Institut für Systemsoftware
O.Univ.-Prof. Dr. Hanspeter Mössenböck

JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Beispielprogramm Kundenbeziehungsmanagement "Plux-CRM" für die Plugin-Plattform Plux.NET

Diplomaufgabe für Sabine Weiss
Matrikelnummer: 0456241
EMail: weiss.sabine@gmx.at

Das Ziel bei Plux.NET sind Programme deren Funktionsumfang sich für jeden Anwender und jede Aufgabe anpassen lassen. Plux-CRM soll am Beispiel Customer Relationship Management zeigen, wie man Plux-Programme entwerfen soll. Plux-CRM soll als Referenzimplementierung im Quellcode veröffentlicht werden. Da Entwickler den Quellcode zum Erlernen der Plux Programmierung verwenden sollen, sind klare Struktur und Lesbarkeit wichtige Ziele der Implementierung. Die Implementierung von Plux-CRM soll zeigen...

- wie man Programme in Bausteine (*Extensions*) zerlegt
- wie man *Slots* und *Plugs* verwendet um diese Bausteine mit der Laufzeitumgebung von Plux.NET zu einem Programm zusammenzustecken
- wie man *Slots* und *Plugs* konfiguriert damit Bausteine zur Laufzeit ein-, aus- und umgesteckt werden können um so das Programm umzukonfigurieren
- wie man mit den Steuerelementen der Plux-Bibliothek Benutzeroberflächen baut, die sich dynamisch an geänderte Bausteinkonfigurationen anpassen
- wie man mit speziellen Sicherheits-Slots der Plux-Bibliothek steuert wo Bausteine von Dritten angesteckt werden dürfen und welche Rechte diese Bausteine haben

Funktionsumfang

In dieser Arbeit sollen die wesentlichen Funktionen für das Management von Kundenbeziehungen entwickelt werden. Da Plux-Programme von Grund auf erweiterbar sind, können spätere Arbeiten den Funktionsumfang schrittweise ausbauen. Die Kernbausteine eines CRM-Programms sind:

- Kontakte verwalten Firmen, Personen und zugehörige Kontaktdaten wie Anschrift, Telefonnummer, EMail

- ein Journal zeigt zu jeder Firma oder Person historische Notizen zu Telefonaten und Besprechungen, den EMail-Verkehr sowie Dokumente
- mit Aufgaben plant der Benutzer eigene Aufgaben, oder delegiert Aufgaben an andere Benutzer; Fälligkeitstermine und Erinnerungen unterstützen beim Erledigen der Aufgaben
- ein "schwarzes Brett" zeigt chronologisch alle Aktivitäten eines Benutzers
- Projekte fassen Kontakte, Notizen, Dokumente und Personen auf einer Übersicht zusammen; Projekte werden ausgewählten Benutzern zugeteilt
- die Benutzerverwaltung legt fest welche Funktionen ein Benutzer verwendet und an welchen Projekten er beteiligt ist

Implementierungshinweise

Die Benutzerschnittstelle von Plux-CRM soll einfach und übersichtlich sein. Jeder Benutzer verwendet nur einen kleinen Teil der Funktionen und das Programm kann für jeden Benutzer individuell konfiguriert werden. So kann auf umfangreiche Menüleisten mit zahlreichen Funktionen verzichtet werden. Der verfügbare Bildschirmplatz soll primär zur Präsentation der Daten verwendet werden. Zur Auswahl von Funktionen reichen minimale Steuerelemente aus.

Alle Steuerelemente in der Benutzerschnittstelle sollen sich dynamisch an Konfigurationsänderungen anpassen. Einige solcher dynamischer Steuerelemente sind in der Plux-Bibliothek bereits vorhanden und können verwendet werden. Neu zu erstellende Steuerelemente sollen in gleicher Weise auf Änderungen reagieren. Wird ein neues Steuerelement in Plux-CRM häufig verwendet, soll das Steuerelement allgemein implementiert und in die Plux-Bibliothek aufgenommen werden.

Ein Prinzip von Plux-Programmen ist, dass Benutzerinteraktion durch An- oder Abstecken von Bausteinen umgesetzt wird. Wenn ein Benutzer z.B. auf eine Schaltfläche klickt, wird diese Funktion so umgesetzt, dass eine *Extension* in einen *Slot* eingesteckt wird. Auf diese Weise kann jede Funktion leicht durch eine andere *Extension* ersetzt werden. Außerdem kann damit jede Funktion auch über den Skript-Interpreter der Plux-Werkzeuge ausgeführt werden.

Für die Datenhaltung soll keine SQL-Datenbank oder Bibliotheken von Drittherstellern verwendet werden. Wenn Plux-CRM künftig als Referenzbeispiel von der Plux-Webseite geladen wird, sollen keine externen Bibliotheken oder Programme installiert werden müssen. Ziel ist, dass jeder, der das .NET Framework installiert hat, Plux-CRM ausführen kann. Es reicht also aus, Daten im Hauptspeicher zu halten und zum Programmende

in eine Datei zu serialisieren. Die Datenzugriffsschicht soll aber austauschbar sein, damit man später die dateibasierte Datenhaltung gegen eine Datenbank austauschen kann.

Plux-CRM ist mit Visual Studio 2005 in C# und mit Windows Forms 2.0 zu implementieren. Zusätzliche Bibliotheken sollen nur dann verwendet werden, wenn es sich um frei verfügbare Software handelt, der Quellcode frei verfügbar ist und keine Installationsprogramme ausgeführt werden müssen.

Nähere Auskünfte: Mag. Reinhard Wolfinger
Beginn: Oktober 2008