# Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

# Webapplicatie-generatie

## *Web-app-technologie en geautomatiseerde systeemontwikkeling*

*Door: drs. ing. Arnoud van Bers.*
*Met medewerking van: Ir. Eddy Luursema, Misja Nabben, beiden Hogeschool van Arnhem en Nijmegen, lectoraat Model Based Information Systems.*
*Trefwoorden: web technologie, mobile apps, generatieve systeemontwikkeling.*

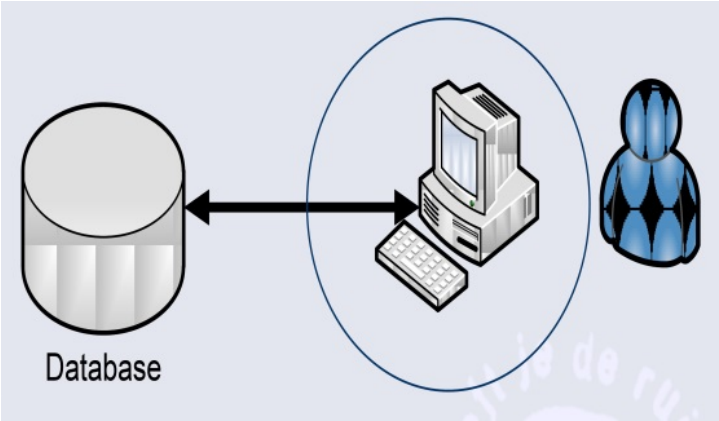
**De ontwikkeling en realisatie van mobile-(web)apps is ontegenzeggelijk een onderwerp dat in het beroepenveld en het onderwijs van de ICT/media-opleidingen snel een belangrijke positie verwerft. De nieuwe versie van de minor Webtechnologie staat in de steigers. Voortbouwend op de onderzoeksresultaten van het Lectoraat Model Based Information Systems zijn in prototypen en pilots webapps gerealiseerd. Daarbij worden de methoden en technieken van geautomatiseerde systeemontwikkeling nu ook op mobile-webapps toegepast.**
De bijdrage presenteerde de resultaten en voortgaande ontwikkeling van de generatieve methoden en technieken en demonstreerde de realisatie van mobile-apps. De presentatie introduceert de kenniskring Model-Based Information Systems en richt de aandacht op data intensieve systemen. Na de beschrijving van de globale requirements voor systeemontwikkeling komen de technieken voor applicatiegeneratie aan bod en wordt de ontwikkeling en realisatie van een applicatiegenerator beschreven. In de conclusies wordt de state of the art en de blik op toekomstige realisaties beschreven

### Inleiding

De Informatica Communicatie Academie (ICA) is het ICT en Media instituut van de HAN University of Applied Sciences. Een onderzoeksgroep in het lectoraat Model Based Information Systems (MBIS) bestaat uit docenten van de ICT bachelors opleidingen en de Master opleiding Information Systems Development (Ms ISD). De expertise van de onderzoekers ligt op de vakgebieden: Informatie modellering, Databases en Webtechnologie en ze ontwikkelen het onderwijs in Entity Relation, FCO-Information Modeling, Business Intelligence (Data Vault en Anchormodeling) en Ontwikkeling van Informatie Systemen door applicatiegeneratie. Sinds 1995 zijn tools ontwikkeld waarmee de informatiemodellering en applicatiegeneratie voor het onderwijs en voor de concrete realisatie van systemen wordt gefaciliteerd in het onderwijs en in de concrete praktijk. Grote geïmplementeerde informatiesystemen zijn gerealiseerd op basis van generatietechnieken voor de Hogeschool (HAN-SIS), Faculteit (FIS), Informatie Communicatie Academie (SAS) en Individueel onderwijs (IO).
De oudste van deze systemen functioneren al meer dan 20 jaar. Het meest omvangrijke en complexe systeem faciliteert de registratie en studievoortgang voor alle 30000 studenten en 3000 medewerkers van de HAN.

