



W H I T E P A P E R

PARAKLET®

Framework der PARAGON Systemhaus GmbH

Inhalt

	Vorwort	3
I	Logisch, universell, neutral – die Basis von PARAKLET®	4
II	Anwendungsentwicklung auf Erfolgskurs	5
	Grundstein für Effizienz – das Software-Modellhaus	6
	Wiederverwendung mit Konzept – die Servicekomponenten	7
	Prozessoptimierung mit Qualität – der Application Modeler	8
	Automatisch strukturiert – Zusammenspiel der Servicekomponenten	9
	Konzentration auf das Wesentliche – das Mapping Tool	10
	Das Mapping Tool im Detail	11
	<i>Verknüpfung des BusinessModels</i>	11
	<i>Predications</i>	12
	<i>Tabellenspalten</i>	12
	<i>Trees</i>	12
III	Flexible Interaktion – die Laufzeitstrukturen	13
	Intelligente Kapselung – die Interaktionstypen	13
	Laufzeitstrukturen im Vergleich	14
IV	Die Leistungsmerkmale von PARAKLET® im Kurzüberblick	15
	Strukturen und Paradigmen im Großen und im Kleinen	15
	Die lokalen und konsistenten Systeme	15
	Werkzeuge, Funktionalitäten und Leistungen	15
V	Die PARAGON Systemhaus GmbH	16

Vorwort

Bei großen Bauvorhaben präsentieren Architekten ihren Kunden vor Baubeginn ein virtuelles Modell des geplanten Objekts. Während eines virtuellen Rundgangs durch das Gebäude werden gemeinsam Fragen geklärt wie: Sind Aussparungen für Türen und Fenster an den richtigen Stellen? Ist genügend Platz für die vorgesehene Inneneinrichtung? Sind Strom- und Wasserleitungen dort verlegt, wo sie sinnvoll sind, damit später alles am gewünschten Platz montiert werden kann? Sind Gebäude und einzelne Räume für die vorgesehene Nutzung überhaupt geeignet?

In der Architektur sind solche virtuellen Modelle Standard, denn sie vermeiden teure Fehler oder gar Bauruinen. Mit dem Framework PARAKLET® hat die PARAGON Systemhaus GmbH das Konzept des virtuellen Modells auf die Anwendungsentwicklung übertragen. **Lokale und konsistente Systeme** ist der Begriff dafür, und diese Software-Modellhäuser können wie ein Architekturmodell getestet, überprüft und den Anforderungen entsprechend angepasst werden. Doch gibt es zu Architekturmodellen einen feinen, aber entscheidenden Unterschied: das mit PARAKLET® erstellte Software-Modellhaus ist bereits bewohnbar und wird entlang der gesetzten Meilensteine des Software-Projekts sukzessive weiter entwickelt, um es am Ende in die Zielarchitektur zu integrieren.

Die von der PARAGON Systemhaus GmbH entwickelten lokalen und konsistenten Systeme machen die Methodik eines funktionsfähigen Software-Modellhauses in der Praxis anwendbar. **Laut Untersuchungen des Fraunhofer Instituts für angewandte Informationstechnik birgt dieses Verfahren zahlreiche Vorteile.** So kann die entstehende Software bereits zu Beginn eines Projekts auf ihre Eignung für die Umsetzung der definierten Geschäftsprozesse hin überprüft werden und bringt das Software-Projekt frühzeitig auf Erfolgskurs.

Mit dieser benutzerorientierten Vorgehensweise sowie praxisbewährten Features in PARAKLET® löst die PARAGON Systemhaus GmbH drängende Herausforderungen, vor denen die Anwendungsentwicklung im Enterprise-Bereich heute steht: zum Beispiel steigender Kostendruck sowie der Wunsch, neue Technologien und Systeme schnell und möglichst ohne Brüche zu adaptieren, um die Wettbewerbsfähigkeit zu erhalten. Mit seinen Möglichkeiten bildet PARAKLET® die optimale Basis, um diesen und anderen Herausforderungen zu begegnen, Fehler zu vermeiden und Software-Projekte im Enterprise-Bereich rasch und erfolgreich umzusetzen.

Logisch, universell, neutral – die Basis von PARAKLET®

Mit PARAKLET® lassen sich Anwendungen für nahezu jeden Architekturtyp erstellen. Das mag überraschen, doch Fakt ist: Die bei unseren Kunden umgesetzten Entwicklungen sind bis heute in den unterschiedlichsten Architekturszenarien im Einsatz, etwa als **hostzentrierte Architekturen**, als **Java EE-Architekturen** oder auch als **serviceorientierte Architekturen (Service Orientated Architecture, SOA)**. Zudem können neue Entwicklungen mit sehr viel geringerem Aufwand adaptiert und sehr viel leichter mit den bestehenden und mit PARAKLET® umgesetzten Anwendungen verbunden werden.

Möglich wird das durch ein logisches Modell, das der Struktur von PARAKLET® zugrunde liegt und das automatisch die strukturelle Basis jeder mit PARAKLET® entwickelten Anwendung bildet. Entwickler erhalten so eine einheitliche Vorstellung davon, wie die spätere Anwendung in ihrer äußeren und inneren Struktur beschaffen sein wird.

Obendrein wird durch das logische Modell ein hohes Maß an Standardisierung erreicht und auch die Wartbarkeit wird erhöht. Zusätzlich bieten mit PARAKLET® entwickelte Anwendungen definierte Ansatzpunkte für Loggingmechanismen, Performanceuntersuchungen, Laufzeitmessungen und weitere Verfahren.

