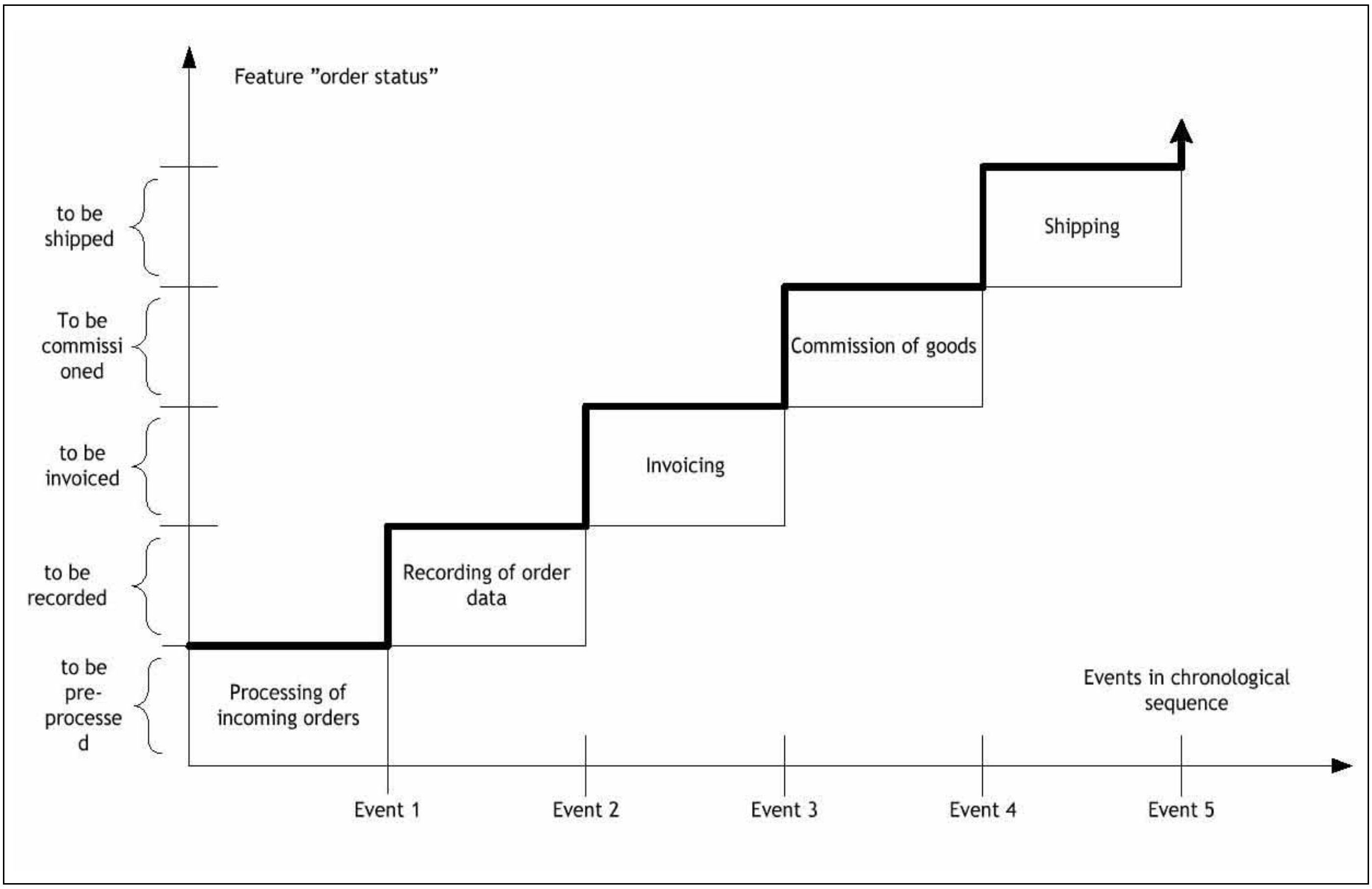
→ Feature-based event-driven process chains