### Data Intensieve Systemen

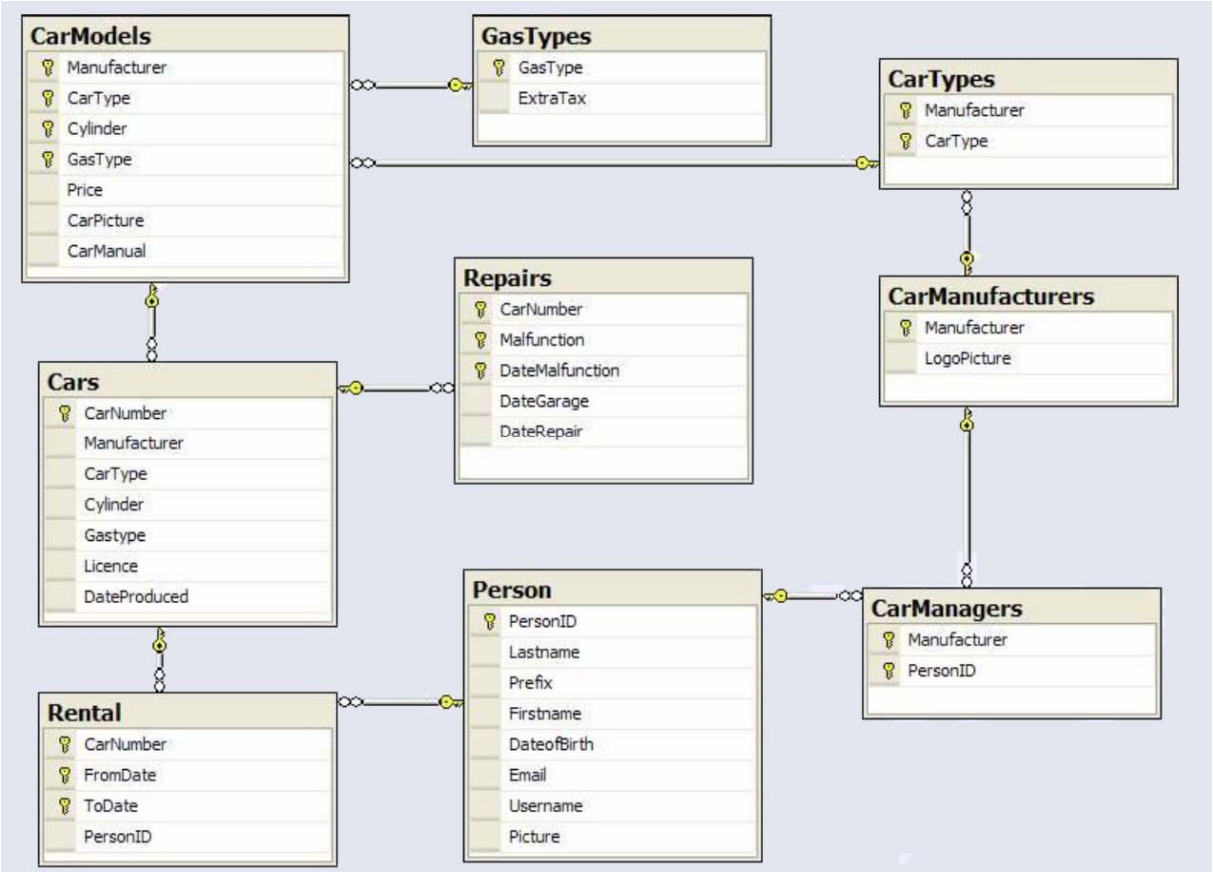
Conceptueel bestaat een informatiesysteem uit een database met een managementsysteem die de gebruikers op hun beeldscherm voorziet van alle functionele transacties die nodig of gewenst zijn.



Figuur 1. Informatiesysteem.

