INTERNATIONAL STANDARD

ISO 18629-42

First edition
2006-08-15

Industrial automation systems and integration — Process specification language —

Part 42: Definitional extension: Temporal and state extensions

Systèmes d'automatisation industrielle et intégration — Langage de spécification de procédé —

Partie 42: Extension de définition: Extensions temporelle et d'état

Reference number
ISO 18629-42:2006(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

1 Scope 1

2 Normative References 1

3 Terms, definitions and abbreviations 2

3.1 Terms and definitions 2

3.2 Abbreviations 5

4 Overview of parts 41 to 49 of ISO 18629 5

5 Organization of this part of ISO 18629 6

5.1 Introduction 6

5.2 Extensions in ISO 18629-42 6

6 State-based preconditions for activities 7

6.1 Primitive lexicon of the State-based preconditions for activities 7

6.2 Defined lexicon for concepts of State-based preconditions for activities 7

6.3 Core theories required by State-based preconditions for activities 8

6.4 Definitional extensions required by State-based preconditions for activities 8

6.5 Definitions of concepts for State-based preconditions for activities 8

6.5.1 state_equiv 8

6.5.2 markov_precond 8

6.5.3 partial_state 9

6.5.4 rigid_state 9

6.6 Grammar for relations of State-based preconditions for activities 9

7 Time-based preconditions for activities 10

7.1 Primitive lexicon of Time-based preconditions for activities 10

7.2 Defined lexicon for concepts of Time-based preconditions for activities 10

7.3 Theories required by Time-based preconditions for activities 11

7.4 Definitional extensions required by Time-based preconditions for activities 11

7.5 Definitions of Time-based preconditions for activities 11

7.5.1 begin_equiv 11

7.5.2 time_precond 11

7.5.3 partial_time_precond 11

7.5.4 rigid_time 12

7.6 Grammar for process descriptions of Time-based preconditions for activities 12

8 Preconditions based on state and time 13

8.1 Primitive lexicon of Preconditions based on state and time 13

8.2 Defined lexicon of Preconditions based on state and time 13
8.3 Theories required by Preconditions based on state and time 13
8.4 Definitional extensions required by Preconditions based on state and time 13
8.5 Definitions of Preconditions based on state and time 14
8.5.1 mixed_precond 14
8.5.2 partial_mixed 14
8.5.3 rigid_mixed 14
8.6 Grammar for Preconditions based on state and time 14
9 Occurrence-based preconditions for activities 15
9.1 Primitive lexicon of Occurrence-based preconditions for activities 15
9.2 Defined relations of Occurrence-based preconditions for activities 15
9.3 Theories required by Occurrence-based preconditions for activities 16
9.4 Definitional extensions required by Occurrence-based preconditions for activities 16
9.5 Definitions of Occurrence-based preconditions for activities 16
9.5.1 tree_equiv 16
9.5.2 occurrence_constrained 16
9.5.3 occurrence_dependent 17
9.5.4 occurrence_independent 17
9.6 Grammar for Occurrence-based preconditions for activities 17
10 Preventable conditions for activities 18
10.1 Primitive lexicon of Preventable conditions for activities 18
10.2 Defined relations of Preventable conditions for activities 18
10.3 Theories required by Preventable conditions for activities 18
10.4 Definitional extensions required by Preventable conditions for activities 18
10.5 Definitions of Preventable conditions for activities 19
10.5.1 preventable 19
10.5.2 possibly_preventable 19
10.5.3 unpreventable 19
10.6 Grammar for process descriptions of Preventable conditions for activities 20
11 Periodic preconditions for activities 20
11.1 Primitive lexicon of Periodic preconditions for activities 20
11.2 Defined lexicon of Periodic preconditions for activities 20
11.3 Theories required by Periodic preconditions for activities 20
11.4 Definitional extensions required by Periodic preconditions for activities 21
11.5 Definitions of Periodic preconditions for activities 21
11.5.1 periodic 21
11.5.2 intermittent 21
11.5.3 aperiodic 21

11.6 Grammar for of Periodic preconditions for activities ... 22

12 Spoilage preconditions for activities ... 22

12.1 Primitive lexicon of Spoilage preconditions for activities ... 22

12.2 Defined lexicon of Spoilage preconditions for activities ... 22

12.3 Theories required by Spoilage preconditions for activities ... 23

12.4 Definitional extensions required by Spoilage preconditions for activities ... 23

12.5 Definitions of Spoilage preconditions for activities ... 23

12.5.1 spoilage ... 23

12.5.2 possible_spoilage ... 23

12.5.3 nonspoilage ... 24

12.6 Grammar of Spoilage preconditions for activities ... 24

13 Effects of activities ... 24

13.1 Primitive lexicon of Effects of activities ... 24

13.2 Defined lexicon of Effects of activities ... 24

13.3 Theories required by Effects of activities ... 25

13.4 Definitional extensions required by Effects of activities ... 25

13.5 Definitions of Effects of activities ... 25

13.5.1 effects_equiv ... 25

13.5.2 context_free ... 25

13.5.3 null ... 25

13.6 Grammar for Effects of activities ... 26

14 State-based effects of activities ... 26

14.1 Primitive lexicon of State-based effects of activities ... 26

14.2 Defined lexicon of State-based effects of activities ... 26

14.3 Core theories required by State-based effects of activities ... 27

14.4 Definitional extensions required by State-based effects of activities ... 27

14.5 Definitions of State-based effects of activities ... 27

14.5.1 markov_effects ... 27

14.5.2 partial_state_effects ... 27

14.5.3 rigid_state_effects ... 28

14.6 Grammar for State-based Effects Activities. ... 28

15 Time-based effects of activities ... 28

15.1 Primitive lexicon of Time-based effects of activities ... 28

15.2 Defined lexicon of Time-based effects of activities ... 28

15.3 Core theories required by Time-based effects of activities ... 29

15.4 Definitional extensions required by Time-based effects of activities ... 29

15.5 Definitions of Time-based effects of activities 29
15.5.1 temporal_effects 29
15.5.2 partial_temporal 29
15.5.3 nontemporal 29
15.6 Grammar for Time-based effects of activities 30
16 Occurrence-based effects of activities 30
16.1 Primitive lexicon of Occurrence-based effects of activities 30
16.2 Defined lexicon of Occurrence-based effects of activities 30
16.3 Core theories required by Occurrence-based effects of activities 30
16.4 Definitional extensions required by Occurrence-based effects of activities 31
16.5 Definitions of Occurrence-based effects of activities 31
16.5.1 occ_effects 31
16.5.2 occ_depend_effects 31
16.5.3 nonocc_effects 31
16.6 Grammar for Occurrence-based effects of activities 32
17 Effects of activities: Occurrence Constraints 32
17.1 Primitive lexicon of Effects of activities: Occurrence Constraints 32
17.2 Defined lexicon of Effects of activities: Occurrence Constraints 32
17.3 Core theories required by Effects of activities: Occurrence Constraints 32
17.4 Definitional extensions required by Effects of activities: Occurrence Constraints 32
17.5 Definitions of Effects of activities: Occurrence Constraints 33
17.5.1 quantum 33
17.5.2 semiclassical 33
17.5.3 classical 33
17.6 Grammar for Effects of activities: Occurrence Constraints 34
18 Effects of activities: Temporal and Occurrence Constraints 34
18.1 Primitive lexicon of Effects of activities: Temporal and Occurrence Constraints 34
18.2 Defined lexicon of Effects of activities: Temporal and Occurrence Constraints 34
18.3 Core theories required by Effects of activities: Temporal and Occurrence Constraints 34
18.4 Definitional extensions required by Effects of activities: Temporal and Occurrence Constraints 35
18.5 Definitions of Effects of activities: Temporal and Occurrence Constraints 35
18.5.1 relativistic 35
18.5.2 seminewton 35
18.5.3 newtonian 35
18.6 Grammar for Effects of activities: Temporal and Occurrence Constraints 36
19 Fluent trees 36

19.1 Primitive lexicon of Fluent trees 36
19.2 Defined relations of Fluent trees 36
19.3 Core theories required by Fluent trees 36
19.4 Definitional extensions required by Fluent trees 37
19.5 Definitions of Fluent trees 37
19.5.1 achieved 37
19.5.2 falsified 37
19.5.3 irreversible 37
19.5.4 unachievable 37
19.5.5 bounded 38
19.6 Grammar for process descriptions of Fluent trees 38
20 Distribution of complex activities 38
20.1 Primitive lexicon of Distribution of complex activities 38
20.2 Defined relations of Distribution of complex activities 38
20.3 Core theories required by Distribution of complex activities 38
20.4 Definitional extensions required by Distribution of complex activities 38
20.5 Definitions of Distribution of complex activities 39
20.5.1 profile 39
20.5.2 root_equiv 39
20.5.3 universal 39
20.5.4 local 40
20.5.5 restricted 40
20.6 Grammar for process descriptions of Distribution of complex activities 40
21 State-based distribution of complex activities 41
21.1 Primitive lexicon of State-based distribution of complex activities 41
21.2 Defined relations of State-based distribution of complex activities 41
21.3 Theories required by State-based distribution of complex activities 41
21.4 Definitional extensions required by State-based distribution of complex activities 41
21.5 Definitions of State-based distribution of complex activities 41
21.5.1 trigger 42
21.5.2 partial_trigger 42
21.5.3 nontrigger 42
21.6 Grammar for process descriptions of State-based distribution of complex activities ... 43
22 Time-based distribution of complex activities 43
22.1 Primitive lexicon of Time-based distribution of complex activities 43
22.2 Defined relations of Time-based distribution of complex activities 43
22.3 Theories required by Time-based distribution of complex activities 43
22.4 Definitional extensions required by Time-based distribution of complex activities 43

22.5 Definitions of Time-based distribution of complex activities 44
22.5.1 launch 44
22.5.2 partial_launch 44
22.5.3 rigid_launch 44
22.6 Grammar for process descriptions of Time-baseddistribution of complex activities ... 45
23 Mixed distribution of complex activities 45
23.1 Primitive lexicon of Mixed distribution of complex activities 45
23.2 Defined relations of Mixed distribution of complex activities 45
23.3 Core theories required by Mixed distribution of complex activities 45
23.4 Definitional extensions required by Mixed distribution of complex activities........ 45
23.5 Definitions of Mixed distribution of complex activities 46
23.5.1 conditional_launch 46
23.5.2 partial_conditional_launch 46
23.5.3 unconditional_launch 47
23.6 Grammar for process descriptions of Mixed distribution of complex activities 47
24 Variation of complex activities 47
24.1 Primitive lexicon of Variation of complex activities 47
24.2 Defined relations of Variation of complex activities 47
24.3 Theories required by Variation of complex activities 48
24.4 Definitional extensions required by Variation of complex activities 48
24.5 Definitions of Variation of complex activities 48
24.5.1 min_equiv 48
24.5.2 uniform 48
24.5.3 variegated 48
24.5.4 multiform 49
24.6 Grammar for process descriptions of Variation of complex activities 49
25 State-based variation of complex activities 50
25.1 Primitive lexicon of State-based variation of complex activities 50
25.2 Defined relations of State-based variation of complex activities 50
25.3 Theories required by State-based variation of complex activities 50
25.4 Definitional extensions required by State-based variation of complex activities 50
25.5 Definitions of State-based variation of complex activities 50
25.5.1 conditional 51
25.5.2 partial_conditional 51
25.5.3 rigid_conditional 51
25.6 Grammar for process descriptions of State-based variation of complex activities 52
26 Time-based variation of complex activities 52

26.1 Primitive lexicon of Time-based variation of complex activities 52
26.2 Defined relations of Time-based variation of complex activities 52
26.3 Theories required by Time-based variation of complex activities 52
26.4 Definitional extensions required by Time-based variation of complex activities 52
26.5 Definitions of Time-based variation of complex activities 53
26.5.1 time_conditional 53
26.5.2 partial_time_conditional 53
26.5.3 rigid_time_conditional 53
26.6 Grammar for process descriptions of Time-based variation of complex activities 54
27 Mixed variation of complex activities 54
27.1 Primitive lexicon of Mixed variation of complex activities 54
27.2 Defined relations of Mixed variation of complex activities 54
27.3 Theories required by Mixed variation of complex activities 55
27.4 Definitional extensions required by Mixed variation of complex activities 55
27.5 Definitions of Mixed variation of complex activities 55
27.5.1 mixed_conditional 55
27.5.2 partial_mixed_conditional 55
27.5.3 rigid_mixed_conditional 56
27.6 Grammar for process descriptions of Mixed variation of complex activities 56
28 Embedded activities: plans 56
28.1 Primitive lexicon of Embedded activities: plans 57
28.2 Defined relations of Embedded activities: plans 57
28.3 Theories required by Embedded activities: plans 57
28.4 Definitional extensions required by Embedded activities: plans 57
28.5 Definitions of Embedded activities: plans 57
28.5.1 plan 57
28.5.2 nondet_plan 58
28.5.3 unplan 58
28.6 Grammar for process descriptions of Embedded activities: plans 58
29 Embedded activities: temporal spread 59
29.1 Primitive lexicon of Embedded activities: temporal spread 59
29.2 Defined relations of Embedded activities: temporal spread 59
29.3 Theories required by Embedded activities: temporal spread 59
29.4 Definitional extensions required by Embedded activities: temporal spread 59
29.5 Definitions of Embedded activities: temporal spread 59
29.5.1 spread 60
29.5.2 partial_spread 60

29.5.3 tight ... 60
29.6 Grammar for process descriptions of Embedded activities: temporal spread ... 61
30 Variation for upwards atomic activities ... 61
30.1 Primitive lexicon of Variation for upwards atomic activities ... 61
30.2 Defined relations of Variation for upwards atomic activities ... 61
30.3 Theories required by Variation for upwards atomic activities ... 62
30.4 Definitional extensions required by Variation for upwards atomic activities ... 62
30.5 Definitions of Variation for upwards atomic activities ... 62
30.5.1 state_filter ... 62
30.5.2 partial_state_filter ... 63
30.5.3 rigid_state_filter ... 63
30.5.4 time_filter ... 63
30.5.5 partial_time_filter ... 64
30.5.6 rigid_time_filter ... 64
30.5.7 state_nonfilter ... 65
30.5.8 partial_state_nonfilter ... 65
30.5.9 rigid_state_nonfilter ... 65
30.5.10 time_nonfilter ... 66
30.5.11 partial_time_nonfilter ... 66
30.5.12 rigid_time_nonfilter ... 66
30.6 Grammar for process descriptions of Variation for upwards atomic activities ... 67
31 Variation for downwards atomic activities ... 67
31.1 Primitive lexicon of Variation for downwards atomic activities ... 67
31.2 Defined relations of Variation for downwards atomic activities ... 67
31.3 Theories required by Variation for downwards atomic activities ... 68
31.4 Definitional extensions required by Variation for downwards atomic activities ... 68
31.5 Definitions of Variation for downwards atomic activities ... 68
31.5.1 state_ideal ... 68
31.5.2 partial_state_ideal ... 68
31.5.3 rigid_state_ideal ... 69
31.5.4 time_ideal ... 69
31.5.5 partial_time_ideal ... 69
31.5.6 rigid_time_ideal ... 70
31.5.7 state_nonideal ... 70
31.5.8 partial_state_nonideal ... 71
31.5.9 rigid_state_nonideal ... 71
31.5.10 time_nonideal ... 71
31.5.11 partial_time_nonideal ... 72
31.5.12 rigid_time_nonideal ... 72
31.6 Grammar for process descriptions of Variation for downwards atomic activities ... 73
32 Preconditions for concurrent activities ... 73

