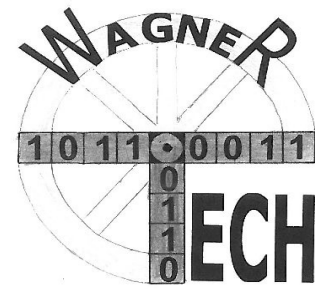


Generische Administration

WagnerTech UG, Turfstr. 18a, 81929 München, www.wagnertech.de



Viele Anwendungen haben Stammdaten mit geringer Änderungsrate, für die die Erstellung einer eigenen GUI nicht wirtschaftlich ist. Um dem Anwender dennoch eine komfortable Möglichkeit der Pflege dieser Daten bieten zu können, wurde mit der *Generischen Administration* eine Infrastruktur geschaffen, die allein auf konfigurativem Wege eine derartige Funktionalität bereitstellt. Dabei wird durch den GenAd-Kern das interne durch einen OR-Mapper auf einer Datenbank abgebildete Modell in ein externes ER-Modell gewandelt. An der GUI werden die Entitäten gepflegt.

Viele Anwendungen haben Stammdaten, die nur selten verändert werden müssen. Die Pflege dieser Daten rechtfertigt nur selten die Erstellung einer eigenen graphischen Oberfläche. Andererseits stellt das Nicht-Vorhanden-Sein einer solchen Oberfläche einen Nachteil in der Benutzerfreundlichkeit eines solchen Produkts dar.

Die hier vorgestellte *Generische Administration*, kurz GenAd ermöglicht es, allein auf konfigurativem Wege eine einfache GUI zu erstellen. Abbildung 1 zeigt die Bestandteile der *Generischen Administration*.

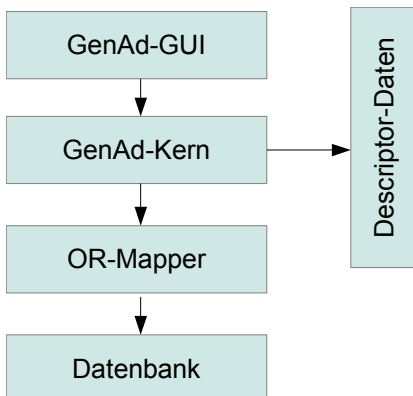


Abbildung 1: Systemaufbau

Der OR-Mapper¹ der verwendeten Programmiersprache setzt das relationale Modell der Datenbank in Objekte der Programmiersprache um. Der GenAd-Kern wiederum setzt dieses Modell über die Abbildungsvorschriften, die in den Descriptor Daten hinterlegt sind, in das extern sichtbare ER-Modell² um. Für das externe Modell werden folgende Eigenschaften unterstützt:

- 1:1-Relationen zwischen Entitäten
- 1:n-Relationen zwischen Entitäten
- n:m-Relationen zwischen Entitäten
- Vererbungsrelationen
- Strukturierte Attribute
- Unterstützung von Enumerationen und Boolean

Für die Überführung strukturierter Attribute in das Datenbankmodell werden in den Descriptor Daten entsprechende Umwandlungsklassen hinterlegt.

Die Schnittstelle des GenAd-Kerns bietet neben den CRUD³-Methoden auf den Entitäten des externen Modells reflexive Methoden zur Exploration dieses

Modells an. Der Kern kann auch unabhängig von der GenAd-GUI in Anwendungen mit nativer GUI verwendet werden.

Die GenAd-GUI ist gemäß den Model-View-Controll-Muster aufgebaut und stellt sich selbst als Bibliothek dar, die in die GUI der Anwendung integriert wird. Zur Darstellung der Entitäten werden eine Listenansicht (Abbildung 2) und eine Entitätsansicht (Abbildung 3) angeboten.

name	host_addr	local
LOALMGHO	host_name/localhost	true
IPv4/2.3.4.5	IPv4/2.3.4.5	false

Abbildung 2: Listenansicht einer einfachen Entität

name	STALFWIN
admin_state	unlocked
evt_sel_mode	exclude

Abbildung 3: Entitätsansicht einer Entität mit zwei abhängigen Listen zu anderen Entitäten

Die Namen von Entitätsklassen sowie von Attributen können frei vergeben werden. Damit ist die GenAd-GUI auch internationalisierbar.

Bisher wurde GenAd für den LAMP⁴-Stack realisiert. GenAd lässt sich einfach in andere Programmiersprachen portieren, sofern für diese ein OR-Mapper verfügbar ist.

Eingesetzt wird GenAd aktuell in zwei Produkten: Administration der Basisentitäten von System Alarming und Pflege der Lohn-Basisdaten eines Lohnabrechnungsprogramms.

1 Objekt-Relationaler Mapper
2 Entity-Relationship-Modell
3 Create, Read, Update, Delete

4 Linux, Apache, Mysql, PHP