Data intensiteit kent vijf conceptuele parameters: aantallen feittypen, aantallen relationele tabellen in de database, aantallen records in de tabellen, aantallen gebruikers van het systeem en het aantal bedrijfsregels. Daarnaast zijn de complexiteit van de relationele structuur en de complexiteit van de bedrijfsregels, gebruikersrollen en toegangsregels belangrijke factoren die de realisatie van data intensieve informatiesystemen sterk beïnvloeden.
Gehanteerde kentallen voor de data intensieve systemen binnen de onderzoeksgroep zijn in combinatie: meer dan 1000 feittypen, meer dan 100 tabellen met meer dan 1000 records en meer dan 10000 gebruikers en meer dan 100 complexe bedrijfsregels.

**Generator requirements**
De eisen en wensen voor de ontwikkeling van een applicatiegenerator zijn divers. Als eerste eis geldt (minimaal): generatie van een hoofdmenu met als functionaliteit voor elke tabel de standaard recordbewerkingen Create, Read, Update en Delete (CRUD) met standaard voor elke tabel eenvoudige generieke Search, Export en Rapportages (SER). Vervolgens is een 'Single Point of Definition' van alle systeemelementen in de generator en in de gegenereerde informatiesystemen een essentiële eis. Belangrijk uitgangspunt is ook dat de generator de ontwikkelingskosten (specificatie, ontwerp, generatie, testen, implementatie) én de beheerkosten reduceert. De onderzoeksgroep hanteert voor de gegenereerde code als eis dat de generatie code een hogere (4G, SQL) programmeertaal levert, zodat voor de systeemontwikkeling geen kennis van 3G-talen zoals C#, ASP.NET, Ajax of Javascript nodig is.
In de optimalisatieprocessen voor de generatoren is de aandacht voor performance-aspecten van elk detail in de generatie steeds belangrijker gebleken. Reponstijden voor de gebruiker moeten voldoen aan de eis dat voor standaardfunctionaliteit binnen enkele seconden een systeemreactie volgt.
Als algemene eis is webbased functionaliteit van belang, waardoor elke browser als cliënt kan worden gebruikt en daaraan gekoppeld zijn er eisen voor security, personalisatie en meertaligheid.
De basisprincipes van applicatiegeneratie zijn aan de hand van een simpel modelsysteem voor een autodealer te demonstreren (figuur 2). De generator bouwt code in de vorm van SQL-queries door de systeemtabellen (repository) in het databasemanagementsysteem daarvoor te gebruiken (figuur 3). Voor elke tabel wordt een 'formulier' voor de gebruikersinteractie op een beeldscherm gegenereerd.



Figuur 2. Demonstratiemodel voor applicatiegeneratie.

| SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Rental' | | | | | | |
|---|---|---|---|---|---|---|
| **Table** | **Column** | **Seq** | **Default** | **Nullable** | **Type** | **Length** |
| Rental | CarNumber | 1 | NULL | NO | varchar | 20 |
| Rental | PersonID | 2 | NULL | NO | int | NULL |
| Rental | FromDate | 3 | NULL | NO | datetime | NULL |
| Rental | ToDate | 4 | NULL | NO | datetime | NULL |