32.1 Primitive lexicon of Preconditions for concurrent activities 73
32.2 Defined relations of Preconditions for concurrent activities 73
32.3 Theories required by Preconditions for concurrent activities 73
32.4 Definitional extensions required by Preconditions for concurrent activities 73
32.5 Definitions of Preconditions for concurrent activities 73
32.5.1 preserved_filter 74
32.5.2 noninterfering 74
32.5.3 interfering 74
32.5.4 imperial 74
32.5.5 global_interfere 74
32.6 Grammar for process descriptions of Preconditions for concurrent activities 75
33 Effects for concurrent activities 75
33.1 Primitive lexicon of Effects for concurrent activities 75
33.2 Defined relations of Effects for concurrent activities 75
33.3 Theories required by Effects for concurrent activities 75
33.4 Definitional extensions required by Effects for concurrent activities 75
33.5 Definitions of Effects for concurrent activities 76
33.5.1 preserved_effects 76
33.5.2 nonclobbering 76
33.5.3 clobbering 76
33.5.4 meddling 76
33.5.5 global_clobber 77
33.6 Grammar for process descriptions of Effects for concurrent activities 77
34 Variation of interfering preconditions 77
34.1 Primitive lexicon of Variation of interfering preconditions 77
34.2 Defined relations of Variation of interfering preconditions 77
34.3 Theories required by Variation of interfering preconditions 77
34.4 Definitional extensions required by Variation of interfering preconditions 78
34.5 Definitions of Variation of interfering preconditions 78
34.5.1 state_interfere 78
34.5.2 partial_interfere 78
34.5.3 unconditional_interfere 79
34.5.4 time_interfere 79
34.5.5 sometime_interfere 79
34.5.6 rigid_interfere 80
34.6 Grammar for process descriptions of Variation of interfering preconditions 80
35 Variation off clobbering effects 80
35.1 Primitive lexicon of Variation off clobbering effects 80
35.2 Defined relations of Variation off clobbering effects 80

35.3 Theories required by Variation off clobbering effects 80
35.4 Definitional extensions required by Variation off clobbering effects 81
35.5 Definitions of Variation of clobbering effects 81
35.5.1 state_clobber 81
35.5.2 partial_state_clobber 81
35.5.3 unconditional_clobber 82
35.5.4 time_clobber 82
35.5.5 sometime_clobber 82
35.5.6 rigid_clobber 83
35.6 Grammar for process descriptions of Variation off clobbering effects 83
Annex A (normative) ASN.1 Identifier of ISO 18629-42 84
Annex B (informative) Example of process description using ISO 18629-42 85
Bibliography 94
Index 95

Figures :

Figure B.1: TOP level process for manufacturing a GT350 86
Figure B.2: PROCESS for manufacturing the 350–Engine 87
Figure B.3: PROCESS for manufacturing the 350–Block 89
Figure B.4: PROCESS for manufacturing the 350–Harness 90
Figure B.5: PROCESS for manufacturing the harness wire 91
Figure B.6 : Process for manufacturing the 350-Wire 92

Foreword

The International Organisation for Standardisation (ISO) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organisations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

This part of ISO 18629 was prepared by the Technical committee ISO/TC184 *Industrial automation systems and integration,* Subcommittee SC4, *Industrial data.*

A complete list of parts of ISO 18629 is available from the Internet:

http://www.tc184-sc4.org/titles/

Introduction

ISO 18629 is an International Standard for the computer-interpretable exchange of information related to manufacturing processes. Taken together, all the parts contained in the ISO 18629 Standard provide a generic language for describing a manufacturing process throughout the entire production process within the same industrial company or across several industrial sectors or companies, independently from any particular representation model. The nature of this language makes it suitable for sharing process specifications and properties related to manufacturing during all the stages of a production process.

This part of ISO 18629 provides a description of the definitional extensions of the language related to activity extensions defined within ISO 18629.

All parts of ISO 18629 are independent of any specific process representation model used in a given application. Collectively, they provide a structural framework for improving the interoperability of these applications.

Industrial automation systems and integration — Process specification language —

Part 42: Definitional extension: Temporal and state extensions

1 Scope

This part of ISO 18629 provides a specification of non-primitive concepts of the language, using a set of definitions written in the language of ISO 18629. These definitions provide axioms for terminology in this part of ISO 18629.

The following is within the scope of this part of ISO 18629:

— definitions of state and time-related concepts specified using concepts of ISO 18629-11 and ISO 18629-12;

— constraints on the occurrences of activities that are expressed using time relations from ISO 18629-11 and state relations from ISO 18629-12.

The following is outside the scope of this part of ISO 18629:

— definitions of concepts specified using concepts of ISO 18629-11 and ISO 18629-12 that are independent of state and time relations.

NOTE: state and time-related concepts are concepts that are related to the time and the status of any entity related to manufacturing process and needed to specify it.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824-1, *Information technology - Abstract Syntax Notation One (ASN.1) - Part 1: Specification of basic notation*

ISO 10303-1, *Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles*

ISO 15531-1, *Industrial automation systems and integration - Industrial manufacturing management data - Part 1: General overview*

ISO 18629-1: 2004, *Industrial automation systems and integration – Process specification language – Part 1: Overview and basic principles*

ISO 18629-11: 2005, *Industrial automation systems and integration – Process specification language – Part 11 : PSL core*

ISO 18629-12, *Industrial automation systems and integration – Process specification language – Part 12 : Outer core*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply:

3.1.1
axiom
well-formed formula in a formal language that provides constraints on the interpretation of symbols in the lexicon of a language

[ISO 18629-1]

3.1.2
defined lexicon
set of symbols in the non-logical lexicon which denote defined concepts

NOTE Defined lexicon is divided into constant, function and relation symbols.

EXAMPLE terms with conservative definitions.

[ISO 18629-1]

3.1.3
definitional extension
extension of PSL-Core that introduces new linguistic items which can be completely defined in terms of the PSL-Core

NOTE: Definitional extensions add no new expressive power to PSL-Core but are used to specify the semantics and terminology in the domain application.

[ISO 18629-1]

3.1.4
extension
augmentation of PSL-Core containing additional axioms

NOTE 1 The PSL-Core is a relatively simple set of axioms that is adequate for expressing a wide range of basic processes. However, more complex processes require expressive resources that exceed those of the PSL-Core. Rather than clutter the PSL-Core itself with every conceivable concept that might prove useful in describing one process or another, a variety of separate, modular extensions need to be developed and added to the PSL-Core as necessary. In this way a user can tailor the language precisely to suit his or her expressive needs.

NOTE 2 All extensions are core theories or definitional extensions.

3.1.5
grammar
specification of how logical symbols and lexical terms can be combined to make well-formed formulae

[ISO 18629-1]

3.1.6
information
facts, concepts, or instructions

[ISO 10303-1]

3.1.7
language
combination of a lexicon and a grammar

[ISO 18629-1]

3.1.8
manufacturing
function or act of converting or transforming material from raw material or semi-finished state to a state of further completion

[ISO 15531-1]

3.1.9
manufacturing process
structured set of activities or operations performed upon material to convert it from the raw material or a semifinished state to a state of further completion

NOTE Manufacturing processes may be arranged in process layout, product layout, cellular layout or fixed position layout. Manufacturing processes may be planned to support make-to-stock, make-to-order, assemble-to-order, etc., based on strategic use and placements of inventories.

[ISO 15531-1]

3.1.10
model
combination of a set of elements and a truth assignment that satisfies all well-formed formulae in a theory

NOTE 1 The word "model" is used, in logic, in a way that differs from the way it is used in most scientific and everyday contexts : if a sentence is true in a certain interpretation, it is possible to say that the interpretation is a model of the sentence. The kind of semantics presented here is often called model-theoretical semantics.

NOTE 2 A model is typically represented as a set with some additional structure (partial ordering, lattice, or vector space). The model then defines meanings for the terminology and a notion of truth for sentences of the language in terms of this model. Given a model, the underlying set of axioms of the mathematical structures used in the set of axioms then becomes available as a basis for reasoning about the concepts intended by the terms of the language and their logical relationships, so that the set of models constitutes the formal semantics of the ontology.

[ISO 18629-1]

3.1.11
ontology
a lexicon of specialised terminology along with some specification of the meaning of terms in the lexicon

NOTE 1: structured set of related terms given with a specification of the meaning of the terms in a formal language. The specification of meaning explains why and how the terms are related and conditions how the set is partitioned and structured.

NOTE 2: The primary component of a process specification language such as ISO 18629 is an ontology. The primitive concepts in the ontology according to ISO 18629 are adequate for describing basic manufacturing, engineering, and business processes.

NOTE 3: The focus of an ontology is not only on terms, but also on their meaning. An arbitrary set of terms is included in the ontology, but these terms can only be shared if there is an agreement about their meaning. It is the intended semantics of the terms that is being shared, not simply the terms.

NOTE 4: Any term used without an explicit definition is a possible source of ambiguity and confusion. The challenge for an ontology is that a framework is needed for making explicit the meaning of the terms within it. For the ISO 18629 ontology, it is necessary to provide a rigorous mathematical characterisation of process information as well as a precise expression of the basic logical properties of that information in the ISO 18629 language.

NOTE 5 In practice, extensions incorporate the axioms of the Outer Core.

[ISO 18629-1]

3.1.12
primitive concept
lexical term that has no conservative definition

[ISO 18629-1]

3.1.13
primitive lexicon
set of symbols in the non-logical lexicon which denote primitive concepts

NOTE Primitive lexicon is divided into constant, function and relation symbols.

[ISO 18629-1]

3.1.14
process
structured set of activities involving various enterprise entities, that is designed and organised for a given purpose

NOTE The definition provided here is very close to that given in ISO 10303-49. Nevertheless ISO 15531 needs the notion of structured set of activities, without any predefined reference to the time or steps. In addition, from the point of view of flow management, some empty processes may be needed for a synchronisation purpose although they are not actually doing anything (ghost task).

[ISO 15531-1]

3.1.15
product
a thing or substance produced by a natural or artificial process

[ISO 10303-1]

3.1.16
resource
any device, tool and means, excepted raw material and final product components, at the disposal of the enterprise to produce goods or services

NOTE 1 Resources as they are defined here include human resources considered as specific means with a given capability and a given capacity. Those means are considered as being able to be involved in the manufacturing process through assigned tasks. That does not include any modelling of an individual or common behaviour of human resource excepted in their capability to perform a given task in the manufacturing process (e.g.: transformation of raw material or component, provision of logistic services). That means that human resources are only considered, as the other, from the point of view of their functions, their capabilities and their status (e.g.: idle, busy). That excludes any modelling or representation of any aspect of individual or common «social» behaviour.

NOTE 2 This definition includes ISO 10303-49 definition but is included in the definition that applies for ISO 18629-14 and ISO 18629-44 that includes raw materials and consumables.

[ISO 15531-1]

3.1.17
theory
set of axioms and definitions that pertain to a given concept or set of concepts

NOTE this definition reflects the approach of artificial intelligence in which a theory is the set of assumptions on which the meaning of the related concept is based.

[ISO 18629-1]

3.2 Abbreviations

— **KIF** Knowledge Interchange Format.

4 Overview of parts 41 to 49 of ISO 18629[1]

Parts 41 to 49 of ISO 18629 specify definitional extensions needed to give precise definitions and related axioms of non-primitive concepts of ISO 18629. Definitional extensions are extensions of ISO 18629-11 and ISO 18629-12 that introduce new items for the lexicon. The items found in definitional extensions can be completely defined using theories of ISO 18629-11 and ISO 18629-12. The definitional extensions provide precise semantic definitions for elements used in the specification of individual applications or types of applications for the purpose of interoperability. Definitional extensions exist in the following categories:

- Activity Extensions;
- Temporal and State Extensions;
- Activity Ordering and Duration Extensions;
- Resource Roles;
- Resource Sets;

[1] Certain parts are under development

- Processor Activity Extensions.

Individual users or groups of users of ISO 18629 may need to extend ISO 18629 for specifying concepts that are currently absent in parts 41 to 49 of ISO 18629. They shall use the elements presented in ISO 18629 for doing so. User-defined extensions and their definitions constitute definitional extensions but shall not become part of parts 41 to 49 of ISO 18629.

NOTE: User-defined extensions must conform to ISO 18629 as defined in ISO 18629-1 : 2004, 5.1 and 5.2.

The following are within the scope of parts 41 to 49 of ISO 18629 :

- the semantic definitions, using concepts in ISO 18629-11and ISO 18629-12, of elements that are specific to the six concepts outlined above;
- a set of axioms for constraining the use of elements in definitional extensions.

The following are outside the scope of parts 41 to 49 of ISO 18629:

- definitions and axioms for concepts that are part of ISO18629-11 and ISO 18629-12;
- elements that are not defined using the elements in ISO18629-11 and ISO 18629-12;
- user-defined extensions.

5 Organization of this part of ISO 18629

5.1 Introduction

This clause specifies the fundamental theories of which this part of ISO 18629 is composed.

5.2 Extensions in this part of ISO 18629

The fundamental theories that constitute this part of ISO 18629 are:

- State-based preconditions for activities;
- Time-based preconditions for activities;
- Preconditions based on state and time;
- Occurrence-based preconditions for activities;
- Preventable conditions for activities;
- Periodic preconditions for activities;
- Spoilage preconditions for activities;
- Effects of activities;
- Effects of activities: occurrence constraints;

- Effects of activities: temporal and occurrence constraints;
- Fluent trees;
- Distribution of complex activities;
- State-based distribution of complex activities;
- Time-based distribution of complex activities;
- Mixed distribution of complex activities;
- Variation of complex activities;
- State-based variation of complex activities;
- Time-based variation of complex activities;
- Mixed variation based of complex activities;
- Embedded activities: plans;
- Embedded activities: temporal spread;
- Variation for upwards atomic activities;
- Variation for downwards atomic activities;
- Preconditions for concurrent activities;
- Effects for concurrent activities;
- Variation of interfering preconditions;
- Variation off clobbering effects.

All definitional extensions in this part of ISO 18629 are extensions of ISO 18629-12, itself an extension of ISO 18629-11.

6 State-based preconditions for activities

This clause characterizes all definitions pertaining to state-based preconditions for activities. The criterion used to classify these activities is whether or not legal occurrences of an activity depend only on the state prior to the occurrences.

6.1 Primitive lexicon of the State-based preconditions for activities

No primitive relations are required by the lexicon of State-based preconditions for activities.

6.2 Defined lexicon for concepts of State-based preconditions for activities

The following relations are defined in this clause:

- (state_equiv ?s1 ?s2);
- (markov_precond ?a);
- (partial_state ?a);
- (rigid_state ?a).

Each concept is described by informal semantics and a KIF axiom.

6.3 Core theories required by State-based preconditions for activities

This extension requires:

- act_occ.th;
- complex.th;
- atomic.th;
- subactivity.th;
- disc_state.th;
- occtree.th;
- psl_core.th.

6.4 Definitional extensions required by State-based preconditions for activities

The following definitional extension is required by the State-based preconditions for activities:

- precond.def.

6.5 Definitions of concepts for State-based preconditions for activities

The following concepts are defined for State-based preconditions for activities.

6.5.1 state_equiv

Two activity occurrences are state equivalent if the same states hold after the occurrences.

```
(forall (?s1 ?s2) (iff (state_equiv ?s1 ?s2)
(forall (?f)
        (iff    (holds ?f ?s1)
          (holds ?f ?s2)))))
```

6.5.2 markov_precond

An activity is a markov preconditional activity if whenever two occurrences agree on state, then they agree on the extension of poss for the activity.

```
(forall (?a) (iff (markov_precond ?a)
(forall (?s1 ?s2)
    (implies  (state_equiv ?s1 ?s2)
         (poss_equiv ?a ?s1 ?s2))))
```

6.5.3 partial_state

An activity is a partially state constrained activity if and only if there exist occurrence-preserving fluent permutations.

```
(forall (?a) (iff (partial_state ?a)
(and      (exists (?s1)
               (forall (?s2)
                    (implies  (state_equiv ?s1 ?s2)
                           (poss_equiv ?a ?s1 ?s2))))
          (exists (?s3 ?s4)
               (and      (state_equiv ?s3 ?s4)
                    (not (poss_equiv ?a ?s3 ?s4)))))))
```

6.5.4 rigid_state

An activity is a rigid state activity if and only if the only occurrence-preserving fluent permutation is the trivial one.

```
(forall (?a) (iff (rigid_state ?a)
(forall (?s1)
          (exists (?s2)
               (and      (state_equiv ?s1 ?s2)
                    (not (poss_equiv ?a ?s1 ?s2)))))))
```

6.6 Grammar for relations of State-based preconditions for activities

The following grammar sentences describe process descriptions and auxiliary rules specified in KIF for State-based preconditions for activities.