Das Framework PARAKLET®

Vor acht Jahren in Zusammenarbeit mit der **HUK Coburg** entwickelt, ist PARAKLET® seitdem kontinuierlich gereift und verbessert worden. Seit mehr als fünf Jahren ist PARAKLET® bei unseren Kunden im produktiven Einsatz. Ausgangs- und Orientierungspunkt von PARAKLET® sind die Erfahrungen, die unsere hochqualifizierten Mitarbeiter in vielen Softwareprojekten gewinnen konnten. Im Ergebnis erhalten unsere Kunden ein praxiserprobtes Framework, das durch die benutzerorientierte Vorgehensweise des lokalen und konsistenten Systems Entwicklungszeiten deutlich reduzieren kann, wie eine Untersuchung des Fraunhofer Instituts für angewandte Informationstechnik belegt. Im Entwicklungsverlauf unterstützt PARAKLET® mit weiteren hilfreichen Features. Das in PARAKLET® realisierte Konzept kontextfreier Applikationskomponenten forciert die Entwicklung von wiederverwendbaren fachlichen Komponenten. Und mit dem Mapping Tool können selbst Laien komplexe deskriptive Verknüpfungen von Oberflächen mit fachlichem Code erzeugen. Aus diesen und weiteren Features ergeben sich in der Praxis zahlreiche weitere Vorteile, z. B.:

- die Entlastung der Entwickler von vielen manuellen Programmierschritten
- die Erhöhung der Entwicklungsgeschwindigkeiten
- die Möglichkeit für Entwickler, sich auf die fachlichen Zusammenhänge zu konzentrieren
- die Erhöhung der Wartbarkeit und Stabilität der erstellten Anwendungen.

Bei der HUK Coburg und weiteren Kunden trägt PARAKLET® so zum Erfolg bei.

PARAKLET® – Anwendungsentwicklung auf Erfolgskurs

Einen soliden Grundstein für ein erfolgreiches Softwareprojekt zu legen, ist kein ganz leichtes Unterfangen. Glaubt man Untersuchungen, scheitern bis zu 75 Prozent aller Projekte und das maßgeblich an den gleichen Ursachen:

- mangelnde Einbeziehung der Anwender
- ungenaue Spezifikationen und Anforderungen
- unrealistische Visionen und Erwartungen
- lange Projektverläufe ohne Meilensteine, die Zwischenergebnisse absichern.

Entwickler stoßen hier auf ein grundsätzliches Problem: herkömmliche Software-Entwicklungsprozesse sehen nicht vor, den Anwendern frühzeitig greifbare Ergebnisse zu liefern und ihnen das Erreichte klar vor Augen zu führen. Greifbare Prototypen lassen sich mit den üblichen Entwicklungswerkzeugen entweder nur mit erheblichem Aufwand oder lediglich in inkonsistenter Form erstellen. Die Ergebnisse der Anforderungsanalyse werden weitestgehend in Form von Fließtextdokumenten, PowerPoint-Präsentationen, UML-Diagrammen oder anderem dokumentiert. Für Anwender sind diese Mittel zur Dokumentation jedoch oft nicht nachvollziehbar. Im Effekt werden bei dieser Vorgehensweise falsche Erwartungen etabliert und Fehler manifestiert.

Mit PARAKLET® wird der Prozess der gemeinsamen Anforderungsanalyse wesentlich verbessert. PARAKLET® bietet die Möglichkeit, ein lokales und konsistentes System zu erstellen – ein Software-Modellhaus. Diese **wissenschaftlich fundierte Methodik** erlaubt eine qualitativ hochwertige und für Anwender anschauliche Analyse der umzusetzenden geschäftlichen Prozesse und bietet beiden Seiten – Entwicklern und Anwendern – zu jeder Zeit eine realistische Sicht auf das Ziel.

Grundstein für Effizienz – das Software-Modellhaus

Bereits mit einem ersten grundlegenden Verständnis der abzubildenden Geschäftsprozesse und der fachlichen Zusammenhänge können Entwickler ein lokales und konsistentes System erschaffen – das Software-Modellhaus. Frühzeitig lassen sich damit die Anwender in den Entwicklungsprozess mit einbeziehen. Mit dem Software-Modellhaus bewegen sich beide Seiten kommunikativ auf einer Ebene. Die Anwender können das lokale und konsistente System live testen, auf ihre spezifischen Anforderungen hin überprüfen und fundierte Aussagen über spezifische Änderungen treffen. Entwickler können das System gezielt dort bearbeiten, wo das Software-Modellhaus den definierten Prozessen und Vorstellungen der Anwender nicht entspricht. Über diesen Weg wird der Prozess der gemeinsamen Anforderungsanalyse wesentlich effizienter. Und nach unserer Erfahrung steigert dieses Konzept auch die Motivation der Beteiligten.

Sukzessive und zielgerichtet statet der Entwickler das lokale und konsistente System mit den entsprechenden Komponenten, fachlichen Zusammenhängen und Verknüpfungen aus. **Nach und nach entsteht so die für die reibungslose Funktion der definierten Geschäftsprozesse passende Anwendung.** Das anfangs erstellte Software-Modellhaus muss somit später nicht abgerissen werden. Vielmehr wird das System entlang der gesetzten Meilensteine kontinuierlich weiter entwickelt und erhält den letzten Schliff. Ist das System zum komplett ausgestatteten und in allen Details stimmigen Wunschhaus geworden, wird es in die Zielarchitektur integriert.



In einer Studie hat Prof. Dr. Volker Wulf vom Fraunhofer Institut für Angewandte Informationstechnik (FIT) im Mai 2005 Vorteile und Wirksamkeit des Konzepts der lokalen und konsistenten Systeme wissenschaftlich bestätigt.



„Das Entwicklungswerkzeug PARAKLET® bietet laut unserer Befragung von Anwendern den Vorteil, innerhalb kurzer Entwicklungszyklen lauffähige Prototypen zu erstellen, die bereits mit einem konsistenten Objektmodell verknüpft sind. Hierfür leistet PARAKLET® selbst ungeübteren Java-Entwicklern eine Unterstützung, die gegenüber herkömmlicher Entwicklung deutliche Effizienzgewinne ermöglichen konnte.“

„In unserer Befragung haben (Paragon-)Entwickler und (HUK-)Anwender angegeben, dass ihnen die Kommunikation aufgrund der praxisnahen Prototyp-Gestaltung deutlich leichter fiel als die Verständigung anhand von Papier basierten oder lediglich auf Power Point-Demonstrationen beruhenden Prototypen bzw. Prozessdiagrammen, szenischen Beschreibungen, Modellen, Anforderungslisten und Lastenheften.“

„Vor diesem Hintergrund besteht ein entscheidender Vorteil der mit PARAKLET® entwickelten, lauffähigen Prototypen in deren potentiell und testweisem Einsatz in praxisnahen Anwendungsszenarien. Dieser Einsatz verbessert die Möglichkeiten systematischer Testung durch Benutzer im Vergleich mit anderen Prototypen erheblich.“

„Es ist zu erwarten, dass der Umsetzungsschritt von einer ersten Anforderungsspezifikation zu einem lauffähigen System, das bis zum späteren Einsatz als Anwendungssystem nur noch geringfügiger Überarbeitungen bedarf, mithilfe des PARAKLET®-Werkzeugs deutlich optimiert werden kann, falls die vorgenannten Voraussetzungen gegeben sind. Damit könnten sich deutliche Kostenreduktionen durch signifikant geringere Nachbesserungsnotwendigkeiten ergeben.“



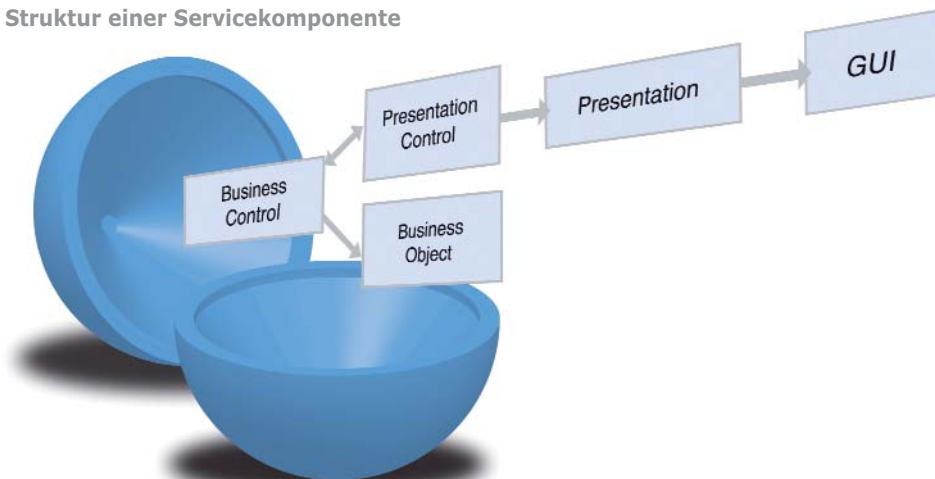
Wiederverwendung mit Konzept – die Servicekomponenten

Fachliche Komponenten möglichst wieder verwenden zu können, ist ein grundlegendes Ziel der Anwendungsentwicklung. Die Wiederverwendbarkeit von einmal entwickelten fachlichen Komponenten verkürzt Entwicklungszeiten, ermöglicht **spätere Änderungen ohne großen Aufwand** und macht Anwendungen zudem flexibler und leichter wartbar. Voraussetzung hierfür sind kontextfreie Komponenten im Sinne der serviceorientierten Architekturen (SOA), die ihre definierten Aufgaben auch in unterschiedlichen Zusammenhängen erledigen können. PARAKLET® forciert die Entwicklung von solch wieder verwendbaren, fachlichen Komponenten mit dem von der PARAGON Systemhaus GmbH realisierten Konzept kontextfreier Servicekomponenten.

Zu jedem fachlichen Vorgang wird mit PARAKLET® eine Servicekomponente erstellt. Jede der separat entwickelten Servicekomponenten erledigt dabei klar umrissene und **kontextfreie Aufgaben**. Die einmal erstellten Servicekomponenten können ihre definierten Funktionen somit auch in unterschiedlichen Zusammenhängen ausführen und müssen nicht neu erstellt werden. Nach und nach werden die einzelnen Servicekomponenten zur Gesamtanwendung zusammengefügt. Für die ressourcensparende Wiederverwendung von fachlichen Komponenten ist dieses Konzept ideal.

Jede Servicekomponente setzt sich aus den Bestandteilen GUI (z. B. Swing, AWT, SWT, JSF, JSP), PresentationControl (PC), BusinessControl (BC) und den BusinessObjects (BO) zusammen. Das BusinessControl repräsentiert einen fachlich abgeschlossenen Prozess, zum Beispiel eine Suchfunktion. Das PresentationControl übernimmt Steuerungsaufgaben bzgl. der Oberfläche. Die BusinessObjects stellen einen Ausschnitt aus dem Daten-/Objektmodell dar.