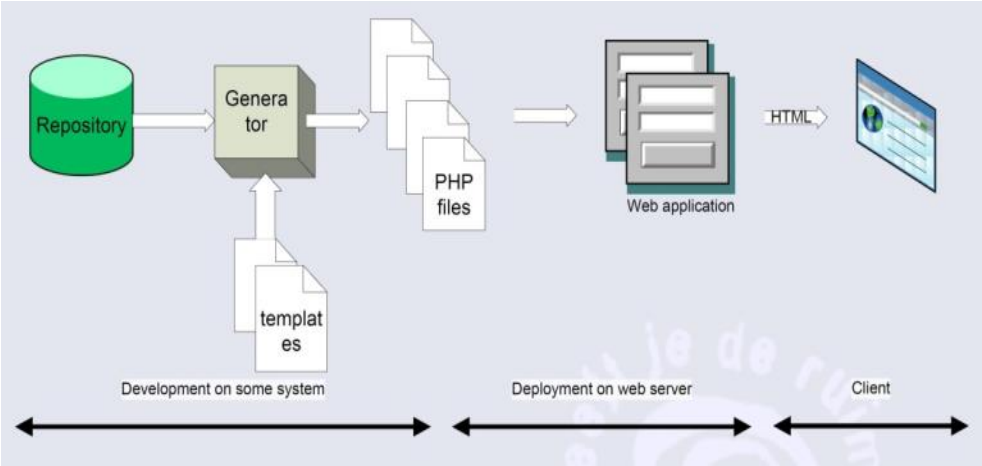
Figuur 3. SQL-query en gegenereerde tabel op basis van de informatie in de repository.

**Typen generatoren**

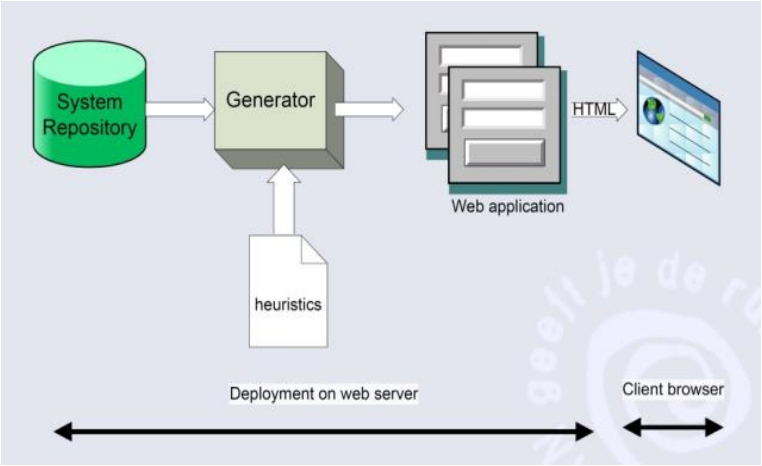
Er zijn twee typen applicatiegeneratoren: code generatoren en runtime generatoren.

De codegeneratoren produceren files met programmacodes in de doeltaal (php, asp, asp.net), die vervolgens door een compiler worden omgevormd tot een uitvoerbaar programma (figuur 4). De generator kan in principe voor een diversiteit van doeltalen code produceren. Het door de compiler vertaalde programma wordt geïnstalleerd en biedt dan de gegenereerde functionaliteit voor de gebruikers.

Een runtime generator (interpreter) bewerkt tijdens het uitvoeren van de programmafunctionaliteit direct de repository en de applicatietabellen in de database (figuur 5). Daardoor is continu tijdens de uitvoering van het programma (runtime) aanpassing van de applicatiefunctionaliteit mogelijk en kan de diversiteit voor gebruikers flexibel worden aangepast. Dit type generator eist een runtime toegang en aanpassing van de repository en is daardoor platformafhankelijk. Het web voldoet aan die eis voor webapplicaties op basis van client-server interactie waarbij de gegenereerde applicatiecode volledig in de repository van de server is opgenomen.



Figuur 4. Codegenerator (voor PHP).



Figuur 5. Runtime applicatiegenerator (interpreter).

**De applicatiegenerator IMAGine**

De onderzoeksgroep heeft vanaf de eerste versies van generatoren rond 2001 gewerkt aan de optimalisering en realisatie van applicatiegeneratoren. De huidige versie is als tool voor studenten, docenten en ontwikkelaars beschikbaar en wordt voor onderwijs en ontwikkeling inmiddels intensief gebruikt. Deze Information Model Application Generation eng*ine* heeft als systeemontwikkeltool onder de naam *IMAGine* waardevolle onderwijskundige bijdragen geleverd. