

# Inhaltsverzeichnis

---

Vorwort . . . . .	iii
Inhaltsverzeichnis . . . . .	v
Zusammenfassung . . . . .	ix

## KAPITEL I

<b>Einordnung. . . . .</b>	<b>1</b>
1. Verteilte Betriebssysteme . . . . .	1
2. <i>PM</i> -Projekt . . . . .	2
3. Einordnung und Ziele. . . . .	4
3.1 Vorgehensweise . . . . .	5
3.2 Einordnung in <i>PM</i> . . . . .	6
4. Gliederung . . . . .	7

## KAPITEL II

<b>Objektmodell. . . . .</b>	<b>8</b>
1. Einführung . . . . .	8
2. <i>PM</i> -Objektmodell . . . . .	8
2.1 Aufgabenstellung . . . . .	8
2.2 Richtlinien. . . . .	9
2.3 <i>Emerald</i> -Objektmodell . . . . .	10
2.4 <i>PM</i> -Weiterentwicklungen. . . . .	11
2.4.1 Strukturmodell. . . . .	12
2.4.2 Koordinierungsmodell . . . . .	15
2.4.3 Verteilungsmodell . . . . .	16
2.5 Implementierbarkeit . . . . .	16
3. <i>Alcaid</i> -Objektmodell . . . . .	17
3.1 Warum eine neue Sprache? . . . . .	17
3.2 Richtlinien. . . . .	18
3.3 Objektmodell . . . . .	18
3.4 Dualität von Methode und Variable . . . . .	19
3.5 <i>Alcaid</i> -Programmiersprache . . . . .	20
3.5.1 Deklarationen . . . . .	21
3.5.2 Methodenaufrufe und Zuweisungen. . . . .	22
3.5.3 Verbundobjekte. . . . .	23

4. Zusammenfassung . . . . .	.23
------------------------------	-----

## KAPITEL III

<b>Konzept der Schablone . . . . .</b>	<b>.24</b>
1. Motivation . . . . .	.24
1.1 Kapselung . . . . .	.24
1.2 Klasse und Methode . . . . .	.25
1.2.1 Gemeinsamkeiten . . . . .	.25
1.2.2 Unterschiede . . . . .	.25
1.3 Ansatzpunkt . . . . .	.26
2. Schablonen . . . . .	.26
2.1 Konzept . . . . .	.26
2.2 Definitionen . . . . .	.28
2.3 Syntax . . . . .	.29
2.4 Veranschaulichung der Semantik . . . . .	.30
3. Aliase . . . . .	.31
3.1 Konzept . . . . .	.31
3.2 Syntax . . . . .	.32
3.3 Veranschaulichung der Semantik . . . . .	.34
4. Implementierbarkeit . . . . .	.36
5. Ausdrucksfähigkeit . . . . .	.38
5.1 Verschachtelte Schablonen . . . . .	.38
5.2 Funktionen . . . . .	.39
5.3 Namensräume . . . . .	.39
5.4 Kapselung der Objekterzeugung . . . . .	.40
5.5 Nebenläufigkeit . . . . .	.40
5.6 Verteilung . . . . .	.40
6. Vergleichbare Arbeiten . . . . .	.41
6.1 Objekte in <i>Self</i> . . . . .	.41
6.2 Patterns in <i>Beta</i> . . . . .	.42
6.3 Modultypen in <i>K2</i> .. . . .	.42
6.4 LNS in <i>Hydra</i> .. . . .	.43
7. Zusammenfassung . . . . .	.43

## KAPITEL IV

<b>Typen . . . . .</b>	<b>.45</b>
1. Motivation . . . . .	.45
1.1 Typprüfungen . . . . .	.45
1.2 Effizienz . . . . .	.46
1.3 Polymorphismus. . . . .	.47

1.4	Verdecken von Internas. . . . .	.48
2.	Typkonzept . . . . .	.48
2.1	Typbegriff . . . . .	.48
2.2	Trennung von Typ und Klasse . . . . .	.49
2.3	Typen in <i>Alcaid</i> . . . . .	.49
2.4	Syntax . . . . .	.50
2.5	Konformität . . . . .	.53
2.5.1	Implizite Konformität . . . . .	.54
2.5.2	Deklarierte Konformität . . . . .	.55
2.5.3	Typenerweiterung . . . . .	.56
3.	Typanpassungen . . . . .	.57
3.1	Motivation . . . . .	.57
3.2	Nachträgliche Konformität . . . . .	.60
3.2.1	Gültigkeitsbereich . . . . .	.61
3.2.2	Lokale nachträgliche Konformität . . . . .	.62
3.2.3	Begrenzte nachträgliche Konformität. . . . .	.62
3.2.4	Gültigkeitsregeln. . . . .	.63
3.3	Anpassungen . . . . .	.64
3.3.1	Typanpassung durch Umbenennung . . . . .	.64
3.3.2	Algorithmische Anpassung . . . . .	.65
3.3.3	Beispiel . . . . .	.66
4.	Implementierbarkeit . . . . .	.67
5.	Vergleichbare Arbeiten . . . . .	.68
5.1	Hölzles Typanpassungen . . . . .	.68
5.2	Typen in <i>Emerald</i> . . . . .	.69
6.	Zusammenfassung . . . . .	.69

## KAPITEL V

	<b>Vererbung . . . . .</b>	<b>.71</b>
1.	Einführung . . . . .	.71
2.	Vererbung als Grundkonzept. . . . .	.72
2.1	Grundprinzip . . . . .	.72
2.2	Erweiterungen. . . . .	.75
2.2.1	Nicht-redefinierbare Methoden . . . . .	.75
2.2.2	Mehrfachvererbung . . . . .	.76
2.2.3	Mixin-Vererbung. . . . .	.76
2.3	Probleme. . . . .	.77
2.3.1	Kontrolle über die Redefinition . . . . .	.77
2.3.2	Anfälligkeit gegen Änderungen . . . . .	.77
2.3.3	Kompositionsdilemma . . . . .	.79
3.	Vererbung in <i>Alcaid</i> . . . . .	.80

3.1	Vererbung durch Aggregation . . . . .	80
3.2	Behandlung der Erweiterungen . . . . .	82
3.3	Behandlung der Probleme . . . . .	83
3.3.1	Kontrolle über die Redefinition . . . . .	83
3.3.2	Anfälligkeit gegen Änderungen . . . . .	84
3.3.3	Kompositionsdilemma . . . . .	86
3.4	Typisierte Vererbung . . . . .	86
3.4.1	Typprüfung bei Klassenänderungen . . . . .	87
3.4.2	Zusammenfassung der Klassenparameter . . . . .	90
3.4.3	Vererbungsschnittstelle . . . . .	92
3.5	Typisierung der Selbstreferenzen . . . . .	92
3.6	Basismechanismen . . . . .	94
3.7	Verteilung . . . . .	94
4.	Vergleichbare Arbeiten . . . . .	95
5.	Zusammenfassung . . . . .	95

## KAPITEL VI

	<b>Ergebnisse und Ausblick . . . . .</b>	<b>97</b>
1.	Überblick . . . . .	97
2.	Zusammenfassung und Bewertung . . . . .	97
2.1	Vergleich mit den Zielen . . . . .	97
2.2	Vergleich mit den Richtlinien. . . . .	99
3.	Prototyp-Implementierung . . . . .	99
3.1	Aufbau . . . . .	100
3.2	Umfang . . . . .	100
3.3	Status . . . . .	101
4.	Integration in <i>PM</i> . . . . .	102
4.1	Nebenläufigkeit und Koordinierung . . . . .	102
4.2	Verteilung . . . . .	103
4.2.1	<i>Alcaid</i> und Objektbenennung . . . . .	103
4.2.2	<i>Alcaid</i> und dynamische Verteilung . . . . .	104
4.2.3	Weiterführende Arbeiten . . . . .	104
5.	Ausblick . . . . .	104
6.	Schlußbemerkung. . . . .	105
	<b>Anhang . . . . .</b>	<b>107</b>
A.1.	Syntaxbeschreibung . . . . .	107
A.2.	Übersicht der Bilder aus Kap. V . . . . .	108
	<b>Literaturverzeichnis . . . . .</b>	<b>110</b>