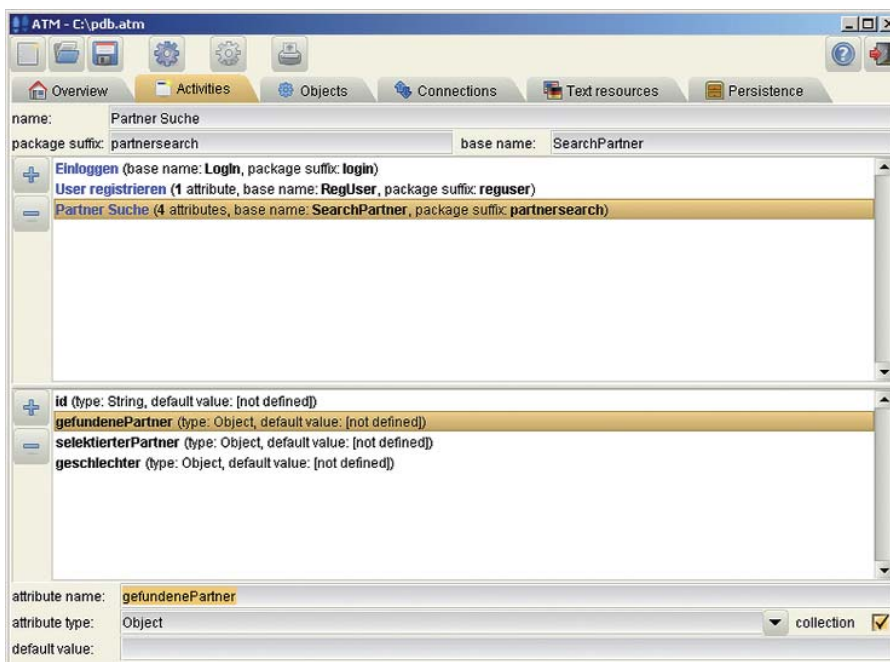
Struktur einer Servicekomponente



Prozessoptimierung mit Qualitäten – der Application Modeler

Das automatisierte Generieren von benötigtem Programmcode ist ein wesentliches Merkmal technologisch führender Software-Entwicklungsumgebungen. Der Entwickler wird von zeitintensiven Pflichtübungen befreit und gleichzeitig wird durch den Wegfall von Fehlerquellen die Qualität des Programmcodes erhöht. PARAKLET® gibt Entwicklern mit dem Application Modeler ein Werkzeug an die Hand, das Programmcode zuverlässig und schnell automatisiert erstellt. Dabei fungiert der Application Modeler gleichzeitig als offene Integrationsplattform. So lassen sich die Ergebnisse unterschiedlicher Methoden aus dem Bereich der Prozessmodellierung über den Application Modeler einbinden und die definierten Prozesse ebenfalls automatisiert in den benötigten Programmcode umwandeln.

Application Modeler



Konkret kann der Entwickler mit dem Application Modeler

- ein ausführbares Anwendungsgerüst bestehend aus Servicekomponenten,
- ein Datenmodell,
- ein objekt-relationales Mapping,
- Definitionen für die Internationalisierung und anderes generieren.

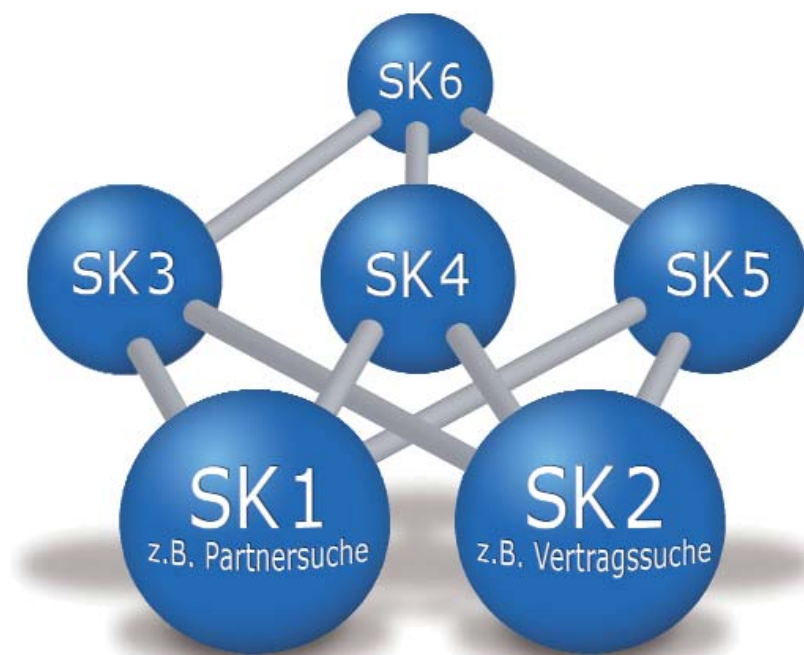
Automatisch strukturiert – Zusammenspiel der Servicekomponenten

Dem Application Modeler liegen übergeordnete Modelle im Sinne der Model Driven Architecture (MDA) zugrunde. Dabei kann der Application Modeler als offene Integrationsplattform auch Ergebnisse anderer und beim Kunden bereits etablierter Modellierungsmethoden verarbeiten. Ebenso lässt sich der Application Modeler um individuelle Werkzeuge erweitern, die dem jeweiligen Kundenumfeld entsprechend konfiguriert werden können. Im Ergebnis erstellt der Application Modeler aus beliebigen Prozessdefinitionen automatisiert Programmcode.