Note: The function and importance of grammar sentences in ISO 18629 is explained in ISO18629-1 : 2004, 3.3.8, 4.2.4, and 5.1.

```
< simple_state_precond > ::=    (forall (?s)
                                     (implies (and     (occurrence ?s )
                                                       (legal ?s))
                                             < simple_state_axiom >))) |
                                (forall (?s)
                                     (implies  (and    (legal ?s)
                                                        < simple_state_axiom >)
                                                 (legal (successor < term > ?s))))
< state_precond > ::=           (forall (< variable >*)
                                        (implies  (and   (occurrence < variable > < term >)
                                                        (legal < term >))
                                                 < state_axiom >))) |
                                (forall (< variable >*)
                                        (implies  (and   (legal < term >)
                                                        < state_axiom >)
                                                 (legal (successor < term > < term >))))
< simple_state_literal > ::=    (prior < term  > ?s)
< simple_state_formula > ::=    < simple_state_literal > |
                                (not < simple_state_formula >) |
                                ({and | or} < simple_state_formula >*) |
                                ({implies | iff} < simple_state_formula >)
< simple_state_axiom > ::=      ({forall | exists} < variable >*)
                                        < simple_state_formula >)
< state_literal > ::=           (prior < term > < variable >)
< state_formula > ::=           < state_literal > |
                                (not < state_formula >) |
                                ({and | or} < state_formula >) |
                                ({implies | iff} < state_formula >*)
< state_axiom > ::=             (forall (< variable >+) < state_formula >)
```

7 Time-based preconditions for activities

This clause characterizes all definitions pertaining to Time-based preconditions for activities. The criterion used to classify these activities is whether or not legal occurrences of an activity depend only on the time at which the activities occur.

7.1 Primitive lexicon of Time-based preconditions for activities

No primitive relations are required by the lexicon of Time-based preconditions for activities.

7.2 Defined lexicon for concepts of Time-based preconditions for activities

The following relations are defined in this clause:

— (begin_equiv ?s1 ?s2);

— (time_precond ?a);

- (partial_time ?a);
- (rigid_time ?a).

Each concept is described by informal semantics and a KIF axiom.

7.3 Theories required by Time-based preconditions for activities

This theory requires :

- disc_state.th;
- occtree.th;
- psl_core.th.

7.4 Definitional extensions required by Time-based preconditions for activities

The following definitional extensions are required by Time-based preconditions for activities:

- precond.def;
- start.def.

7.5 Definitions of Time-based preconditions for activities

The following concepts are defined for Time-based preconditions for activities.

7.5.1 begin_equiv

Two atomic activity occurrences are begin-equivalent if and only if they have equal beginof timepoints.

(forall (?s1 ?s2) (iff (begin_equiv ?s1 ?s2)

(= (beginof ?s1) (beginof ?s2))))

7.5.2 time_precond

An activity is a time precondition activity if whenever two atomic occurrences in the occurrence tree agree on their beginof timepoints, then they agree on the extension of poss for the activity.

(forall (?a) (iff (time_precond ?a)

(forall (?s1 ?s2)

(implies (begin_equiv ?s1 ?s2))

(poss_equiv ?a ?s1 ?s2)))))

7.5.3 partial_time_precond

An activity is a partially temporally constrained activity if and only if there exist occurrence-preserving permutations

```
(forall (?a) (iff (partial_time ?a)
(and    (exists (?s1)
            (forall (?s2)
                  (implies  (begin_equiv ?s1 ?s2)
                          (poss_equiv ?a ?s1 ?s2))))
      (exists (?s3 ?s4)
            (and    (begin_equiv ?s3 ?s4)
                  (not (poss_equiv ?a ?s3 ?s4)))))))
```

7.5.4 rigid_time

An activity is a rigid time activity if and only if the only occurrence-preserving timeline permutation is the trivial one.

```
(forall (?a) (iff (rigid_time ?a)
(forall (?s1)
      (exists (?s2)
            (and    (begin_equiv ?s1 ?s2)
                  (not (poss_equiv ?a ?s1 ?s2)))))))
```

7.6 Grammar for process descriptions of Time-based preconditions for activities

The following grammar sentences describe process descriptions and auxiliary rules specified in KIF for Time-based preconditions for activities.

```
< simple_time_precond > ::=    (forall (?s)
                                   (implies  (and    (occurrence ?s < term >)
                                                     (legal ?s))
                                             < simple_time_axiom >))) |
                               (forall (?s)
                                   (implies  (and    (legal ?s)
                                                     < simple_time_axiom >)
                                             (legal (successor < term > ?s))))
< time_precond > ::=           (forall (?s)
                                   (implies  (and    (occurrence ?s < term >)
                                                     (legal ?s))
                                             < time_axiom >))) |
                               (forall (?s)
                                   (implies  (and    (legal ?s)
                                                     < time_axiom >)
```

```
                                        (legal (successor < term > ?s))))
< simple_time_literal > ::=      (= (beginof ?s) < term >)
< simple_time_formula > ::=      (and < simple_time_literal >*)
< simple_time_axiom > ::=        ({forall | exists} < variable >*) < simple_time_formula >)
< ordered_time_literal > ::=     < simple_time_literal > |
                                 (before (beginof ?s) < term >) |
                                 (before < term > (beginof ?s))
< time_formula > ::=             < ordered_time_literal > |
                                 (not < time_formula >) |
                                 ({and | or} < time_formula >*) |
                                 ({implies | iff} < time_formula >)
< time_axiom > ::=               ({forall | exists} < variable >*) < time_formula >)
```

8 Preconditions based on state and time

This clause characterizes all definitions pertaining to Preconditions based on state and time. The criterion used to classify these activities is whether or not legal occurrences of an activity depend on the state prior to the occurrences and the times at which the activities occur.

8.1 Primitive lexicon of Preconditions based on state and time

No primitive relations are required by the lexicon of Preconditions based on state and time.

8.2 Defined lexicon of Preconditions based on state and time

The following relations are defined in this clause:

- (mixed_precond ?s);
- (partial_mixed ?s);
- (rigid_mixed ?s).

Each concept is described by informal semantics and a KIF axiom.

8.3 Theories required by Preconditions based on state and time

This theory requires:

- disc_state.th;
- occtree.th;
- psl_core.th.

8.4 Definitional extensions required by Preconditions based on state and time

No Definitional Extensions are required by Preconditions based on state and time.

8.5 Definitions of Preconditions based on state and time

The following concepts are defined for Preconditions based on state and time.

8.5.1 mixed_precond

An activity is a mixed preconditional activity iff whenever two occurrences agree on state and their beginof timepoints, then they agree on the extension of poss for the activity.

```
(forall (?a) (iff (mixed_precond ?a)
(forall (?s1 ?s2)
    (implies  (and       (state_equiv ?s1 ?s2)
                         (begin_equiv ?s1 ?s2))
         (poss_equiv ?a ?s1 ?s2)))))
```

8.5.2 partial_mixed

An activity is a partially mixed constrained activity if and only if there exist occurrence-preserving permutations.

```
(forall (?a) (iff (partial_mixed ?a)
(and    (exists (?s1)
            (forall (?s2)
                (implies  (and     (state_equiv ?s1 ?s2)
                                         (begin_equiv ?s1 ?s2))
                     (poss_equiv ?a ?s1 ?s2))))
    (exists (?s3 ?s4)
        (and    (state_equiv ?s3 ?s4)
                    (begin_equiv ?s3 ?s4)
            (not (poss_equiv ?a ?s3 ?s4)))))))
```

8.5.3 rigid_mixed

An activity is a rigid mixed activity if and only if the only occurrence-preserving permutation that is both a fluent permutation and a timeline permutation, is the trivial one.

```
(forall (?a) (iff (rigid_mixed ?a)
(forall (?s1)
    (exists (?s2)
        (and    (state_equiv ?s1 ?s2)
                    (begin_equiv ?s1 ?s2)
            (not (poss_equiv ?a ?s1 ?s2)))))))
```

8.6 Grammar for Preconditions based on state and time

The following grammar sentences describe process descriptions and auxiliary rules specified in KIF for Preconditions based on state and time.

```
< simple_mix_precond > ::=    (forall (?s)
                                   (implies  (and   (occurrence ?s < term >)
                                                    (legal ?s))
                                              < simple_mix_axiom >))) |
                      (forall (?s)
                          (implies  (and  (legal ?s)
                                 < simple_mix_axiom >)
                             (legal (successor < term > ?s))))
< mix_precond > ::=           (forall (?s)
                                   (implies  (and   (occurrence ?s < term >)
                                                    (legal ?s))
                                              < mix_formula >)))
< simple_mix_literal > ::=    (= (beginof ?s) < term >) (prior  ?s)
< simple_mix_formula > ::=    (and < simple_mix_literal >*)
< simple_mix_axiom > ::=      ({forall | exists} < variable >*) < simple_mix_formula >)
< ordered_mix_literal > ::=   < simple_mix_literal > |
                              (before (beginof ?s) < term >) (prior < term > ?s) |
                              (before < term > (beginof ?s)) (prior < term > ?s)
< mix_formula > ::=           < ordered_mix_literal > |
                              (not < mix_formula >) |
                              ({and | or} < mix_formula >*) |
                              ({implies | iff} < mix_formula >)
< mix_axiom > ::=             ({forall | exists} < variable >*) < mix_formula >)
```

9 Occurrence-based preconditions for activities

This clause characterizes all definitions pertaining to Occurrence-based preconditions for activities. The criterion used to classify these activities is whether or not legal occurrences of an activity depend on the occurrences of other activities.

9.1 Primitive lexicon of Occurrence-based preconditions for activities

No primitive relations are introduced in the lexicon of Occurrence-based preconditions for activities.

9.2 Defined relations of Occurrence-based preconditions for activities

The following relations are defined in this clause:

— (tree_equiv ?s1 ?s2) ;

— (occurrence_constrained ?a);

- (occurrence_dependent ?a);
- (occurrence_independent ?a).

Each concept is described by informal semantics and a KIF axiom.

9.3 Theories required by Occurrence-based preconditions for activities

This theory requires:

- occtree.th;
- psl_core.th.

9.4 Definitional extensions required by Occurrence-based preconditions for activities

The following Definitional Extensions are required by the Occurrence-based preconditions for activities:

- precond.def.

9.5 Definitions of Occurrence-based preconditions for activities

The following concepts are defined for Occurrence-based preconditions for activities.

9.5.1 tree_equiv

Two activity occurrences are tree equivalent if only if they are mapped to each other by some transformation of the occurrence tree.

```
(forall (?a ?s1 ?s2)
(iff      (tree_equiv (successor ?a ?s1) (successor ?a ?s2))
          (and      (legal_equiv ?s1 ?s2)
                    (or       (tree_equiv ?s1 ?s2)
                              (initial ?s1)
                              (initial ?s2)))))
```

9.5.2 occurrence_constrained

An activity is occurrence constrained if and only if all transformations of the occurrence tree preserve legal occurrences of the activity.

```
(forall (?a) (iff (occurrence_constrained ?a)
(forall (?s1 ?s2)
          (implies  (and    (occurrence ?s1 ?a)
                            (occurrence ?s2 ?a)
                            (tree_equiv ?s1 ?s2))
                   (legal_equiv ?s1 ?s2)))))
```

9.5.3 occurrence_dependent

An activity is occurrence dependent if and only if there exist transformation of the occurrence tree that preserve legal occurrences of the activity.

```
(forall (?a) (iff (occurrence_dependent ?a)
(exists (?s1)
        (and    (occurrence ?s1 ?a)
                (forall (?s2)
                        (implies  (and    (occurrence ?s2 ?a)
                                          (tree_equiv ?s1 ?s2))
                                 (legal_equiv ?s1 ?s2)))))))
```

9.5.4 occurrence_independent

An activity is occurrence independent if and only if there is no nontrivial transformations of the occurrence tree that preserves legal occurrences of the activity.

```
(forall (?a) (iff (occurrence_independent ?a)
(forall (?s1)
        (implies  (occurrence ?s1 ?a)
                  (exists (?s2)
                        (and    (occurrence ?s2 ?a)
                                (tree_equiv ?s1 ?s2)
                                (not (legal_equiv ?s1 ?s2))))))))
```

9.6 Grammar for Occurrence-based preconditions for activities

The following grammar sentences describe process descriptions and auxiliary rules specified in KIF for Occurrence-based preconditions for activities.

```
< occ_constrained_precond > ::=
(forall (?s)
     ({implies | iff} (and   (occurrence ?s < term >)
                    (legal ?s)
                                (ubiquitous < term > < term >))
                {< leaf_constrained_axiom > | < inner_constrained_axiom})
< leaf_constrained_formula > ::=  (exists (?occ ?s1)
                                        (and    (occurrence ?occ < term >)
                                                (leaf_occ ?s1 ?occ)
                                                (= ?s (successor < term > ?s1))))
< leaf_constrained_axiom > ::= < leaf_constrained_formula > |
                                (not < leaf_constrained_formula >)
< inner_constrained_formula > ::=  (exists (?occ ?s1 ?s2)
                                      (and     (occurrence ?occ < term >)
                                               (subactivity_occurrence ?s1 ?occ)
                                               (subactivity_occurrence ?s2 ?occ)
                                               (precedes ?s1 ?s)
                                               (precedes ?s ?s2)))
< inner_constrained_axiom > ::=          < inner_constrained_formula > |
                                (not < inner_constrained_formula >)
```

10 Preventable conditions for activities

This clause characterizes all definitions pertaining to Preventable conditions for activities. The criterion used to classify these activities is whether or not legal occurrences of an activity depend on the state prior to the occurrences as well as occurrences of other activities.

10.1 Primitive lexicon of Preventable conditions for activities

No primitive relations are required by the lexicon of Preventable conditions for activities.

10.2 Defined relations of Preventable conditions for activities

The following relations are defined in this clause:

— (preventable ?a);

— (possibly_preventable ?a);

— (unpreventable ?a).

Each concept is described by informal semantics and a KIF axiom.

10.3 Theories required by Preventable conditions for activities

This theory requires:

— occtree.th;

— psl_core.th.

10.4 Definitional extensions required by Preventable conditions for activities

This extension requires:

— occ_precond.def;

— state_precond.def;

— precond.def.

10.5 Definitions of Preventable conditions for activities

The following concepts are defined for Preventable conditions for activities.