**Models do not become better automatically, but
this approach makes it easier to discuss and,
thus, to improve them.**

2.3 Business process decomposition and gestalt-theoretical features 2

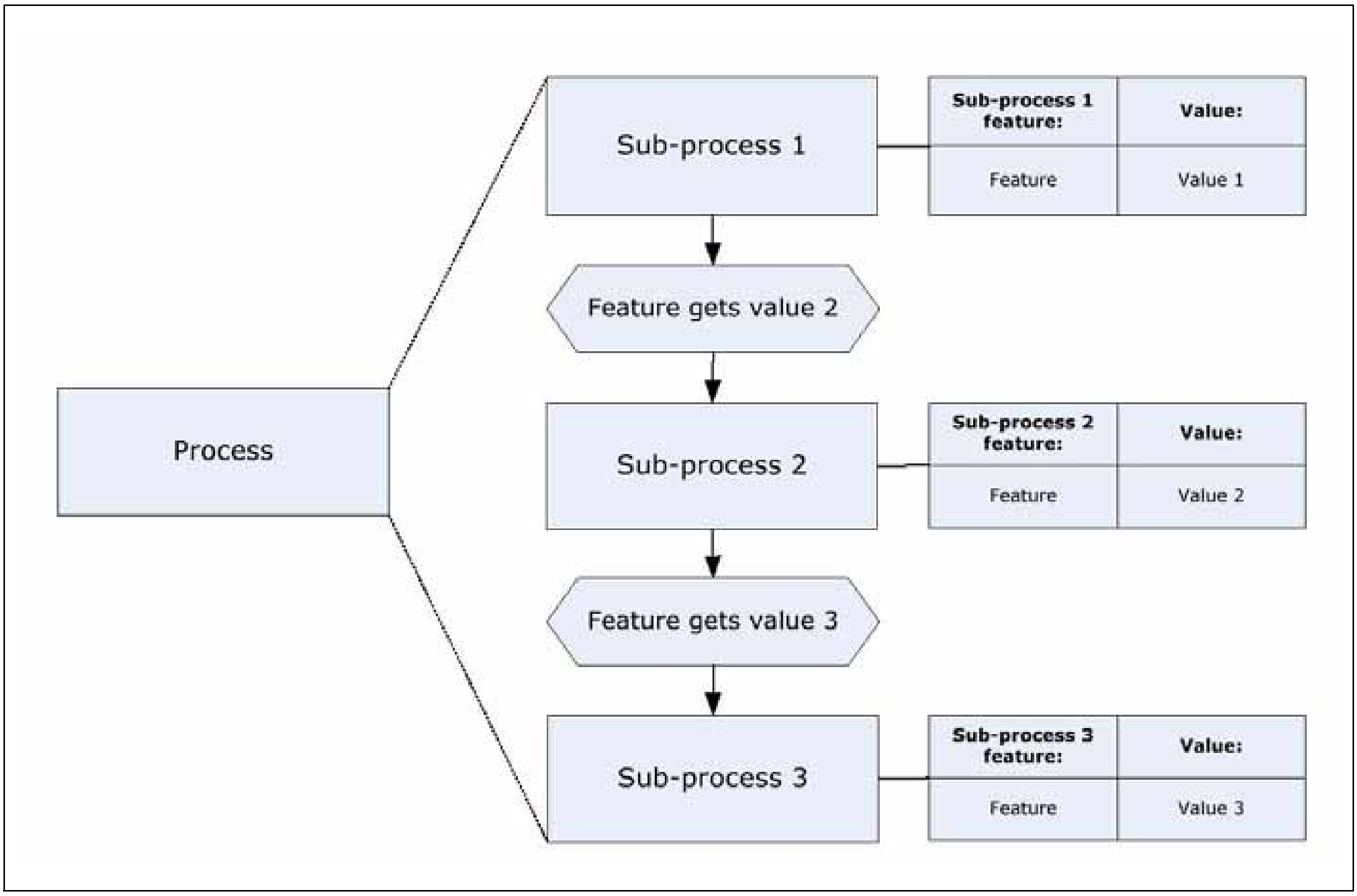
Sub-processes	Values of the feature "order status"
Order acceptance check	To be checked
Order data recording	To be recorded
Invoicing	To be invoiced
Commissioning	To be commissioned
Shipping	To be shipped

Sub-processes and their feature values



Changes of a feature visualized as mathematical step function (Holl / Krach / Mnich 2000, 207)

2.3 Business process decomposition and gestalt-theoretical features 3



Process representation
with sub-processes, events and features
(Holl / Krach / Mnich 2000, 208)

3 References

Böhm, Corrado; Jacopini, Giuseppe:
Flow diagrams, Turing machines and languages with only two formation rules.
***Communications of the ACM* 9(1966) 5, 366-371.**

Dijkstra, Edsger:
GOTO statement considered harmful.
***Communications of the ACM* 11(1968) 3, 147-148.**

Grünauer, Karin:
***Business process modeling*. Växjö (Master thesis) 2008**

Holl, Alfred; Valentin, Gregor:
Structured business process modeling.
Contribution to:
Information Systems Research in Scandinavia (IRIS'27),
Falkenberg/Sweden 2004, CD-ROM.

Holl, Alfred; Krach, Thomas; Mnich, Roman:
Geschäftsprozessmodellierung und Gestalttheorie.
In: Britzelmaier, Bernd et al. (ed.): *Information als Erfolgsfaktor. 2. Liechtensteinisches Wirtschaftsinformatik-Symposium an der FH Liechtenstein*.
Stuttgart: Teubner 2000, 197-209, ISBN 3-519-00317-1.

Lorenz, Konrad (1903-1989):
Gestalt perception as fundamental to scientific knowledge
[original 1959 in German: Gestaltwahrnehmung als Quelle wissenschaftlicher Erkenntnis. *Zeitschrift für experimentelle und angewandte Psychologie* 6(1959) 118-165].
***General systems* 7 (1962) 37-56 [= Bertalanffy, L. v.; Rapoport, A. (ed.): *Yearbook of the Society for General Systems Research*].**

Riedl, Rupert:
***Begriff und Welt – Biologische Grundlagen des Erkennens und Begreifens*. Berlin, Hamburg: Parey 1987.**

Yourdon, Edward:
***Modern structured analysis*. Englewood Cliffs NJ 1989.**

pdf-files of my own publications: see my homepage