Die Grafik unten zeigt beispielhaft, wie eine mit dem Application Modeler automatisch generierte Struktur aus sechs kollaborierenden Servicekomponenten die notwendigen Funktionalitäten für eine zuvor definierte Anwendung zur Verfügung stellt.

Aufgrund der weit reichenden Werkzeugunterstützung kann sich der Entwickler nun ganz auf die nächsten Schritte konzentrieren - die Ergänzung der erzeugten Servicekomponenten um die fachlichen Programmteile, die Vervollständigung des Daten-/Objektmodells und die Verknüpfung der Oberfläche mit den einzelnen fachlichen Programmteilen. Auch hierzu steht ein hilfreiches und zum Patent angemeldetes Werkzeug zur Verfügung: das Mapping Tool.

Struktur kollaborierender Servicekomponenten



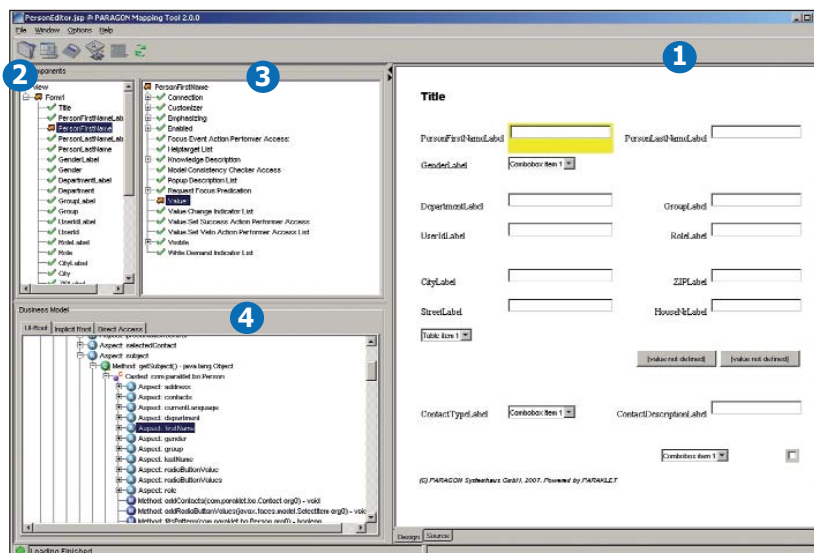
Konzentration auf das Wesentliche – das Mapping Tool

Das Mapping Tool ist der zentrale Bestandteil von PARAKLET® für die Entwicklungsarbeit. Das Besondere: Mit dem Mapping Tool werden alle für die Abbildung der definierten Geschäftsprozesse notwendigen Verknüpfungen und Abhängigkeiten über eine **intuitiv bedienbare grafische Oberfläche** erstellt. Gerade auch ungeübteren Java-Entwicklern hilft das Mapping Tool, komplexe Anforderungen eines Softwareprojekts umzusetzen, sich dabei auf das Wesentliche zu konzentrieren und Fehler zu vermeiden.

Im Mapping Tool werden spezifische Eigenschaften der Oberflächenelemente mit Teilen der Applikationskomponente (z. B. PresentationControl, BusinessControl, BusinessObjects) verknüpft. Auch dient das Mapping Tool zur Steuerung von weiteren Eigenschaften der Oberflächenelemente wie Sichtbarkeit, Verfügbarkeit u.a. Hierzu können die Oberflächenelemente mit beliebigen fachlichen Programmteilen verknüpft werden.

Prinzipiell können mit dem Mapping Tool selbst Laien arbeiten, denn es ist so konzipiert, dass **kein spezifisches Wissen über Technologien notwendig** ist. Entwickler können somit unabhängig von der jeweils eingesetzten Technologie wie z. B. WEB 2.0 (JSF- ICEFaces, Tobago u.a.), Swing, AWT, SWT u.a. immer in der gleichen Art und Weise arbeiten und müssen nicht zwischen verschiedenen Werkzeugen wechseln oder sich auf neue Arbeitsweisen einstellen.

Mapping Tool



Das Mapping Tool besteht aus insgesamt vier Bereichen. Im rechten Bereich (**1 UI**) wird die zu verknüpfende Oberfläche grafisch angezeigt. Im linken oberen Bereich (**2 UIComponents**) lässt sich der strukturelle Aufbau der zu verknüpfenden Oberfläche bis hinunter zu den einzelnen Elementen wie z. B. einem Textfeld ansehen. Im mittleren oberen Teil (**3 UIProperties**) werden die verknüpfbaren Eigenschaften des in **1** oder **2** ausgewählten Oberflächenelements dargestellt. Im linken unteren Teil (**4 Business Model**) wird das BusinessModel gezeigt, das der Anwendung zugrunde liegt.

Entwickler haben die Möglichkeit, das Mapping Tool entsprechend ihres jeweiligen Kenntnisstands und ihrer spezifischen Anforderungen, die sich aus der jeweiligen Aufgabenstellung des Projekts ergeben, zu konfigurieren. Hierdurch behalten Entwickler die für ihre Arbeit jeweils **relevanten Aspekte stets im Blick**.

Das Mapping Tool im Detail

In seinen Funktionen bietet das Mapping Tool zahlreiche Möglichkeiten, die komplexen Anforderungen aus der Projektarbeit einfach umzusetzen. Exemplarisch sind hier vier aufgeführt.