10.5.1 preventable

An activity is preventable if and only if legal occurrences depend on the fluents that hold at other activity occurrences in the tree.

```
(forall (?a) (iff (preventable ?a)
(forall (?s1 ?s2)
          (implies  (and    (occurrence ?s1 ?a)
                            (occurrence ?s2 ?a)
                            (state_equiv ?s1 ?s2)
                            (tree_equiv ?s1 ?s2))
                    (legal_equiv ?s1 ?s2)))))
```

10.5.2 possibly_preventable

An activity is possibly preventable if and only if legal occurrences depend on the fluents that hold at other activity occurrences in the tree.

```
(forall (?a) (iff (possibly_preventable ?a)
(exists (?s1)
          (and      (occurrence ?s1 ?a)
                    (forall (?s2)
                            (implies  (and  (occurrence ?s2 ?a)
                                            (state_equiv ?s1 ?s2)
                                            (tree_equiv ?s1 ?s2))
                                      (legal_equiv ?s1 ?s2)))))))
```

10.5.3 unpreventable

An activity is unpreventable if and only if there is no relationship between legal occurrences and the fluents that hold at other activity occurrences.

```
(forall (?a) (iff (unpreventable ?a)
(forall (?s1)
          (implies  (occurrence ?s1 ?a)
                    (exists (?s2)
                            (and      (occurrence ?s2 ?a)
                                      (state_equiv ?s1 ?s2)
                                      (tree_equiv ?s1 ?s2)
                                      (not (legal_equiv ?s1 ?s2))))))))
```

10.6 Grammar for process descriptions of Preventable conditions for activities

The following grammar sentences describe process descriptions specified in KIF for Preventable conditions for activities

```
< preventable_precond > ::=
(forall (?s)
      ({implies | iff} (and   (occurrence ?s < term >)
                        (legal ?s)
                                     (ubiquitous < term > < term >)
                                     < simple_state_axiom >)
{< leaf_constrained_axiom > | < inner_constrained_axiom}))
< possibly_preventable_precond > ::=
(forall (?s)
      ({implies | iff} (and   (occurrence ?s < term >)
                        (legal ?s)
                                     (ubiquitous < term > < term >)
                                     < state_axiom >)
{< leaf_constrained_axiom > | < inner_constrained_axiom}))
```

11 Periodic preconditions for activities

This clause characterizes all definitions pertaining to Periodic preconditions for activities. The criterion used to classify these activities is whether or not legal occurrences of an activity depend on the time at which the activity occurs as well as occurrences of other activities.

11.1 Primitive lexicon of Periodic preconditions for activities

No primitive relations are required by the lexicon of Periodic preconditions for activities.

11.2 Defined lexicon of Periodic preconditions for activities

The following relations are defined in this clause:

— (periodic ?a);

— (intermittent ?a);

— (aperiodic ?a).

Each concept is described by informal semantics and a KIF axiom.

11.3 Theories required by Periodic preconditions for activities

This theory requires:

- occtree.th;
- psl_core.th.

11.4 Definitional extensions required by Periodic preconditions for activities

This extension requires:

- occ_precond.def;
- time_precond.def;
- precond.def.

11.5 Definitions of Periodic preconditions for activities

The following concepts are defined for Periodic preconditions for activities.

11.5.1 periodic

An activity is periodic if and only if legal occurrences depend on the timepoints at which other activities occur.

```
(forall (?a) (iff (periodic ?a)
(forall (?s1 ?s2)
        (implies  (and   (occurrence ?s1 ?a)
                         (occurrence ?s2 ?a)
                         (begin_equiv ?s1 ?s2)
                         (tree_equiv ?s1 ?s2))
                 (legal_equiv ?s1 ?s2)))))
```

11.5.2 intermittent

An activity is intermittent if and only if legal occurrences depend on the timepoints at which other activities occur.

```
(forall (?a) (iff (intermittent ?a)
(exists (?s1)
        (and     (occurrence ?s1 ?a)
                 (forall (?s2)
                         (implies  (and   (occurrence ?s2 ?a)
                                          (begin_equiv ?s1 ?s2)
                                          (tree_equiv ?s1 ?s2))
                                  (legal_equiv ?s1 ?s2)))))))
```

11.5.3 aperiodic

An activity is aperiodic if and only if there is no relationship between legal occurrences and the timepoints at which other activities occur.

```
(forall (?a) (iff (aperiodic ?a)
(forall (?s1)
        (implies  (occurrence ?s1 ?a)
                 (exists (?s2)
                        (and    (occurrence ?s2 ?a)
                                (begin_equiv ?s1 ?s2)
                                (tree_equiv ?s1 ?s2)
                                (not (legal_equiv ?s1 ?s2))))))))
```

11.6 Grammar for of Periodic preconditions for activities

The following grammar sentences describe process descriptions specified in KIF for Periodic preconditions for activities.

```
< periodic_precond > ::=
(forall (?s)
     ({implies | iff} (and   (occurrence ?s < term >)
                    (legal ?s)
                                (ubiquitous < term > < term >)
                                < simple_time_axiom >)
                {< leaf_constrained_axiom > | < inner_constrained_axiom}))
< intermittent_precond > ::=
(forall (?s)
     ({implies | iff} (and   (occurrence ?s < term >)
                    (legal ?s)
                                (ubiquitous < term > < term >)
                                < time_axiom >)
                {< leaf_constrained_axiom > | < inner_constrained_axiom}))
```

12 Spoilage preconditions for activities

This clause characterizes all definitions pertaining to Spoilage preconditions for activities The criterion used to classify these activities is whether or not legal occurrences of an activity depend on the state prior to the occurrences, the times at which the activity occurs, and occurrences of other activities.

12.1 Primitive lexicon of Spoilage preconditions for activities

No primitive relations are required by the lexicon of Spoilage preconditions for activities.

12.2 Defined lexicon of Spoilage preconditions for activities

The following relations are defined in this clause:

- (spoilage ?a);
- (possible_spoilage ?a) ;
- (nonspoilage ?a).

Each concept is described by informal semantics and a KIF axiom.

12.3 Theories required by Spoilage preconditions for activities

This theory requires:

- occtree.th;
- psl_core.th.

12.4 Definitional extensions required by Spoilage preconditions for activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- time_precond.def;
- precond.def.

12.5 Definitions of Spoilage preconditions for activities

The following concepts are defined for Spoilage preconditions for activities.

12.5.1 spoilage

An activity has spoilage preconditions if

```
(forall (?a) (iff (spoilage ?a)
(forall (?s1 ?s2)
     (implies  (and    (occurrence ?s1 ?a)
                       (occurrence ?s2 ?a)
                       (state_equiv ?s1 ?s2)
                       (begin_equiv ?s1 ?s2)
                       (tree_equiv ?s1 ?s2))
                  (legal_equiv ?s1 ?s2))))
```

12.5.2 possible_spoilage

An activity has possible spoilage preconditions if

```
(forall (?a) (iff (possible_spoilage ?a)
(exists (?s1)
          (and      (occurrence ?s1 ?a)
                    (forall (?s2)
                              (implies  (and    (occurrence ?s2 ?a)
                                                (state_equiv ?s1 ?s2)
                                                (begin_equiv ?s1 ?s2)
                                                (tree_equiv ?s1 ?s2))
                                        (legal_equiv ?s1 ?s2)))))))
```

12.5.3 nonspoilage

An activity has nonspoilage precondiitions if

```
(forall (?a) (iff (nonspoilage ?a)
(forall (?s1)
          (implies  (occurrence ?s1 ?a)
                    (exists (?s2)
                              (and      (occurrence ?s2 ?a)
                                        (state_equiv ?s1 ?s2)
                                        (begin_equiv ?s1 ?s2)
                                        (tree_equiv ?s1 ?s2)
                                        (not (legal_equiv ?s1 ?s2))))))))
```

12.6 Grammar of Spoilage preconditions for activities

The grammar for process descriptions of Spoilage preconditions for activities is equivalent to the grammar of a general process description sentence specified in ISO 18629-11.

13 Effects of activities

This clause characterizes all definitions pertaining to Effects of activities. The criterion used to classify these activities is whether or the effects of an activity are the same for all occurrences of the activity.

13.1 Primitive lexicon of Effects of activities

No primitive relations are required by the lexicon of Effects of activities

13.2 Defined lexicon of Effects of activities

The following relations are defined in this clause:

— (effects_equiv ?a ?s1 ?s2);

- (context_free ?a);
- (null ?a).

Each concept is described by informal semantics and a KIF axiom.

13.3 Theories required by Effects of activities

This theory requires:

- occtree.th;
- psl_core.th.

13.4 Definitional extensions required by Effects of activities

The following Definitional Extensions are required by Effects of activities:

13.5 Definitions of Effects of activities

The following concepts are defined for Spectrum and Subtree Containement.

13.5.1 effects_equiv

Two occurrences of an atomic activity are effects equivalent iff the same states hold after the occurrences.

```
(forall (?a ?s1 ?s2) (iff (effects_equiv ?a ?s1 ?s2)
(and    (occurrence ?s1 ?a)
        (occurrence ?s2 ?a)
        (forall (?f)
                (iff    (holds ?f ?s1)
                        (holds ?f ?s2))))))
```

13.5.2 context_free

An activity is context-free iff any fluent changed by one occurrence of the activity is changed by all occurrences of the activity.

```
(forall (?a) (iff (context_free ?a)
(forall (?s1 ?s2)
        (implies  (and  (occurrence ?s1 ?a)
                        (occurrence ?s2 ?a))
                  (effects_equiv ?a ?s1 ?s2)))))
```

13.5.3 null

An activity is null if and only if it does not achieve or falsify any fluent.

(forall (?a) (iff (null ?a)

```
(forall (?s)
        (implies  (occurrence ?s ?a)
                  (forall (?f)
                       (iff    (holds ?f ?s)
                               (prior ?f ?s)))))))
```

13.6 Grammar for Effects of activities

The following grammar sentences describe process descriptions specified in KIF for Effects of activities:

```
< context_free_effect > ::=     (forall (?s)
                               (implies (occurrence ?s )
                                       < simple_holds_axiom >))
< null_effect > ::=            (forall (?s)
                               (implies (occurrence ?s )
                                       (iff   < simple_holds_axiom >
                                              < simple_state_axiom >)))
< simple_holds_literal > ::=    (holds < term  > ?s)
< simple_holds_formula > ::=    < simple_holds_literal > |
                         (not < simple_holds_formula >) |
                         ({and | or} < simple_holds_formula >*) |
                         ({implies | iff} < simple_holds_formula >)
< simple_holds_axiom > ::=      ({forall | exists} < variable >*)
                               < simple_holds_formula >)
```

14 State-based effects of activities

This clause characterizes all definitions pertaining to State-based effects of activities. The criterion used to classify these activities is whether or not the effects of an activity depend only on the state prior to the occurrences.

14.1 Primitive lexicon of State-based effects of activities

No primitive relations are required by the lexicon of State-based effects of activities.

14.2 Defined lexicon of State-based effects of activities

The following relations are defined in this clause:

- (markov_effects ?a);

- (partial_state_effects ?a);

- (rigid_state_effects ?a).

Each concept is described by informal semantics and a KIF axiom.

14.3 Core theories required by State-based effects of activities

This definitional extension requires:

- occtree.th;
- psl_core.th.

14.4 Definitional extensions required by State-based effects of activities

The following Definitional Extensions are required by State-based effects of activities:

- precond.def.

14.5 Definitions of State-based effects of activities

The following concepts are defined for State-based effects of activities

14.5.1 markov_effects

An activity is a Markovian effect activity iff whenever any two occurrences of the activity agree on state, then they agree on the effects (i.e. the states that hold after the occurrence.

```
(forall (?a) (iff (markov_effects ?a)
(forall (?s1 ?s2)
    (implies  (and  (occurrence ?s1 ?a)
                    (occurrence ?s2 ?a)
                    (state_equiv ?s1 ?s2))
             (effects_equiv ?a ?s1 ?s2)))))
```

14.5.2 partial_state_effects

An activity is a partially state constrained activity if and only if there exist effect-preserving fluent permutations.

```
(forall (?a) (iff (partial_state_effects ?a)
(and   (exists (?s1)
        (forall (?s2)
            (implies  (state_equiv ?s1 ?s2)
                 (effects_equiv ?a ?s1 ?s2))))
    (exists (?s3 ?s4)
        (and    (state_equiv ?s3 ?s4)
            (not (effects_equiv ?a ?s3 ?s4)))))))
```

14.5.3 rigid_state_effects

An activity is a rigid state activity if and only if the only effects-preserving fluent permutation is the trivial one.

```
(forall (?a) (iff (rigid_state_effects ?a)
(forall (?s1)
      (exists (?s2)
            (and    (state_equiv ?s1 ?s2)
                  (not (effects_equiv ?a ?s1 ?s2)))))))
```

14.6 Grammar for State-based Effects Activities.

The grammar for process descriptions of Embedding Constraints for Activities :

```
< simple_state_effect > ::=     (forall (?s)
                               (implies (and   (occurrence ?s )
                                               < simple_state_axiom >)
                                          < simple_holds_axiom >)))
< state_effect > ::=           (forall (< variable >*)
                               (implies  (and  (occurrence < variable > < term >)
                                               < state_axiom >)
                                          < simple_holds_axiom >)))
```

15 Time-based effects of activities

This clause characterizes all definitions pertaining to Time-based effects of activities. The criterion used to classify these activities is whether or not the effects of an activity depend only on the times at which the activity occurs.

15.1 Primitive lexicon of Time-based effects of activities

No primitive relations are required by the lexicon of Time-based effects of activities.

15.2 Defined lexicon of Time-based effects of activities

The following relations are defined in this clause:

— (temporal_effects ?a);

— (partial_temporal ?a);

— (nontemporal ?a).

Each concept is described by informal semantics and a KIF axiom.

15.3 Core theories required by Time-based effects of activities

This definitional extension requires:

— occtree.th;

— psl_core.th.

15.4 Definitional extensions required by Time-based effects of activities

This extension requires Time-based effects of activities.

15.5 Definitions of Time-based effects of activities

The following concepts are defined for Time-based effects of activities.

15.5.1 temporal_effects

An activity is a temporal effect activity iff whenever any two occurrences of the activity agree on their beginof timepoints, then they agree on the effects (i.e. the states that hold after the occurrence).

```
(forall (?a) (iff (temporal_effects ?a)
(forall (?s1 ?s2)
    (implies  (and  (occurrence ?s1 ?a)
                    (occurrence ?s2 ?a)
                    (begin_equiv ?s1 ?s2))
               (effects_equiv ?a ?s1 ?s2)))))
```

15.5.2 partial_temporal

An activity is a partially time effects activity if and only if there exist effect-preserving timeline permutations.

```
(forall (?a) (iff (partial_temporal ?a)
(and    (exists (?s1)
            (forall (?s2)
                (implies  (begin_equiv ?s1 ?s2)
                          (effects_equiv ?a ?s1 ?s2))))
    (exists (?s3 ?s4)
            (and    (begin_equiv ?s3 ?s4)
                (not (effects_equiv ?a ?s3 ?s4)))))))
```

15.5.3 nontemporal

An activity is a rigid time effects activity if and only if the only effects-preserving timeline permutation is the trivial one.

```
(forall (?a) (iff (nontemporal ?a)
(forall (?s1)
        (exists (?s2)
                (and    (begin_equiv ?s1 ?s2)
                        (not (effects_equiv ?a ?s1 ?s2)))))))
```

15.6 Grammar for Time-based effects of activities

The grammar for process descriptions of Time-based effects of activities:

```
< simple_time_effects > ::=     (forall (?s)
                                (implies  (and  (occurrence ?s < term >)
                                                (legal ?s))
                                          < simple_holds_axiom >)))
< time_effects > ::=            (forall (?s)
                                (implies  (and  (occurrence ?s < term >
                                                (legal ?s)
                                                < time_axiom >)
                                          < simple_holds_axiom >))
```

16 Occurrence-based effects of activities

This clause characterizes all definitions pertaining to Occurrence-based effects of activities. The criterion used to classify these activities is whether or not the effects of an activity depend only on the occurrences of other activities.

16.1 Primitive lexicon of Occurrence-based effects of activities

No primitive relations are required by the lexicon of Occurrence-based effects of activities.

16.2 Defined lexicon of Occurrence-based effects of activities

The following relations are defined in this clause:

— (occ_effects ?a ?s);

— (occ_depend_effects ?a ?s);

— (nonocc_effects ?a).

Each concept is described by informal semantics and a KIF axiom.

