

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Methodology	2
2	KBS Concepts	3
	The Knowledge Base.	3
	Contrast to Conventional Techniques.	3
	Value of KBS Techniques.	3
2.1	System-level Concepts	4
2.1.1	KBS applications	4
2.1.2	KBS techniques	4
2.1.3	KBS languages	4
2.2	Computational Concepts	5
2.2.1	Facts and Rules	5
2.2.2	Formal Representation of Facts and Rules	5
	Abstract Model.	5
	Design of Knowledge Representation.	6
2.2.3	Digraph Model of Derivation	7
2.2.4	Forward-chaining and Backward-chaining	9
	Finding a Relevant Rule.	10
	Applicability.	11
2.2.5	Pattern-matching	11
2.2.6	Inference Engine	14
2.2.7	Control Strategy	14
2.2.8	Contrast to Conventional Techniques, Revisited	15
2.3	Advanced Concepts	16

3	Language Support for KBS	18
3.1	The Knowledge Base	18
3.1.1	Atomic Values	18
3.1.2	Constants and Variables	18
3.1.3	Facts	19
	Keyword vs. Positional.	20
	Constraints.	21
3.1.4	Rules and Patterns	21
	Identification of Rules.	21
	Computed Expressions in Rules.	22
	If-part Features in Lisp.	22
	If-part Features in Prolog.	22
	If-part Features in OPS5.	22
	If-part Summary.	23
	Then-part Features.	23
3.1.5	KB Operations	24
3.2	Derivation	26
3.2.1	Inference Engine	26
3.2.2	User Control	27
3.2.3	Goals and Termination Conditions	28
3.3	User Interface	28
3.3.1	Invoking and Exiting	29
	Invoking.	29
	Exiting.	29
3.3.2	Watching the Derivation	31
3.3.3	Pausing and Stepping	31
	Pausing.	31
	Stepping.	31
3.3.4	Inspecting the KBS	31
	Inspecting KB objects by Name.	32

	Inspecting KB objects by Matching.	32
	Inspecting the State of the Derivation.	32
3.3.5	Manipulating the KBS	32
	Manipulating KB objects.	32
	Manipulating the State of the Derivation.	32
3.3.6	Summary of User Interface	33
3.4	Dimensions of Knowledge Design	33
3.4.1	Review of Knowledge Types	33
3.4.2	Declarative vs. Procedural Style	34
3.4.3	Depth and Width of KBS	35
	Width.	35
	Depth.	35
3.4.4	Global Organization of KB	36
3.4.5	Persistence of KB Operations and Monotonicity	36
3.4.6	Domain Dependence	38
3.4.7	Roles and Responsibilities	38
3.5	Flexibility and Power for KBS Applications	39
3.6	Summary of Language Support for KBS	40
	4 Conventional Language Factors	42
4.1	Syntactic Style	42
4.2	Control of Execution	43
	4.2.1 Structured Programming	43
	4.2.2 Blocks	43
	4.2.3 User-defined Functions and Subroutines	44
4.3	Control of Data	44
	4.3.1 Scope of Data	44
	4.3.2 Data Type Checking	45
	4.3.3 Data Abstraction	45
4.4	Packages	45
4.5	Elementary Data Types and Operations	46

4.5.1	Symbols	46
4.5.2	Numbers	47
4.5.3	Characters	47
4.5.4	Logical	48
4.5.5	Bit	48
4.6	Aggregate Data	48
4.6.1	Lists and Arrays	48
4.6.2	Records	50
4.6.3	Sets	51
4.6.4	Files and I/O	51
4.6.5	Executable Code	52
4.7	Macros and Pre-processing	53
4.8	Miscellaneous Language Features	53
4.9	Simplicity	54
4.10	Standardization	54
4.11	Software Availability	55
5	User Requirements	56
5.1	Functional Operations	56
5.2	End-user Interaction	57
5.3	Size And Complexity	57
5.4	Execution Efficiency	57
5.5	Reliability	57
5.6	Application Stability	58
5.7	Timeframe	58
5.8	Number Of Programmers	58
5.9	Programmer Expertise	59
5.10	Portability	59
5.11	Language Availability and Hardware Support	59
5.12	Compatibility With Existing Software	60

6	Summary and Conclusions	61
6.1	Individual Language Review	61
6.1.1	Lisp	61
6.1.2	Prolog	62
6.1.3	OPS5	63
6.2	Summary Comparison of KBS Languages	64
6.2.1	Support of KBS Techniques	64
6.2.2	Support for User Requirements	65
A	Bibliography	66
B	Glossary	68

List of Tables

1	Example of Representation of Facts and Rules	6
2	Interpretation of Digraph	8
3	Language Implementation of Facts	20
4	If-part Pattern Matching	23
5	KB Operations	25
6	User Interface Features	30
7	Characteristics of Lists and Arrays	49
8	Files and I/O Features	52
9	Language Support for KBS Features	65

List of Figures

1	A Small Derivation Digraph	8
2	Typical Data Flow in KBS	41