



Ricardo Alvarez

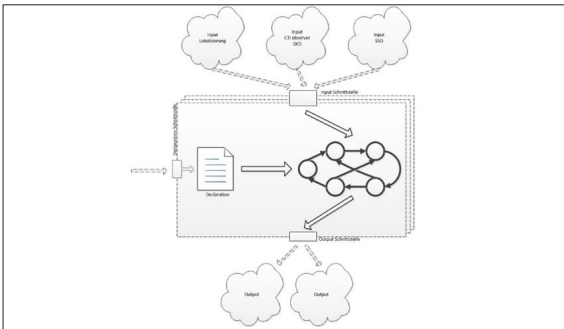


Reto Brühwiler

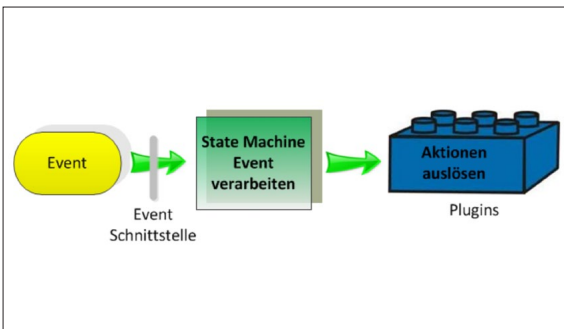
Diplomanden	Ricardo Alvarez, Reto Brühwiler
Examinator	Prof. Beat Stettler
Experte	Remo Gloor, bbv IT Solutions, Zug
Themengebiet	Internet-Technologien und -Anwendungen
Projektpartner	bbv IT Solutions, Zug

Presence Engine im Spitalumfeld

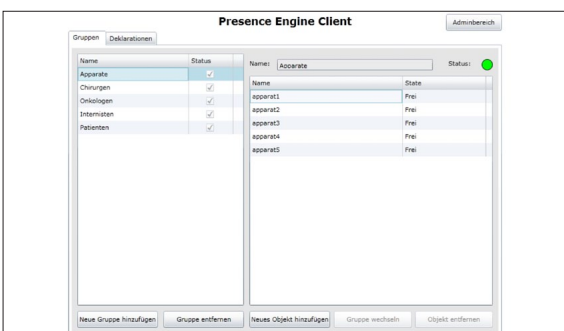
Mandantenfähige Presence Engine



Problem domain



Eventablauf



GUI-Startseite

Ausgangslage: Der Spitalalltag besteht für die Angestellten zu einem beträchtlichen Anteil aus dem Suchen nach freien Geräten, Ärzten oder Patienten. Als Abhilfe dagegen werden zunehmend Lokalisierungstechnologien eingesetzt. Allerdings hilft es nicht, ein Gerät zu lokalisieren, das nicht verfügbar ist. Neben dem Standort ist also auch der Zustand des Objektes von Interesse. Bei Menschen sind zusätzlich deren Fähigkeiten wichtig, da es beispielsweise wenig nützt, eine freie Person mit den falschen Fähigkeiten zu alarmieren. Von dieser Situation ausgehend war es unser Ziel, eine flexible State Machine zu programmieren, welche verschiedene Zustände von Personen oder Objekten verwalten kann. Zustandsübergänge sollen von Fremdsystemen automatisch durch Ereignisnachrichten oder manuell ausgelöst werden können. Zusätzlich müssen bei Übergängen von einem Zustand zu einem anderen auch Aktionen wie z. B. eine Alarmierung ausgeführt werden können. Ausserdem sollte das System nach der Inbetriebnahme mit weiteren State Machines oder mit neuen Aktionen ergänzt werden können.

Vorgehen/Technologien: Zuerst musste eine geeignete Technologie evaluiert werden, die die hohen Anforderungen erfüllen kann. Die State Machine als zentrales Element der Applikation war ausschlaggebend für die Wahl des .Net Framework. Da die Applikation Hunderttausende von Objekten verwalten können soll, musste zuerst mit Hilfe von Prototypen und Tests nach einer geeigneten Architektur gesucht werden. Um die State Machines flexibel erweitern zu können, werden diese Objekte deklarativ in einer Datei beschrieben. Für die Erweiterbarkeit der Aktionen wurde eine Plugin-Infrastruktur gewählt. So können im laufenden Betrieb weitere Funktionen hinzugefügt werden.

Ergebnis: Die erstellte Applikation besteht aus zwei Teilen. Zum einen aus der Presence Engine, welche mit Hilfe der State Machines die Verarbeitung der Ereignisnachrichten und die Verwaltung der Zustände ausführt. Diese läuft selbstständig, verwaltet Objekte und löst Aktionen aus. Zudem wurde eine grafische Oberfläche erstellt, welche die Verwaltung des Systems erlaubt. Diese Oberfläche beinhaltet das Erstellen und Manipulieren neuer State-Machine-Beschreibungen, das Erfassen und Löschen der Objekte und zeigt dem Anwender überdies die aktuellen States der Objekte an.