16.3 Core theories required by Occurrence-based effects of activities

This definitional extension requires:

— occtree.th;

— psl_core.th.

16.4 Definitional extensions required by Occurrence-based effects of activities

No Definitional Extensions are required by Occurrence-based effects of activities.

16.5 Definitions of Occurrence-based effects of activities

The following concepts are defined for Occurrence-based effects of activities.

16.5.1 occ_effects

An activity has occurrence constrained effects if and only if the effects of the activity are preserved by transformations of the occurrence tree.

```
(forall (?a) (iff (occ_effects ?a)
(forall (?s1 ?s2)
    (implies  (and  (occurrence ?s1 ?a)
                    (occurrence ?s2 ?a)
                    (tree_equiv ?s1 ?s2))
             (effects_equiv ?a ?s1 ?s2)))))
```

16.5.2 occ_depend_effects

An activity has occurrence dependent effects if and only if there exist transformations of the occurrence tree that preserve the effects of the activity.

```
(forall (?a) (iff (occ_depend_effects ?a)
(exists (?s1)
    (and   (occurrence ?s1 ?a)
        (forall (?s2)
            (implies  (and  (occurrence ?s2 ?a)
                            (tree_equiv ?s1 ?s2))
                          (effects_equiv ?a ?s1 ?s2)))))))
```

16.5.3 nonocc_effects

An activity has no occurrence dependent effects if and only if the only transformation of the occurrence tree that preserve the effects of the activity is the trivial one.

```
(forall (?a) (iff (nonocc_effects ?a)
(forall (?s1)
    (implies  (occurrence ?s1 ?a)
             (exists (?s2)
                (and   (occurrence ?s2 ?a)
                    (tree_equiv ?s1 ?s2)
                    (not (effects_equiv ?a ?s1 ?s2))))))))
```

16.6 Grammar for Occurrence-based effects of activities

The grammar for the process descriptions of Occurrence-based effects of activities:

```
< occ_effect_axiom > ::=
(forall (?s)
    (implies {< leaf_constrained_axiom > | < inner_constrained_axiom}
        < simple_holds_axiom >))
< occ_depend_effect_axiom > ::=
(forall (?s)
    (implies {< leaf_constrained_axiom > | < inner_constrained_axiom}
        < holds_axiom >))
```

17 Effects of activities: Occurrence Constraints

This clause characterizes all definitions pertaining to Effects of activities: Occurrence Constraints. The criterion used to classify these activities is whether or not the effects of an activity depend only on the state prior to occurrences of other activities.

17.1 Primitive lexicon of Effects of activities: Occurrence Constraints

No primitive relations are required by the lexicon of Effects of activities: Occurrence Constraints.

17.2 Defined lexicon of Effects of activities: Occurrence Constraints

The following relations are defined in this clause:

- (quantum ?a);
- (semiclassical ?a);
- (classical ?a).

Each concept is described by informal semantics and a KIF axiom.

17.3 Core theories required by Effects of activities: Occurrence Constraints

This definitional extensions requires:

- occtree.th;
- psl_core.th.

17.4 Definitional extensions required by Effects of activities: Occurrence Constraints

No definitional extension is required by this extension.

17.5 Definitions of Effects of activities: Occurrence Constraints

The following concepts are defined for Effects of activities: Occurrence Constraints.

17.5.1 quantum

An activity is a quantum activity if and only if its effects depend on the fluents that hold at other activity occurrences in the tree. In particular, effects are preserved by transformations of the occurrence tree that also preserve fluents.

```
(forall (?a) (iff (quantum ?a)
(forall (?s1 ?s2)
    (implies  (and  (occurrence ?s1 ?a)
            (occurrence ?s2 ?a)
            (state_equiv ?s1 ?s2)
            (tree_equiv ?s1 ?s2))
         (effects_equiv ?a ?s1 ?s2)))))
```

17.5.2 semiclassical

An activity is a semiclassical activity if and only if there exist transformations of the occurrence tree that also preserve fluents and the effects of the activity.

```
(forall (?a) (iff (semiclassical ?a)
(exists (?s1)
    (and    (occurrence ?s1 ?a)
        (forall (?s2)
            (implies  (and  (occurrence ?s2 ?a)
                    (state_equiv ?s1 ?s2)
                    (tree_equiv ?s1 ?s2))
                 (effects_equiv ?a ?s1 ?s2)))))))
```

17.5.3 classical

An activity is classical if and only if its effects are independent of the occurrences of any other activities in the occurrence tree.

```
(forall (?a) (iff (classical ?a)
(forall (?s1)
    (implies  (occurrence ?s1 ?a)
         (exists (?s2)
            (and    (occurrence ?s2 ?a)
                (state_equiv ?s1 ?s2)
                (tree_equiv ?s1 ?s2)
                (not (effects_equiv ?a ?s1 ?s2)))))))
```

17.6 Grammar for Effects of activities: Occurrence Constraints

The grammar for the process descriptions of Effects of activities: Occurrence:

```
< quantum_axiom > ::=
(forall (?s)
     (implies (and   {< leaf_constrained_axiom > | < inner_constrained_axiom}
                     < simple_state_axiom >)
               < simple_holds_axiom >))
< semiclassical_axiom > ::=
(forall (?s)
     (implies (and   {< leaf_constrained_axiom > | < inner_constrained_axiom}
                     < state_axiom >)
               < simple_holds_axiom >))
```

18 Effects of activities: Temporal and Occurrence Constraints

This clause characterizes all definitions pertaining to Effects of activities: Temporal and Occurrence Constraints. The criterion used to classify these activities is whether or not the effects of an activity depend only on the times that other activities occur.

18.1 Primitive lexicon of Effects of activities: Temporal and Occurrence Constraints

No primitive relations are required by the lexicon of Effects of activities: Temporal and Occurrence Constraints.

18.2 Defined lexicon of Effects of activities: Temporal and Occurrence Constraints

The following relations are defined in this clause:

- (relativistic ?a);
- (seminewton ?a);
- (newton ?a).

Each concept is described by informal semantics and a KIF axiom.

18.3 Core theories required by Effects of activities: Temporal and Occurrence Constraints

This definitional extension requires:

- occtree.th;
- psl_core.th.

18.4 Definitional extensions required by Effects of activities: Temporal and Occurrence Constraints

This extension requires Preconditions for Activities.

18.5 Definitions of Effects of activities: Temporal and Occurrence Constraints

The following concepts are defined for Effects of activities: Temporal and Occurrence Constraints.

18.5.1 relativistic

An activity is relativistic if and only if its effects depend on the timepoints at which other activities occur in the tree. In particular, effects are preserved by transformations of the occurrence tree that also preserve the timeline.

```
(forall (?a) (iff (relativistic ?a)
(forall (?s1 ?s2)
    (implies  (and  (occurrence ?s1 ?a)
                (occurrence ?s2 ?a)
                (begin_equiv ?s1 ?s2)
                (tree_equiv ?s1 ?s2))
             (effects_equiv ?a ?s1 ?s2)))))
```

18.5.2 seminewton

An activity is a seminewton activity if and only if there exist transformations of the occurrence tree that also preserve the timeline and the effects of the activity.

```
(forall (?a) (iff (seminewton ?a)
(exists (?s1)
    (and    (occurrence ?s1 ?a)
        (forall (?s2)
            (implies  (and  (occurrence ?s2 ?a)
                        (begin_equiv ?s1 ?s2)
                        (tree_equiv ?s1 ?s2))
                     (effects_equiv ?a ?s1 ?s2))))))
```

18.5.3 newtonian

An activity is newtonian if and only if its effects are independent of the beginof timepoints of occurrences of any other activities in the occurrence tree.

```
(forall (?a) (iff (newtonian ?a)
(forall (?s1)
    (implies  (occurrence ?s1 ?a)
         (exists (?s2)
            (and    (occurrence ?s2 ?a)
                        (begin_equiv ?s1 ?s2)
                        (tree_equiv ?s1 ?s2)
                        (not (effects_equiv ?a ?s1 ?s2)))))))
```

18.6 Grammar for Effects of activities: Temporal and Occurrence Constraints

The grammar for the process descriptions of Effects of activities: Temporal and Occurrence Constraints:

```
< relativistic_axiom > ::=
(forall (?s)
    (implies (and   {< leaf_constrained_axiom > | < inner_constrained_axiom}
                    < simple_time_axiom >)
                < simple_holds_axiom >))
< seminewton_axiom > ::=
(forall (?s)
    (implies (and   {< leaf_constrained_axiom > | < inner_constrained_axiom}
                    < time_axiom >)
                < simple_holds_axiom >))
```

19 Fluent trees

This clause characterizes all definitions pertaining to Fluent trees. The criterion used to classify these fluents depends on the activities that can either achieve or falsify a fluent.

19.1 Primitive lexicon of Fluent trees

No primitive relations are required by the lexicon of Fluent trees.

19.2 Defined relations of Fluent trees

The following relations are defined in this clause:

— (achieved ?f ?occ);

— (falsified ?f ?occ);

— (irreversible ?f) ;

— (unachievable ?f) ;

— (bounded ?f).

Each concept is described by informal semantics and a KIF axiom.

19.3 Core theories required by Fluent trees

This definitional extension requires:

- occtree.th;
- psl_core.th.

19.4 Definitional extensions required by Fluent trees

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

19.5 Definitions of Fluent trees

The following concepts are defined for Fluent trees.

19.5.1 achieved

A fluent ?f is achieved by an occurrence ?occ iff it does not hold before the occurrence, but it does hold after the occurrence.

```
(forall (?f ?occ) (iff (achieved ?f ?occ)
(and    (holds ?f ?occ)
        (not (prior ?f ?occ)))))
```

19.5.2 falsified

A fluent ?f is falsified by an occurrence ?occ iff it holds in before the occurrence but it does not hold after the occurrence.

```
(forall (?f ?occ) (iff (falsified ?f ?occ)
(and    (not (holds ?f ?occ))
        (prior ?f ?occ))))
```

19.5.3 irreversible

A fluent is irreversible if it is never falsified.

```
(forall (?f) (iff (irreversible ?f)
(not (exists (?occ)
        (falsified ?f ?occ)))))
```

19.5.4 unachievable

A fluent is unachievable if it is never achieved.

```
(forall (?f) (iff (unachievable ?f)
(not (exists (?occ)
        (achieved ?f ?occ)))))
```

19.5.5 bounded

```
(forall (?f) (iff (bounded ?f)
(exists (?occ1 ?occ2)
        (and    (achieved ?f ?occ1)
                (falsified ?f ?occ2)))))
```

19.6 Grammar for process descriptions of Fluent trees

The following grammar sentences describe process descriptions specified in KIF for Fluent trees

20 Distribution of complex activities

This clause characterizes all definitions pertaining to Distribution of complex activities. The criterion used to classify these activities is whether or not legal occurrences of a complex activity are constrained in some way.

20.1 Primitive lexicon of Distribution of complex activities

No primitive relations are required by the lexicon of Distribution of complex activities.

20.2 Defined relations of Distribution of complex activities

The following relations are defined in this clause:

- (profile ?occ ?a);
- (root_equiv ?a ?occ1 ?occ2);
- (universal ?a) ;
- (restricted ?a) ;
- (constrained ?a).

Each concept is described by informal semantics and a KIF axiom.

20.3 Core theories required by Distribution of complex activities

This definitional extension requires:

- occtree.th;
- psl_core.th.

20.4 Definitional extensions required by Distribution of complex activities

This extension requires:

- occ_precond.def;

- state_precond.def;
- precond.def.

20.5 Definitions of Distribution of complex activities

The following concepts are defined for Distribution of complex activities.

20.5.1 profile

The activity occurrence ?occ is an element of the profile for the activity ?a iff ?occ is the initial occurrence in a branch of the occurrence tree that is occurrence-isomorphic to a branch of an activity tree for ?a.

```
  (forall (?occ ?a) (iff (profile ?occ ?a)
(exists (?occ1 ?occ2 ?a1)
    (and    (occurrence_of ?occ1 ?a)
        (occurrence_of ?occ ?a1)
        (occurrence_of ?occ2 ?a1)
        (root_occ ?occ2 ?occ1)
        (forall (?occ3 ?a2)
            (implies  (and  (subactivity_occurrence ?occ3 ?occ1)
                    (occurrence_of ?occ3 ?a2))
                (exists (?occ4)
                    (and    (occurrence_of ?occ4 ?a2)
                        (precedes ?occ ?occ4)))))))))
```

20.5.2 root_equiv

Two activity occurrences ?occ1 and ?occ2 in the occurrence tree are root-equivalent with respect to an activity ?a if and only if there exist occurrences of ?a succeeding both ?occ1 and ?occ2.

```
(forall (?a ?occ1 ?occ2) (iff (root_equiv ?a ?occ1 ?occ2)
(implies  (and  (profile ?occ1 ?a)
        (profile ?occ2 ?a))
     (iff  (root ?occ1 ?a)
        (root ?occ2 ?a)))))
```

20.5.3 universal

An activity is universal if it occurs whenever it is possible.

```
(forall (?a) (iff (universal ?a)
(forall (?occ1 ?occ2)
    (implies  (and  (profile ?occ1 ?a)
            (profile ?occ2 ?a))
```

```
              (root_equiv ?a ?occ1 ?occ2)))))
```

20.5.4 local

An activity is constrained iff every activity has occurrences that are not root equivalent.

```
(forall (?a) (iff (local ?a)
(forall (?a1)
      (exists (?occ1 ?occ2)
            (and    (occurrence_of ?occ1 ?a1)
                  (occurrence_of ?occ2 ?a1)
                  (profile ?occ1 ?a)
                  (profile ?occ2 ?a)
                  (not (root_equiv ?a ?occ1 ?occ2)))))))
```

20.5.5 restricted

An activity is restricted iff there exist activities that are root equivalent.

```
(forall (?a) (iff (restricted ?a)
(and    (exists (?a1)
            (forall (?occ1 ?occ2)
                  (implies  (and    (occurrence_of ?occ1 ?a1)
                                (occurrence_of ?occ2 ?a1)
                                (profile ?occ1 ?a)
                                (profile ?occ2 ?a))
                         (root_equiv ?a ?occ1 ?occ2))))
      (exists (?a2 ?occ3 ?occ4)
            (and    (occurrence_of ?occ3 ?a2)
                  (occurrence_of ?occ4 ?a2)
                  (profile ?occ3 ?a)
                  (profile ?occ4 ?a)
                  (not (root_equiv ?a ?occ3 ?occ4)))))))
```

20.6 Grammar for process descriptions of Distribution of complex activities

The following grammar sentences describe process descriptions specified in KIF for Distribution of complex activities

```
< universal_axiom > ::= (forall (?s)
                        (implies  (legal ?s)
                               < distribution_formula>))
< restricted_axiom > ::= (forall (?s)
                        (implies  < successor_axiom >
                                  < distribution_formula>))
< distribution_formula > ::=    (exists (?occ)
                                (and    (occurrence ?occ ?a)
                                        (root_occ ?s ?occ)))
```

21 State-based distribution of complex activities

This clause characterizes all definitions pertaining to State-based distribution of complex activities. The criterion used to classify these activities is whether or not legal occurrences of a complex activity depend only on the state prior to the occurrences.

21.1 Primitive lexicon of State-based distribution of complex activities

No primitive relations are required by the lexicon of State-based distribution of complex activities.

21.2 Defined relations of State-based distribution of complex activities

The following relations are defined in this clause:

- (trigger ?a)
- (partial_trigger ?a)
- (nontrigger ?a)

Each concept is described by informal semantics and a KIF axiom.

21.3 Theories required by State-based distribution of complex activities

This theory requires:

- occtree.th;
- psl_core.th.

21.4 Definitional extensions required by State-based distribution of complex activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

21.5 Definitions of State-based distribution of complex activities

The following concepts are defined for State-based distribution of complex activities.