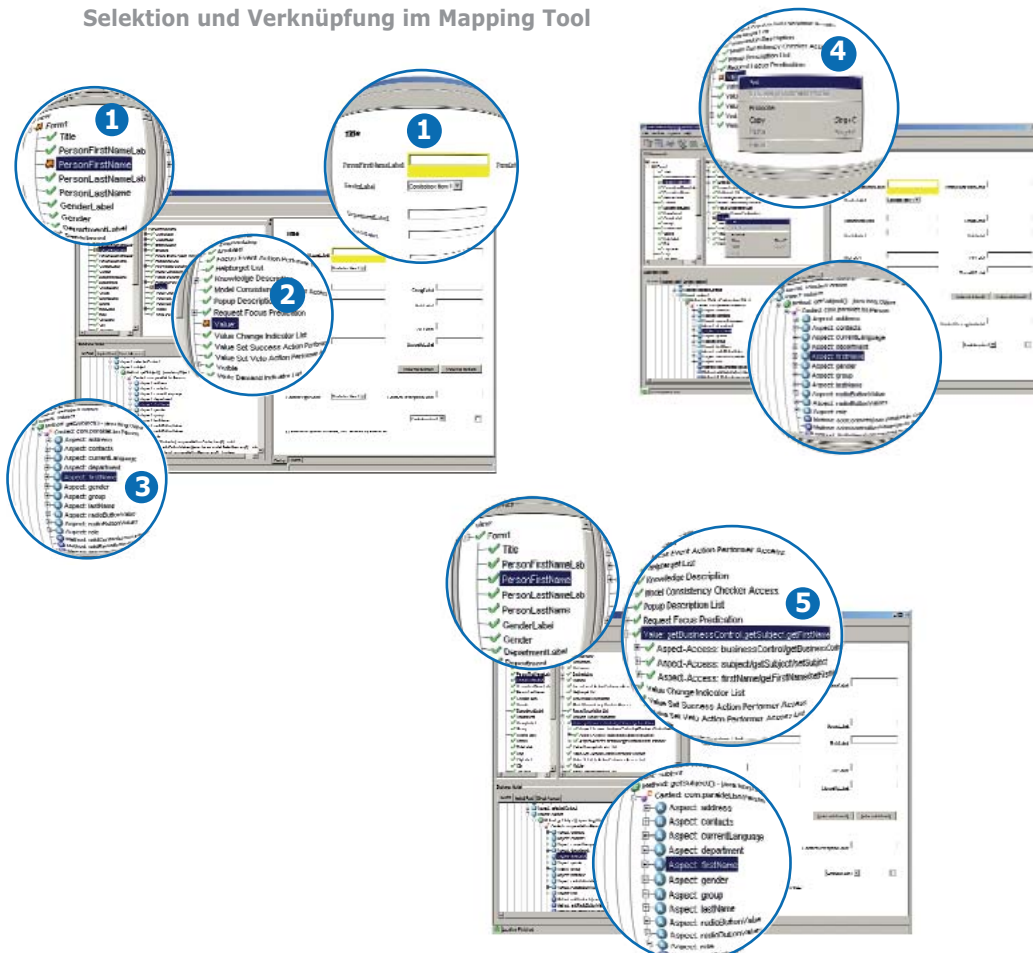
1. Verknüpfung eines Oberflächenelements mit einem Element des BusinessModels

Für die Verknüpfung mit dem BusinessModel stehen verschiedene, flexibel konfigurierbare Möglichkeiten zur Verfügung. Im Regelfall führt man die nachfolgenden Arbeitsschritte aus:

- Auswahl des zu verknüpfenden Oberflächenelements in UI oder UIComponents (1 PersonFirstName)
- Auswahl der zu verknüpfenden Eigenschaft des Oberflächenelements (2 Value)
- Auswahl des Wertes/Aspekts im BusinessModel, der angezeigt werden soll (3 Aspect: firstName)
- Erstellung der Verknüpfung über das Popup-Menu (4)
- Die Verknüpfung ist erstellt und wird in den UIProperties mit einem grünen Haken angezeigt (5)

Zum Tragen kommen diese Verfahrensweisen etwa bei der Verknüpfung von Aktionen, Selektionen und Anzeigeaspekten von Spalteneinträgen, bei der Spezifikation von Stellvertreterobjekten für boolesche Werte oder auch bei der Definition von Konvertoren für die Übersetzung von Werten.

Selektion und Verknüpfung im Mapping Tool



2. Predications

Mit Predications (logischen Aussagen) werden in PARAKLET® unter anderem die Eigenschaften von Oberflächenelementen gesteuert, z. B.

- deren generelle Verfügbarkeit (enabled oder disabled),
- die Sichtbarkeit eines Oberflächenelements,
- die Eigenschaft, ein Mussfeld oder ein Defaultbutton zu sein, und
- die Eigenschaft, wann es den Fokus an sich ziehen soll.

Dem Entwickler steht neben den elementaren Predications „Und“, „Oder“, „Nicht“, „Wahr“ und „Falsch“ auch eine Predication „wertabhängig“ zur Verfügung. Mit dieser Grundmenge an Predications sowie der Möglichkeit, Predications miteinander zu kombinieren, können alle in einem Projekt auftretenden Konstellationen leicht abgebildet werden.

3. Tabellenspalten

Das Verhalten einer Tabellenspalte wird in PARAKLET® mit ColumnDescriptions beschrieben. Die Zuordnung der Definitionen von ColumnDescriptions geschieht wie in 1. und 2. beschrieben.

4. Trees

Für die Verknüpfung von Trees werden im Mapping Tool Knotenerzeuger definiert.

Der für eine spezifische Konstellation zuständige Knotenerzeuger wird wiederum mittels einer Predication ermittelt und aktiviert.

Zusätzlich werden Aspekte wie Darstellungsform, Iconzugriff, Expansionselementzugriff und Knotenexpansionssteuerung definiert.

Flexible Interaktion – die Laufzeitstrukturen

Die im Mapping Tool erzeugten Informationen, die in Bezug zu den Laufzeitstrukturen stehen, werden in PARAKLET® in deskriptiver, sprich beschreibender Form abgelegt. Der entstehende Vorteil: selbst massive Technologiegrenzen werden überwunden, denn aus den deskriptiven Informationen können Laufzeitstrukturen für unterschiedliche Plattformen abgeleitet werden.

Im Ergebnis wird die Entwicklungsarbeit sehr viel einfacher und flexibler. So kann zum Beispiel eine Oberfläche nachträglich etwa in ihrem Aussehen moderner gestaltet werden, ohne darunter liegende Anwendung verändern zu müssen. Auch lassen sich aus den selben deskriptiven Informationen Laufzeitstrukturen für eine Web- oder für eine herkömmliche Anwendung ableiten

Intelligente Kapselung – die Interaktionstypen

PARAKLET® benutzt in den Laufzeitstrukturen so genannte „Interaktionstypen“. Durch die Interaktionstypen werden die jeweils verwendeten Oberflächentechnologien von der darunter liegenden Anwendung vollständig entkoppelt. So ist es sogar möglich, ein und die selbe Anwendung ohne Änderungen als Web- oder als herkömmlichen Client zu betreiben. Die Oberflächentechnologien, wie z. B. WEB 2.0 (JSF- ICEFaces, Tobago u.a.), Swing, AWT, SWT u.a., werden dazu einfach gegeneinander ausgetauscht.

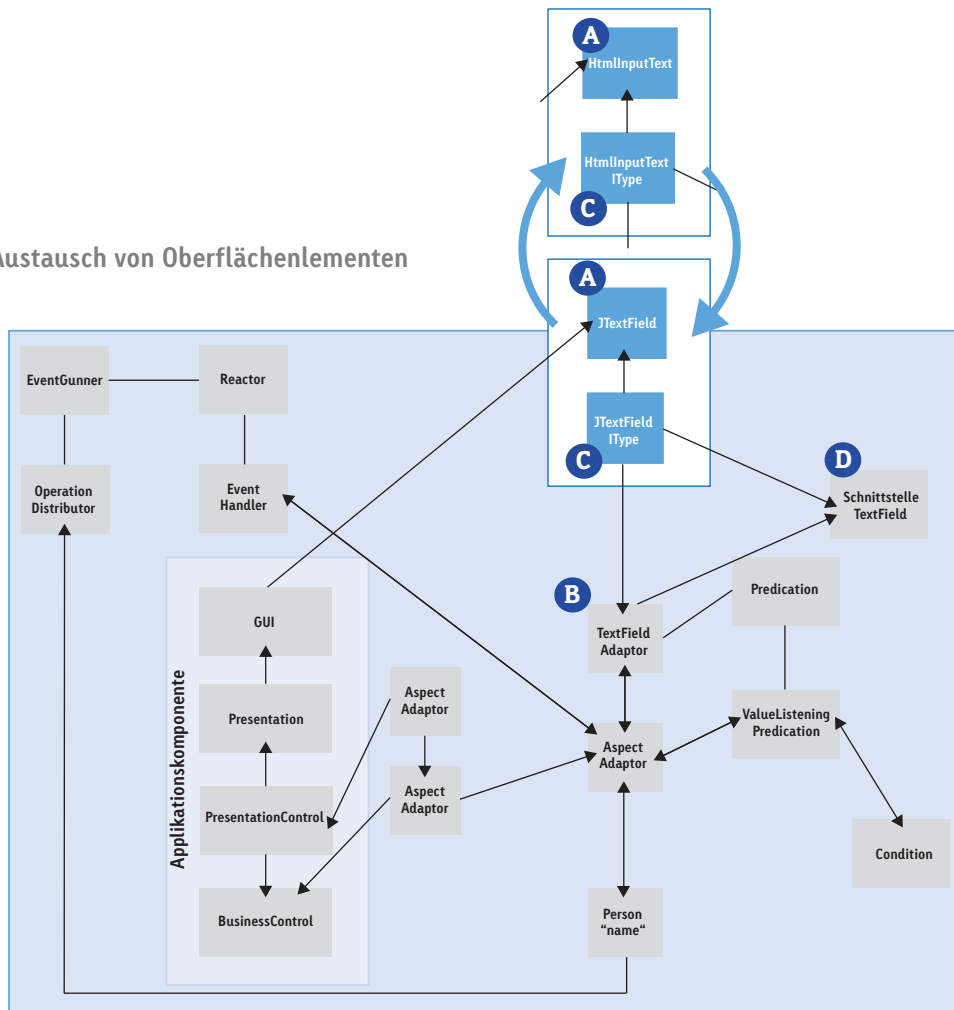
Laufzeitstrukturen im Vergleich

Zur Verdeutlichung der Konzepte, Strukturen und Möglichkeiten der Interaktionstypen wird beispielhaft der Aufbau der Laufzeitstrukturen eines Web (JSF)- und eines herkömmlichen Clients (Swing) dargestellt und erläutert.

Während sich die Oberflächenbibliotheken JSF und Swing erheblich voneinander unterscheiden, sind die Unterschiede in der von PARAKLET® aufgebauten Laufzeitstruktur marginal. Das jeweilige Oberflächenelement (**A**: JTextField und HtmlInputText) wird durch einen TextFieldAdaptor (**B**) bzgl. der möglichen Eigenschaften (z. B. Inhalt, Verfügbarkeit) gesteuert. Der TextFieldAdaptor kommuniziert mit den Interaktionstypen JTextFieldIType (**C**) bzw. HtmlInputTextIType (**C**) über die implementierte Schnittstelle TextField (**D**). Durch die Kapselung geht der TextFieldAdaptor jedoch nur scheinbar mit der Schnittstelle TextField um, in Wirklichkeit jedoch mit dem eigens dafür implementierten Interaktionstyp. Die spezifischen Interaktionstypen erkennen dabei die konkrete Implementierung des Textfeldes und sind in der Lage, sie gemäß ihrer Schnittstelle richtig zu behandeln.

An diesem Beispiel lassen sich weitere etablierte Designpatterns erkennen, mit denen PARAKLET® arbeitet, z. B. Reactor, AspektAdaptoren und Condition.

Austausch von Oberflächenelementen



Die Leistungsmerkmale von PARAKLET® im Kurzüberblick

Strukturen und Paradigmen im Großen und im Kleinen

- logisches, an Schichten orientiertes Java-Framework für die Entwicklung jeder Art von Anwendungen (Model View Controller Entwurfsmuster)
- optimale Basis für die Erstellung von kontextfreien Komponenten
- hohe Wiederverwendbarkeit von fachlichem Code in verschiedenen Zusammenhängen
- höhere Standardisierung, gute Wartbarkeit, hohe Flexibilität und einfache Integrierbarkeit der entwickelten Anwendungen durch das Benutzen von Entwurfsmustern auf allen Ebenen (MVC, Interaktionstypen, Reactor, Value Models u.a.)
- hohe Adaptivität neuer Technologien
- logische und technische Architekturneutralität
- Konformität zu den Anforderungen an serviceorientierte Architekturen (SOA) und einzigartige Möglichkeiten des Logging, Monitoring und Debugging
- einheitliches und zentrales Event-Handling

Die lokalen und konsistenten Systeme

- Erhöhung der Erfolgsaussichten eines Softwareprojekts durch das Konzept der Software-Modellhäuser
- frühes Einbeziehen der Anwender ermöglicht ein enges und transparentes Controlling des Projektfortschritts und die konkrete Absicherung der Anforderungen
- unrealistische Erwartungen werden frühzeitig ausgeräumt
- zentraler Bestandteil der Software-Entwicklung mit PARAKLET®, der in ein dem Kundenumfeld angemessenes Vorgehensmodell integriert werden kann

Werkzeuge, Funktionalitäten und Leistungen

- Vereinfachung der fachlichen Arbeit durch intuitive Bedienung des zentralen Mapping Tools und weiterer Werkzeuge
- Unterstützung jeglicher Art von Clients: WEB 2.0 (JSF-ICEFaces, Tobago u.a.), Swing, AWT, SWT u.a.
- optimierte Performance und optimierter Speicherbedarf durch weit reichende Caching-Verfahren
- unterstützende Tools für den gesamten Entwicklungsprozess
- Realisierung von Aspekten des Konzepts der Model Driven Architecture (MDA)
- umfangreiche Basisklassenbibliothek
- spezielle Möglichkeiten zur 3270-Emulations-Integration

PARAGON Systemhaus GmbH

Die PARAGON Systemhaus GmbH betreut Unternehmen kompetent bei allen Projekten der Informationstechnologie. Bei unseren Dienstleistungen haben wir uns auf die Beratung, Konzeption, Realisierung und Integration von Enterprise-Systemen im Umfeld objektorientierter Anwendungslandschaften spezialisiert.

Uns steht ein Team aus erfahrenen und hoch qualifizierten Fachkräften zur Verfügung, die seit Anfang der neunziger Jahre zahlreiche Projekte in mittelständischen Betrieben und Großunternehmen erfolgreich abgeschlossen haben. Regelmäßige Weiterbildungen, Tests und Zertifizierungen unserer Mitarbeiter durch unabhängige Unternehmen gewährleisten beständig hohe Qualität und Kompetenz.

Unsere Mitarbeiter bewegen sich in unterschiedlichen Architekturen (Java EE, serviceorientierte oder auch hostzentrierte Architekturen) und arbeiten mit Vorgehensmodellen, die dem individuellen Umfeld des Kunden angemessen sind.

Bevorzugt beraten wir unsere Kunden im Java-Umfeld. Dies beinhaltet Beratungsleistungen von Open Source-Produkten (wie z. B. JBoss) bis hin zu proprietären Enterprise-Produkten (z. B. SAP NetWeaver).

Mit einem breiten Spektrum an Seminaren halten unsere Experten Kunden auf dem neuesten Wissensstand. Detailliert und praxisnah werden neueste Strategien und Technologien vermittelt. Auf Wunsch führen wir Seminare Inhouse durch.

Wir hoffen, dass wir Ihnen mit diesem White Paper einen ersten Eindruck der grundlegenden Vorteile und Möglichkeiten des Frameworks PARAKLET® geben konnten. Ihre Fragen beantworten wir gern.

Kontakt:

PARAGON Systemhaus GmbH
Hermann-Heinrich-Gossen-Str. 3
50858 Köln
Telefon: 02234 9799 400
Fax: 02234 9799 429
Internet: www.paragon-systemhaus.de
E-Mail: info@paragon-systemhaus.de