21.5.1 trigger

An activity is a trigger activity iff whenever two occurrences agree on state, then they agree on the occurrences of the activity.

```
(forall (?a) (iff (trigger ?a)
(forall (?occ1 ?occ2)
    (implies  (and  (profile ?occ1 ?a)
            (profile ?occ2 ?a)
            (state_equiv ?occ1 ?occ2))
         (root_equiv ?a ?occ1 ?occ2)))))
```

21.5.2 partial_trigger

An activity is a partial trigger activity iff there exist occurrences that agree on state and on the occurrence of the activity.

```
(forall (?a) (iff (partial_trigger ?a)
(and    (exists (?occ1)
        (forall (?occ2)
            (implies  (and  (profile ?occ1 ?a)
                    (profile ?occ2 ?a)
                    (state_equiv ?occ1 ?occ2))
                 (root_equiv ?a ?occ1 ?occ2))))
    (exists (?occ3 ?occ4)
        (and    (profile ?occ3 ?a)
            (profile ?occ4 ?a)
            (state_equiv ?occ3 ?occ4)
            (not (root_equiv ?a ?occ3 ?occ4)))))))
```

21.5.3 nontrigger

An activity is a nontrigger activity if and only if the only occurrence-preserving fluent permutation is the trivial one.

```
(forall (?a) (iff (nontrigger ?a)
(forall (?occ1)
    (implies  (profile ?occ1 ?a)
         (exists (?occ2)
            (and    (profile ?occ2 ?a)
                (state_equiv ?occ1 ?occ2)
                (not (root_equiv ?a ?occ1 ?occ2)))))))
```

21.6 Grammar for process descriptions of State-based distribution of complex activities

The following grammar sentences describe process descriptions specified in KIF for State-based distribution of complex activities

```
< trigger_activity > ::=      (forall (?s)
                             (implies  < simple_state_axiom >
                                       < distribution_formula>))
< partial_trigger > ::=       (forall (?s < variable >+)
                             (implies  < state_axiom >
                                       < distribution_formula>))
```

22 Time-based distribution of complex activities

This clause characterizes all definitions pertaining to Time-based distribution of complex activities. The criterion used to classify these activities is whether or not legal occurrences of a complex activity depend only on the times at which the activity occurs.

22.1 Primitive lexicon of Time-based distribution of complex activities

No primitive relations are required by the lexicon of Time-baseddistribution of complex activities.

22.2 Defined relations of Time-based distribution of complex activities

The following relations are defined in this clause:

- (launch ?a);
- (partial_launch ?a);
- (rigid_launch ?a).

Each concept is described by informal semantics and a KIF axiom.

22.3 Theories required by Time-based distribution of complex activities

This theory requires:

- occtree.th;
- psl_core.th.

22.4 Definitional extensions required by Time-based distribution of complex activities

This extension requires:

- occ_precond.def;
- state_precond.def;

— precond.def.

22.5 Definitions of Time-based distribution of complex activities

The following concepts are defined for Time-based distribution of complex activities.

22.5.1 launch

An activity is a launch activity if and only if whenever two occurrences agree on their beginof timepoints, then they agree on the occurrences of the activity.

```
(forall (?a) (iff (launch ?a)
(forall (?occ1 ?occ2)
(implies  (and  (profile ?occ1 ?a)
            (profile ?occ2 ?a)
                    (begin_equiv ?occ1 ?occ2))
               (root_equiv ?a ?occ1 ?occ2)))))
```

22.5.2 partial_launch

An activity is a partial launch activity if and only if there exist occurrences that agree on their beginof timepoints and on the occurrence of the activity.

```
(forall (?a) (iff (partial_launch ?a)
(and    (exists (?occ1)
             (forall (?occ2)
                    (implies  (and  (profile ?occ1 ?a)
                                  (profile ?occ2 ?a)
                                  (begin_equiv ?occ1 ?occ2))
                             (root_equiv ?a ?occ1 ?occ2))))
       (exists (?occ3 ?occ4)
             (and    (profile ?occ3 ?a)
                    (profile ?occ4 ?a)
                    (begin_equiv ?occ3 ?occ4))
                    (not (root_equiv ?a ?occ3 ?occ4))))))
```

22.5.3 rigid_launch

An activity is a rigid launch activity if and only if the only occurrence-preserving timeline permutation is the trivial one.

```
(forall (?a) (iff (rigid_launch ?a)
(forall (?occ1)
       (implies  (profile ?occ1 ?a)
               (exists (?occ2)
                    (and    (begin_equiv ?occ1 ?occ2)
                            (not (root_equiv ?a ?occ1 ?occ2)))))))
```

22.6 Grammar for process descriptions of Time-baseddistribution of complex activities

The following grammar sentences describe process descriptions specified in KIF for Time-baseddistribution of complex activities.

```
< simple_launch_activity > ::=    (forall (?s)
                            (implies  < simple_time_axiom >
                                      < distribution_formula>))
< partial_launch_activity > ::=  (forall (?s < variable >+)
                            (implies  < time_axiom >
                                      < distribution_formula>))
```

23 Mixed distribution of complex activities

This clause characterizes all definitions pertaining to Mixed distribution of complex activities. The criterion used to classify these activities is whether or not legal occurrences of a complex activity depend on the state prior to the occurrences and the times at which the activity occurs.

23.1 Primitive lexicon of Mixed distribution of complex activities

No primitive relations are required by the lexicon of Mixed distribution of complex activities.

23.2 Defined relations of Mixed distribution of complex activities

The following relations are defined in this clause:

- (conditional_launch ?a);
- (partial_conditional_launch ?a);
- (unconditional_launch ?a).

Each concept is described by informal semantics and a KIF axiom.

23.3 Core theories required by Mixed distribution of complex activities

This theory requires:

- occtree.th;
- psl_core.th.

23.4 Definitional extensions required by Mixed distribution of complex activities

This extension requires:

- occ_precond.def;

- state_precond.def;
- precond.def.

23.5 Definitions of Mixed distribution of complex activities

The following concepts are defined for Mixed distribution of complex activities.

23.5.1 conditional_launch

An activity is a conditional launch activity if and only if whenever two occurrences agree on state and their beginof timepoints, then they agree on the occurrences of the activity.

```
(forall (?a) (iff (conditional_launch ?a)
(forall (?occ1 ?occ2)
    (implies  (and  (profile ?occ1 ?a)
            (profile ?occ2 ?a)
            (state_equiv ?occ1 ?occ2)
            (begin_equiv ?occ1 ?occ2))
         (root_equiv ?a ?occ1 ?occ2)))))
```

23.5.2 partial_conditional_launch

An activity is a partial conditional launch activity if and only if there exist occurrences such that whenever they agree on state and their beginof timepoints, then they also agree on the occurrences of the activity.

```
(forall (?a) (iff (partial_conditional_launch ?a)
(and    (exists (?occ1)
        (forall (?occ2)
            (implies  (and  (profile ?occ1 ?a)
                    (profile ?occ2 ?a)
                    (state_equiv ?occ1 ?occ2)
                    (begin_equiv ?occ1 ?occ2))
                 (root_equiv ?a ?occ1 ?occ2))))
    (exists (?occ3 ?occ4)
        (and    (profile ?occ3 ?a)
            (profile ?occ4 ?a)
            (state_equiv ?occ3 ?occ4)
            (begin_equiv ?occ3 ?occ4)
            (not (root_equiv ?a ?occ3 ?occ4)))))))
```

23.5.3 unconditional_launch

An activity is a nontrigger activity if and only if the only occurrence-preserving permutation that also preserves both state and time is the trivial one.

```
(forall (?a) (iff (unconditional_launch ?a)
(forall (?occ1)
    (implies  (profile ?occ1 ?a)
         (exists (?occ2)
            (and    (profile ?occ2 ?a)
                (state_equiv ?occ1 ?occ2)
                (begin_equiv ?occ1 ?occ2)
                (not (root_equiv ?a ?occ1 ?occ2)))))))
```

23.6 Grammar for process descriptions of Mixed distribution of complex activities

The following grammar sentences describe process descriptions specified in KIF for Mixed distribution of complex activities

```
< conditional_launch_activitiy > ::=    (forall (?s ?s2)
                                (implies < simple_mix_axiom >
                                     < distribution_formula>))
< partial_conditional_launch > ::=  (forall (?s ?s2 < variable >+)
                                (implies < mix_formula >
                                     < distribution_formula>))
```

24 Variation of complex activities

This clause characterizes all definitions pertaining to Variation of complex activities. The criterion used to classify these activities is whether or not occurrences of the subactivities of a complex activity are constrained.

24.1 Primitive lexicon of Variation of complex activities

No primitive relations are required by the lexicon of Variation of complex activities.

24.2 Defined relations of Variation of complex activities

The following relations are defined in this clause:

— (min_equiv ?occ1 ?occ2 ?a);

— (uniform ?a);

— (variegated ?a);

— (multiform ?a).

Each concept is described by informal semantics and a KIF axiom.

24.3 Theories required by Variation of complex activities

This theory requires:

- occtree.th;
- psl_core.th.

24.4 Definitional extensions required by Variation of complex activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

24.5 Definitions of Variation of complex activities

The following concepts are defined for Variation of complex activities.

24.5.1 min_equiv

Two minimal activity trees for ?a are isomorphic if they can each be embedded into the other as a subtree.

```
(forall (?occ1 ?occ2 ?a) (iff (min_equiv ?occ1 ?occ2 ?a)
(and    (subtree_embed ?occ1 ?occ2 ?a)
        (subtree_embed ?occ2 ?occ1 ?a))))
```

24.5.2 uniform

An activity is uniform if all of its minimal activity trees are isomorphic.

```
(forall (?a) (iff (uniform ?a)
(forall (?occ1 ?occ2)
        (implies  (and    (root ?occ1 ?a)
                          (root ?occ2 ?a)
                  (min_equiv ?occ1 ?occ2 ?a)))))
```

24.5.3 variegated

An activity ?a is variegated if all of its minimal activity trees whose root occurrences are occurrence-equivalent are also isomorphic.

```
(forall (?a) (iff (variegated ?a)
(and    (exists (?a1)
            (forall (?occ1 ?occ2)
                (implies  (and    (occurrence_of ?occ1 ?a1)
                          (occurrence_of ?occ2 ?a1)
                          (root ?occ1 ?a)
                          (root ?occ2 ?a))
                      (min_equiv ?occ1 ?occ2 ?a))))
    (exists (?a2 ?occ3 ?occ4)
        (and    (occurrence_of ?occ3 ?a2)
            (occurrence_of ?occ4 ?a2)
            (root ?occ3 ?a)
            (root ?occ4 ?a)
            (not (min_equiv ?occ3 ?occ4 ?a))))))
```

24.5.4 multiform

An activity is multiform if there exist nonisomorphic activity trees whose root occurrences are occurrence equivalent.

```
(forall (?a) (iff (multiform ?a)
(forall (?a1 ?occ1)
    (implies  (and  (root ?occ1 ?a)
            (occurrence_of ?occ1 ?a1))
         (exists (?occ1 ?occ2)
            (and    (occurrence_of ?occ1 ?a1)
                (occurrence_of ?occ2 ?a1)
                (root ?occ1 ?a)
                (root ?occ2 ?a)
                (not (min_equiv ?occ1 ?occ2 ?a))))))))
```

24.6 Grammar for process descriptions of Variation of complex activities

The following grammar sentences describe process descriptions specified in KIF for Variation of complex activities.

```
< uniform_axiom > ::=   (forall (?s1 ?s2)
                (iff    (do < term > ?s1 ?s2)
                    < variation_formula >+))
< variegated_axiom > ::=  (forall (?s ?s2 < variable >+)
                (iff    (do ?a ?s1 ?s2)
                    < occ_variation_formula >+))
< variation_formula > ::=       (exists (< variable >*)
                    (and    (subactivity < term > < term >)
                          (do < term > ?s1 ?s2)))
< occ_variation_formula > ::= (implies  < successor_axiom >
                    < variation_formula >)
```

25 State-based variation of complex activities

This clause characterizes all definitions pertaining to State-based variation of complex activities. The criterion used to classify these activities is whether or not occurrences of the subactivities of a complex activity depend only on the state prior to the occurrences.

25.1 Primitive lexicon of State-based variation of complex activities

No primitive relations are required by the lexicon of State-based variation of complex activities.

25.2 Defined relations of State-based variation of complex activities

The following relations are defined in this clause:

- (conditional ?a);
- (partial_conditional ?a);
- (rigid_conditional ?a).

Each concept is described by informal semantics and a KIF axiom.

25.3 Theories required by State-based variation of complex activities

This theory requires:

- occtree.th;
- psl_core.th.

25.4 Definitional extensions required by State-based variation of complex activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

25.5 Definitions of State-based variation of complex activities

The following concepts are defined for State-based variation of complex activities.

25.5.1 conditional

An activity is conditional if any two of its minimal activity trees are isomorphic if the roots agree on state.

```
(forall (?a) (iff (conditional ?a)
(forall (?occ1 ?occ2)
    (implies  (and  (root ?occ1 ?a)
                (root ?occ2 ?a)
                (state_equiv ?occ1 ?occ2))
        (min_equiv ?occ1 ?occ2 ?a)))))
```

25.5.2 partial_conditional

An activity is partial conditional if there exist isomorphic minimal activity trees that agree on state.

```
(forall (?a) (iff (partial_conditional ?a)
(and    (exists (?occ1)
        (forall (?occ2)
            (implies  (and  (root ?occ1 ?a)
                        (root ?occ2 ?a)
                        (state_equiv ?occ1 ?occ2))
                 (min_equiv ?occ1 ?occ2 ?a))))
    (exists (?occ3 ?occ4)
        (and    (root ?occ3 ?a)
            (root ?occ4 ?a)
            (state_equiv ?occ3 ?occ4)
            (not (min_equiv ?occ3 ?oc4)))))))
```

25.5.3 rigid_conditional

An activity is rigid conditional if for every root occurrence of a minimal activity tree for ?a, there exists another occurrence that agrees on state but which is the root of a nonisomorphic minimal activity tree.

```
(forall (?a) (iff (rigid_conditional ?a)
(forall (?occ1)
    (exists (?occ2)
        (and    (root ?occ1 ?a)
            (root ?occ2 ?a)
            (state_equiv ?occ1 ?occ2)
            (not (min_equiv ?occ1 ?occ2 ?a))))))
```

25.6 Grammar for process descriptions of State-based variation of complex activities

The following grammar sentences describe process descriptions specified in KIF for State-based variation of complex activities

```
< conditional_activity > ::=    (forall (?s ?s2)
                               (iff    (do ?a ?s ?s2)
                                       < simple_conditional >))
< partial_conditional > ::=    (forall (?s ?s2 < variable >+)
                               (iff    (do ?a ?s ?s2)
                                       < conditional_formula >))
< simple_conditional > ::=      (implies < simple_state_axiom >
                                < variation_formula >)
< conditional_formula > ::=     (implies < state_axiom >
                                < variation_formula >)
```

26 Time-based variation of complex activities

This clause characterizes all definitions pertaining to Time-based variation of complex activities. The criterion used to classify these activities is whether or not occurrences of the subactivities of a complex activity depend only on the times at which the activity occurs.

26.1 Primitive lexicon of Time-based variation of complex activities

No primitive relations are required by the lexicon of Time-based variation of complex activities.

26.2 Defined relations of Time-based variation of complex activities

The following relations are defined in this clause:

— (time_conditional ?a);

— (partial_time_conditional ?a);

— (rigid_time_conditional ?a).

Each concept is described by informal semantics and a KIF axiom.

26.3 Theories required by Time-based variation of complex activities

This theory requires:

— occtree.th;

— psl_core.th.

26.4 Definitional extensions required by Time-based variation of complex activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

26.5 Definitions of Time-based variation of complex activities

The following concepts are defined for Preventable conditions for activities.

26.5.1 time_conditional

A complex activity is time conditional iff any two of its minimal activity trees are isomorphic iff the roots agree on their beginof timepoints.

```
(forall (?a) (iff (time_conditional ?a)
(forall (?occ1 ?occ2)
     (implies  (and  (root ?occ1 ?a)
                (root ?occ2 ?a)
                (begin_equiv ?occ1 ?occ2))
           (min_equiv ?occ1 ?occ2 ?a)))))
```

26.5.2 partial_time_conditional

A complex activity is partial time conditional if and only if there exist occurrence-preserving permutations.

```
(forall (?a) (iff (partial_time_conditional ?a)
(and    (exists (?occ1)
          (forall (?occ2)
                (implies (and  (root ?occ1 ?a)
                          (root ?occ2 ?a)
                          (begin_equiv ?occ1 ?occ2))
                      (min_equiv ?occ1 ?occ2 ?a))))
     (exists (?occ3 ?occ4)
          (and    (root ?occ3 ?a)
                (root ?occ4 ?a)
                (begin_equiv ?occ3 ?occ4)
                (not (min_equiv ?occ3 ?oc4)))))))
```

26.5.3 rigid_time_conditional

A complex activity is rigid time conditional if and only if the only occurrence-preserving timeline permutation is the trivial one.

```
(forall (?a) (iff (rigid_time_conditional ?a)
(forall (?occ1)
        (exists (?occ2)
                (and    (root ?occ1 ?a)
                        (root ?occ2 ?a)
                        (begin_equiv ?occ1 ?occ2)
                        (not (min_equiv ?occ1 ?occ2 ?a)))))))
```

26.6 Grammar for process descriptions of Time-based variation of complex activities

The following grammar sentences describe process descriptions specified in KIF for Time-based variation of complex activities.

```
< time_conditional_activity > ::=    (forall (?s ?s2)
                                  (iff    (do ?a ?s ?s2)
                                          < simple_time_conditional >))
< partial_time_conditional > ::=  (forall (?s ?s2)
                                  (iff    (do ?a ?s ?s2)
                                          < time_conditional_formula >))
< simple_time_conditional > ::=  (implies < simple_time_axiom >
                                    < variation_formula >)
< time_conditional_formula > ::=  (implies < time_axiom >
                                    < variation_formula >)
```

27 Mixed variation of complex activities

This clause characterizes all definitions pertaining to Mixed variation of complex activities. The criterion used to classify these activities is whether or not occurrences of the subactivities of a complex activity depend on the state prior to the occurrences and the times at which the activity occurs.

27.1 Primitive lexicon of Mixed variation of complex activities

No primitive relations are required by the lexicon of Mixed variation of complex activities.

27.2 Defined relations of Mixed variation of complex activities

The following relations are defined in this clause:

— (mixed_conditional ?a);

— (partial_mixed_conditional ?a);

— (rigid_mixed_conditional ?a).

Each concept is described by informal semantics and a KIF axiom.

27.3 Theories required by Mixed variation of complex activities

This theory requires:

- occtree.th;
- psl_core.th.

27.4 Definitional extensions required by Mixed variation of complex activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

27.5 Definitions of Mixed variation of complex activities

The following concepts are defined for Mixed variation of complex activities.

27.5.1 mixed_conditional

An activity is mixed conditional if and only if any two of its minimal activity trees are isomorphic if and only if the roots agree on state and the timepoint and which they begin.

```
(forall (?a) (iff (mixed_conditional ?a)
(forall (?occ1 ?occ2)
    (implies  (and  (root ?occ1 ?a)
                    (root ?occ2 ?a)
                    (begin_equiv ?occ1 ?occ2)
                    (state_equiv ?occ1 ?occ2))
                 (min_equiv ?occ1 ?occ2 ?a)))))
```

27.5.2 partial_mixed_conditional

An activity is partial mixed conditional if and only if there exist minimal activity trees that are isomorphic if and only if the roots agree on state and the timepoint and which they begin.

```
(forall (?a) (iff (partial_mixed_conditional ?a)
(and    (exists (?occ1)
            (forall (?occ2)
                    (implies  (and  (root ?occ1 ?a)
                                    (root ?occ2 ?a)
                                    (begin_equiv ?occ1 ?occ2)
                                    (state_equiv ?occ1 ?occ2))
                            (min_equiv ?occ1 ?occ2 ?a))))
            (exists (?occ3 ?occ4)
            (and    (root ?occ3 ?a)
                    (root ?occ4 ?a)
                    (begin_equiv ?occ1 ?occ2)
                    (state_equiv ?occ3 ?occ4)
                    (not (min_equiv ?occ3 ?oc4)))))))
```

27.5.3 rigid_mixed_conditional

An activity is rigid mixed conditional if and only if the only permutations that preserve occurrences, timepoints, and state are the trivial ones.

```
(forall (?a) (iff (rigid_mixed_conditional ?a)
(forall (?occ1)
    (exists (?occ2)
        (and    (root ?occ1 ?a)
            (root ?occ2 ?a)
            (begin_equiv ?occ1 ?occ2)
            (state_equiv ?occ1 ?occ2)
            (not (min_equiv ?occ1 ?occ2 ?a)))))))
```

27.6 Grammar for process descriptions of Mixed variation of complex activities

The following grammar sentences describe process descriptions specified in KIF for Preventable conditions for activities

```
< mixed_conditional > ::=       (forall (?s ?s2)
                        (iff    (do ?a ?s ?s2)
                            < simple_mix_conditional >))
< partial_mixed_conditional > ::=  (forall (?s ?s2 < variable >+)
                        (iff    (do ?a ?s ?s2)
                            < mix_conditional_formula >))
< simple_mix_conditional > ::=      (implies < simple_mix_axiom >
                        < variation_formula >)
< mixed_conditional_formula > ::=    (implies < mix_axiom >
                        < variation_formula >)
```

28 Embedded activities: plans

This clause characterizes all definitions pertaining to Embedded activities: plans. The criterion used to classify these activities is whether or not occurrences of the subactivities of a complex activity depend only on the state during an occurrence of the activity.

28.1 Primitive lexicon of Embedded activities: plans

No primitive relations are required by the lexicon of Embedded activities: plans.

28.2 Defined relations of Embedded activities: plans

The following relations are defined in this clause:

- (preventable ?a);
- (possibly_preventable ?a);
- (unpreventable ?a) .

Each concept is described by informal semantics and a KIF axiom.

28.3 Theories required by Embedded activities: plans

This theory requires:

- occtree.th;
- psl_core.th.

28.4 Definitional extensions required by Embedded activities: plans

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

28.5 Definitions of Embedded activities: plans

The following concepts are defined for Embedded activities: plans.

28.5.1 plan

An activity tree with root occurrence ?s is a plan if and only if all occurrences of subactivities in the maximal embedded subtree that agree on state also agree on being subactivity occurrences.

```
(forall (?s) (iff (plan ?s)
(forall (?a ?s1 ?s2)
    (implies  (and  (root ?s ?a)
                (embed_tree ?s1 ?s2 ?s ?a)
                (state_equiv ?s1 ?s2))
            (subocc_equiv ?s1 ?s2 ?s ?a)))))
```

28.5.2 nondet_plan

An activity tree with root occurrence ?s is a nondeterministic plan if and only if there exist occurrences of subactivities in the maximal embedded subtree that agree on state and that also agree on being subactivity occurrences.

```
(forall (?s) (iff (nondet_plan ?s)
(exists (?a ?s1 ?s3 ?s4)
    (and    (root ?s ?a)
        (forall (?s2)
            (implies  (and  (embed_tree ?s1 ?s2 ?s ?a)
                        (state_equiv ?s1 ?s2))
                     (subocc_equiv ?s1 ?s2 ?s ?a))))
        (state_equiv ?s3 ?s4)
        (embed_tree ?s3 ?s4 ?s5 ?a)
        (not (subocc_equiv ?s3 ?s4 ?s5 ?a))))))
```

28.5.3 unplan

An activity tree with root occurrence ?s is an unplan if and only if for every subactivity occurrence, there exists another occurrence of the subactivity that agrees on state but that is not a subactivity occurrence.

```
(forall (?s) (iff (unplan ?s)
(forall (?s1 ?a)
    (implies  (root ?s ?a)
         (exists (?s2)
            (and    (embed_tree ?s1 ?s2 ?s ?a)
                (state_equiv ?s1 ?s2)
                (not (subocc_equiv ?s1 ?s2 ?s ?a))))))))
```

28.6 Grammar for process descriptions of Embedded activities: plans

The following grammar sentences describe process descriptions specified in KIF for Embedded activities: plans

```
< plan_axiom > ::=      (forall (?s ?occ < variable >*)
                    < simple_plan_axiom >)
< nondet_plan_axiom > ::=  (forall (?s ?occ < variable >+)
                    < nondet_plan_axiom >))
< plan_formula > ::=    (implies < subocc_formula >
                     < simple_state_axiom >)
< simple_plan_axiom > ::=      < plan_formula> |
                          (and < simple_plan_axiom > < simple_plan_axiom >+)
< nondet_plan_formula > ::=    (implies < subocc_formula >
                                < state_axiom >)
< nondet_plan_axiom > ::=      < nondet_plan_formula> |
                          (and < nondet_plan_axiom > < nondet_plan_axiom >+)
```

29 Embedded activities: temporal spread

This clause characterizes all definitions pertaining to Embedded activities: temporal spread. The criterion used to classify these activities is whether or not occurrences of the subactivities of a complex activity depend on the times at which the subactivities occur during an occurrence of the activity.

29.1 Primitive lexicon of Embedded activities: temporal spread

No primitive relations are required by the lexicon of Embedded activities: temporal spread.

29.2 Defined relations of Embedded activities: temporal spread

The following relations are defined in this clause:

- (preventable ?a);
- (possibly_preventable ?a);
- (unpreventable ?a).

Each concept is described by informal semantics and a KIF axiom.

29.3 Theories required by Embedded activities: temporal spread

This theory requires:

- occtree.th;
- psl_core.th.

29.4 Definitional extensions required by Embedded activities: temporal spread

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

29.5 Definitions of Embedded activities: temporal spread

The following concepts are defined for Embedded activities: temporal spread.

29.5.1 spread

An activity occurrence ?occ is spread if and only if every subactivity occurrence of ?occ is temporally constrained.

```
(forall (?occ) (iff (spread ?occ)
(forall (?a ?s1 ?s2 ?s3)
    (implies  (and  (occurrence_of ?occ ?a)
                (root_occ ?s3 ?occ)
                (subactivity_occurrence ?s1 ?occ)
                (iso_occ ?s1 ?s2)
                (embed_tree ?s1 ?s2 ?s3 ?a)
                (begin_equiv ?s1 ?s2))
            (subocc_equiv ?s1 ?s2 ?s3 ?a)))))
```

29.5.2 partial_spread

An activity occurrence ?occ is partially spread if and only if some subactivity occurrences of ?occ are temporally constrained.

```
(forall (?occ) (iff (partial_spread ?occ)
(exists (?a ?s ?s1 ?s3 ?s4)
    (and    (occurrence_of ?occ ?a)
        (root_occ ?s ?occ)
        (subactivity_occurrence ?s1 ?occ)
        (forall (?s2)
            (implies  (and  (iso_occ ?s1 ?s2)
                    (embed_tree ?s1 ?s2 ?s ?a)
                    (begin_equiv ?s1 ?s2))
                (subocc_equiv ?s1 ?s2 ?s ?a))))
        (subactivity_occurrence ?s3 ?occ)
        (iso_occ ?s3 ?s4)
        (begin_equiv ?s3 ?s4)
        (embed_tree ?s3 ?s4 ?s ?a)
        (not (subocc_equiv ?s3 ?S4 ?s ?a))))))
```

29.5.3 tight

An activity occurrence ?occ is partially tight if and only if none of the subactivity occurrences of ?occ are temporally constrained.

```
(forall (?occ) (iff (tight ?occ)
(forall (?s ?s1 ?a)
      (implies  (and  (occurrence_of ?occ ?a)
                (root_occ ?s ?occ)
                (subactivity_occurrence ?s1 ?occ))
            (exists (?s2)
                (and    (iso_occ ?s1 ?s2)
                    (embed_tree ?s1 ?s2 ?s ?a)
                    (begin_equiv ?s1 ?s2)
                    (not (subocc_equiv ?s1 ?s2 ?s ?a)))))))
```

29.6 Grammar for process descriptions of Embedded activities: temporal spread

The following grammar sentences describe process descriptions specified in KIF for Embedded activities: temporal spread

```
< spread_axiom > ::=    (forall (?s ?occ < variable >*)
                    < simple_spread_axiom >)
< partial_spread_axiom > ::=  (forall (?s ?occ < variable >+)
                     < partial_spread_axiom >))
< spread_formula > ::=  (implies < subocc_formula >
                    < simple_time_axiom >)
< simple_spread_axiom > ::=     < spread_formula> |
                    (and < simple_spread_axiom > < simple_spread_axiom >+)
< partial_spread_formula > ::=  (implies < subocc_formula >
                         < time_axiom >)
< partial_spread_axiom > ::=    < partial_spread_formula> |
                    (and < partial_spread_axiom > < partial_spread_axiom >+)
```

30 Variation for upwards atomic activities

This clause characterizes all definitions pertaining to Variation for upwards atomic activities.

30.1 Primitive lexicon of Variation for upwards atomic activities

No primitive relations are required by the lexicon of Variation for upwards atomic activities. The criterion used to classify these activities is whether or not preconditions of the atomic activity depend on the preconditions for superactivities of the activity.

30.2 Defined relations of Variation for upwards atomic activities

The following relations are defined in this clause:

— (state_filter ?a);

— (partial_state_filter ?a);

- (rigid_state_filter ?a) ;
- (time_filter ?a);
- (partial_time_filter ?a);
- (rigid_time_filter ?a) ;
- (state_nonfilter ?a);
- (partial_state_nonfilter ?a);
- (rigid_state_nonfilter ?a) ;
- (time_nonfilter ?a) ;
- (partial_time_nonfilter ?a);
- (rigid_time_nonfilter ?a).

Each concept is described by informal semantics and a KIF axiom.

30.3 Theories required by Variation for upwards atomic activities

This theory requires:

- occtree.th;
- psl_core.th.

30.4 Definitional extensions required by Variation for upwards atomic activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

30.5 Definitions of Variation for upwards atomic activities

The following concepts are defined for Variation for upwards atomic activities.

30.5.1 state_filter

An atomic activity?a is a state_filter activity if and only if any whenever legal occurrences of ?a agree on state, then all subactivities of ?a can legally occur.

```
(forall (?a) (iff (state_filter ?a)
(forall (?a1 ?s1 ?s2)
    (implies  (and  (subactivity ?a1 ?a)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (state_equiv ?s1 ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

30.5.2 partial_state_filter

An atomic activity?a is a partial_state_filter activity if and only if there exist legal occurrences of ?a that agree on state and in which all subactivities of ?a can legally occur.

```
(forall (?a) (iff (partial_state_filter ?a)
(and    (exists (?s1)
        (forall (?s2)
            (implies  (and  (subactivity ?a1 ?a)
                (poss ?a ?s1)
                (poss ?a ?s2)
                (state_equiv ?s1 ?s2))
             (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and    (subactivity ?a2 ?a)
            (poss ?a ?s3)
            (poss ?a ?s4)
            (state_equiv ?s3 ?s4)
            (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.3 rigid_state_filter

An atomic activity?a is a state_filter activity if and only if any whenever legal occurrences of ?a agree on state, then there exist subactivities of ?a that cannot legally occur.

```
(forall (?a) (iff (rigid_state_filter ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and    (state_equiv ?s1 ?s2)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.5.4 time_filter

An atomic activity?a is a time_filter activity if and only if any whenever legal occurrences of ?a agree on their beginof timepoints, then all subactivities of ?a can legally occur.

```
(forall (?a) (iff (time_filter ?a)
(forall (?a1 ?s1 ?s2)
    (implies  (and  (subactivity ?a1 ?a)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (begin_equiv ?s1 ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

30.5.5 partial_time_filter

An atomic activity?a is a partial_time_filter activity if and only if there exist legal occurrences of ?a that agree on their beginof timepoints and in which all subactivities of ?a can legally occur.

```
(forall (?a) (iff (partial_time_filter ?a)
(and    (exists (?s1)
        (forall (?s2)
            (implies  (and  (subactivity ?a1 ?a)
                    (poss ?a ?s1)
                    (poss ?a ?s2)
                    (begin_equiv ?s1 ?s2))
                 (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and    (subactivity ?a2 ?a)
            (poss ?a ?s3)
            (poss ?a ?s4)
            (begin_equiv ?s3 ?s4)
            (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.6 rigid_time_filter

An atomic activity?a is a rigid_time_filter activity if and only if any whenever legal occurrences of ?a agree on their beginof timepoints, then there exist subactivities of ?a that cannot legally occur.

```
(forall (?a) (iff (rigid_time_filter ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and    (begin_equiv ?s1 ?s2)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.5.7 state_nonfilter

An atomic activity?a is a state_nonfilter activity if and only if any whenever legal occurrences of ?a agree on state, then all occurrence of subactivities of ?a agree on legality.

```
(forall (?a) (iff (state_nonfilter ?a)
(forall (?a1 ?s1 ?s2)
    (implies  (and  (subactivity ?a1 ?a)
            (not (poss ?a ?s1))
            (not (poss ?a ?s2))
            (state_equiv ?s1 ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

30.5.8 partial_state_nonfilter

An atomic activity?a is a partial_state_nonfilter activity if and only if there exist legal occurrences of ?a that agree on state and in which all subactivities of ?a agree on legality.

```
(forall (?a) (iff (partial_state_nonfilter ?a)
(and    (exists (?s1)
        (forall (?s2)
            (implies  (and  (subactivity ?a1 ?a)
                    (not (poss ?a ?s1))
                    (not (poss ?a ?s2))
                    (state_equiv ?s1 ?s2))
                 (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and    (subactivity ?a2 ?a)
            (not (poss ?a ?s3))
            (not (poss ?a ?s4))
            (state_equiv ?s3 ?s4)
            (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.9 rigid_state_nonfilter

An atomic activity ?a is a rigid_state_nonfilter activity if and only if any whenever legal occurrences of ?a agree on state, then there exist subactivities of ?a that do not agree on legality.

```
(forall (?a) (iff (rigid_state_nonfilter ?a)
(forall (?s1)
    (exists (?a1 ?s2)
          (and    (state_equiv ?s1 ?s2)
               (not (poss ?a ?s1))
               (not (poss ?a ?s2))
               (not (poss_equiv ?a1 ?s1 ?s2))))))
```

30.5.10 time_nonfilter

An atomic activity ?a is a state_nonfilter activity if and only if any whenever legal occurrences of ?a agree on their beginof timepoints, then all occurrence of subactivities of ?a agree on legality.

```
(forall (?a) (iff (time_nonfilter ?a)
(forall (?a1 ?s1 ?s2)
     (implies  (and  (subactivity ?a1 ?a)
               (not (poss ?a ?s1))
               (not (poss ?a ?s2))
               (begin_equiv ?s1 ?s2))
           (poss_equiv ?a1 ?s1 ?s2)))))
```

30.5.11 partial_time_nonfilter

An atomic activity ?a is a partial_state_nonfilter activity if and only if there exist legal occurrences of ?a that agree on their beginof timepoints and in which all subactivities of ?a agree on legality.

```
(forall (?a) (iff (partial_time_nonfilter ?a)
(and    (exists (?s1)
          (forall (?s2)
               (implies  (and  (subactivity ?a1 ?a)
                         (not (poss ?a ?s1))
                         (not (poss ?a ?s2))
                         (begin_equiv ?s1 ?s2))
                     (poss_equiv ?a1 ?s1 ?s2))))
     (exists (?a2 ?s3 ?s4)
          (and    (subactivity ?a2 ?a)
               (not (poss ?a ?s3))
               (not (poss ?a ?s4))
               (begin_equiv ?s3 ?s4)
               (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

30.5.12 rigid_time_nonfilter

An atomic activity ?a is a rigid_state_nonfilter activity if and only if any whenever legal occurrences of ?a agree on their beginof timepoints, then there exist subactivities of ?a that do not agree on legality.

```
(forall (?a) (iff (rigid_time_nonfilter ?a)
(forall (?s1)
        (exists (?a1 ?s2)
                (and    (begin_equiv ?s1 ?s2)
                        (not (poss ?a ?s1))
                        (not (poss ?a ?s2))
                        (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

30.6 Grammar for process descriptions of Variation for upwards atomic activities

The grammar for process descriptions for classes of activities defined in Variation off clobbering effects is equivalent to the grammar of a general process description sentence as specified in ISO 18629-11.

31 Variation for downwards atomic activities

This clause characterizes all definitions pertaining to Variation for downwards atomic activities. The criterion used to classify these activities is whether or not preconditions of the atomic activity depend on the preconditions for subactivities of the activity

31.1 Primitive lexicon of Variation for downwards atomic activities

No primitive relations are required by the lexicon of Variation for downwards atomic activities.

31.2 Defined relations of Variation for downwards atomic activities

The following relations are defined in this clause:

- (state_ideal ?a);
- (partial_state_ideal ?a);
- (rigid_state_ideal ?a) ;
- (time_ideal ?a);
- (partial_time_ideal ?a);
- (rigid_time_ideal ?a) ;
- (state_nonideal ?a);
- (partial_state_nonideal ?a);
- (rigid_state_nonideal ?a) ;
- (time_nonideal ?a) ;
- (partial_time_nonideal ?a);

- (rigid_time_nonideal ?a).

Each concept is described by informal semantics and a KIF axiom.

31.3 Theories required by Variation for downwards atomic activities

This theory requires:

- occtree.th;
- psl_core.th.

31.4 Definitional extensions required by Variation for downwards atomic activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

31.5 Definitions of Variation for downwards atomic activities

The following concepts are defined for Variation for downwards atomic activities.

31.5.1 state_ideal

An atomic activity ?a is a state_ideal activity if and only if any whenever legal occurrences of ?a agree on state, then all superactivities of ?a agree on legality.

```
(forall (?a) (iff (state_ideal ?a)
(forall (?a1 ?s1 ?s2)
     (implies  (and  (subactivity ?a ?a1)
                     (poss ?a ?s1)
                     (poss ?a ?s2)
                     (state_equiv ?s1 ?s2))
                 (poss_equiv ?a1 ?s1 ?s2)))))
```

31.5.2 partial_state_ideal

An atomic activity ?a is a partial_state_ideal activity if and only if there exist legal occurrences of ?a that agree on state and in which all superactivities of ?a agree on legality.

```
(forall (?a) (iff (partial_state_ideal ?a)
(and    (exists (?s1)
             (forall (?s2)
                    (implies  (and  (subactivity ?a ?a1)
                                    (poss ?a ?s1)
                    (poss ?a ?s2)
                    (state_equiv ?s1 ?s2))
                 (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and    (subactivity ?a ?a2)
            (poss ?a ?s3)
            (poss ?a ?s4)
            (state_equiv ?s3 ?s4)
            (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.3 rigid_state_ideal

An atomic activity ?a is a rigid_state_ideal activity if and only if any whenever legal occurrences of ?a agree on state, then there exist superactivities of ?a that do not agree on legality.

```
(forall (?a) (iff (rigid_state_ideal ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and    (state_equiv ?s1 ?s2)
            (subactivity ?a ?a1)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.5.4 time_ideal

An atomic activity ?a is a time_ideal activity if and only if any whenever legal occurrences of ?a agree on beginof timepoints, then all superactivities of ?a agree on legality.

```
(forall (?a) (iff (time_ideal ?a)
(forall (?a1 ?s1 ?s2)
    (implies  (and  (subactivity ?a ?a1)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (begin_equiv ?s1 ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

31.5.5 partial_time_ideal

An atomic activity ?a is a partial_time_ideal activity if and only if there exist legal occurrences of ?a that agree on their beginof timepoints and in which all superactivities of ?a agree on legality.

```
(forall (?a) (iff (partial_time_ideal ?a)
(and    (exists (?s1)
        (forall (?s2)
            (implies  (and  (subactivity ?a ?a1)
                    (poss ?a ?s1)
                    (poss ?a ?s2)
                    (begin_equiv ?s1 ?s2))
                 (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and    (subactivity ?a ?a2)
            (poss ?a ?s3)
            (poss ?a ?s4)
            (begin_equiv ?s3 ?s4)
            (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.6 rigid_time_ideal

An atomic activity ?a is a rigid_state_nonideal activity if and only if any whenever legal occurrences of ?a agree on state, then there exist superactivities of ?a that do not agree on legality.

```
(forall (?a) (iff (rigid_time_ideal ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and    (begin_equiv ?s1 ?s2)
            (poss ?a ?s1)
            (poss ?a ?s2)
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.5.7 state_nonideal

An atomic activity ?a is a state_nonideal activity if and only if any whenever legal occurrences of ?a agree on state, then all occurrence of superactivities of ?a agree on legality.

```
(forall (?a) (iff (state_nonideal ?a)
(forall (?a1 ?s1 ?s2)
    (implies  (and  (subactivity ?a ?a1)
            (not (poss ?a ?s1))
            (not (poss ?a ?s2))
            (state_equiv ?s1 ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

31.5.8 partial_state_nonideal

An atomic activity ?a is a partial_state_nonideal activity if and only if there exist legal occurrences of ?a that agree on state and in which all superactivities of ?a agree on legality.

```
(forall (?a) (iff (partial_state_nonideal ?a)
(and    (exists (?s1)
            (forall (?s2)
                  (implies  (and  (subactivity ?a ?a1)
                              (not (poss ?a ?s1))
                              (not (poss ?a ?s2))
                              (state_equiv ?s1 ?s2))
                        (poss_equiv ?a1 ?s1 ?s2))))
      (exists (?a2 ?s3 ?s4)
            (and    (subactivity ?a ?a2)
                  (not (poss ?a ?s3))
                  (not (poss ?a ?s4))
                  (state_equiv ?s3 ?s4)
                  (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.9 rigid_state_nonideal

An atomic activity ?a is a rigid_state_nonideal activity if and only if any whenever legal occurrences of ?a agree on state, then there exist superactivities of ?a that do not agree on legality.

```
(forall (?a) (iff (rigid_state_nonideal ?a)
(forall (?s1)
      (exists (?a1 ?s2)
            (and    (state_equiv ?s1 ?s2)
                  (not (poss ?a ?s1))
                  (not (poss ?a ?s2))
                  (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.5.10 time_nonideal

An atomic activity ?a is a time_nonideal activity if and only if any whenever legal occurrences of ?a agree on their beginof timepoints, then all occurrence of superactivities of ?a agree on legality.

```
(forall (?a) (iff (time_nonideal ?a)
(forall (?a1 ?s1 ?s2)
      (implies  (and  (subactivity ?a ?a1)
                  (not (poss ?a ?s1))
```

```
            (not (poss ?a ?s2))
            (begin_equiv ?s1 ?s2))
         (poss_equiv ?a1 ?s1 ?s2)))))
```

31.5.11partial_time_nonideal

An atomic activity ?a is a partial_time_nonideal activity if and only if there exist legal occurrences of ?a that agree on their beginof timepoints and in which all superactivities of ?a agree on legality.

```
(forall (?a) (iff (partial_time_nonideal ?a)
(and    (exists (?s1)
        (forall (?s2)
            (implies  (and  (subactivity ?a ?a1)
                    (not (poss ?a ?s1))
                    (not (poss ?a ?s2))
                    (begin_equiv ?s1 ?s2))
                 (poss_equiv ?a1 ?s1 ?s2))))
    (exists (?a2 ?s3 ?s4)
        (and    (subactivity ?a ?a2)
            (not (poss ?a ?s3))
            (not (poss ?a ?s4))
            (begin_equiv ?s3 ?s4)
            (not (poss_equiv ?a2 ?s3 ?s4)))))))
```

31.5.12 rigid_time_nonideal

An atomic activity ?a is a rigid_time_nonideal activity if and only if any whenever legal occurrences of ?a agree on their beginof timepoints, then there exist superactivities of ?a that do not agree on legality.

```
(forall (?a) (iff (rigid_time_nonideal ?a)
(forall (?s1)
    (exists (?a1 ?s2)
        (and    (begin_equiv ?s1 ?s2)
            (subactivity ?a ?a1)
            (not (poss ?a ?s1))
            (not (poss ?a ?s2))
            (not (poss_equiv ?a1 ?s1 ?s2)))))))
```

31.6 Grammar for process descriptions of Variation for downwards atomic activities

The grammar for process descriptions for classes of activities defined in Variation for downwards atomic activities is equivalent to the grammar of a general process description sentence as specified in ISO 18629-11.

32 Preconditions for concurrent activities

This clause characterizes all definitions pertaining to Preconditions for concurrent activities. The criterion used to classify these activities is whether or not preconditions of the atomic activity are preserved by concurrent composition.

32.1 Primitive lexicon of Preconditions for concurrent activities

No primitive relations are required by the lexicon of Preconditions for concurrent activities.

32.2 Defined relations of Preconditions for concurrent activities

The following relations are defined in this clause:

- (preserved_filter ?a ?a1 ?s);
- (noninterfering ?a) ;
- (interfering ?a) ;
- (imperial ?a) ;
- (global_interfere ?a).

Each concept is described by informal semantics and a KIF axiom.

32.3 Theories required by Preconditions for concurrent activities

This theory requires:

- occtree.th;
- psl_core.th.

32.4 Definitional extensions required by Preconditions for concurrent activities

This extension requires:

- occ_precond.def;
- state_precond.def;
- precond.def.

32.5 Definitions of Preconditions for concurrent activities

The following concepts are defined for Preconditions for concurrent activities.

32.5.1 preserved_filter

Activities ?a and ?a1 are elements of a preserved filter if and only if legal occurrences of any superactivity of ?a are preserved by the concurrent composition with ?a1.

```
(forall (?a ?a1 ?s) (iff (preserved_filter ?a ?a1 ?s)
(forall (?a2)
        (implies  (subactivity ?a ?a2)
                  (iff  (poss ?a2 ?s)
                        (poss (conc ?a2 ?a1) ?s))))))
```

32.5.2 noninterfering

An activity is noninterfering if and only if every atomic activity is an element of a preserved filter with ?a.

```
(forall (?a ?s) (iff (noninterfering ?a ?s)
(forall (?a1)
        (implies  (atomic ?a1)
                  (preserved_filter ?a ?a1 ?s)))))
```

32.5.3 interfering

An activity is interfering if and only if there exists an atomic activity that is not an element of a preerved filter with ?a.

```
(forall (?a ?s) (iff (interfering ?a ?s)
(exists (?a1)
        (and    (atomic ?a1)
                (not (preserved_filter ?a ?a1 ?s))))))
```

32.5.4 imperial

An activity is imperial if and only if there does not exist an atomic activity that is an element of a preserved filter with ?a.

```
(forall (?a ?s) (iff (imperial ?a ?s)
(forall (?a1)
        (implies  (atomic ?a1)
                  (not (preserved_filter ?a ?a1 ?s))))))
```

32.5.5 global_interfere

An activity is a global_interfere activity if and only if all occurrences of ?a have the same preserved filters for ?a.

```
(forall (?a ?s) (iff (global_interfere ?a)
(forall (?a1)
        (implies  (atomic ?a1)
```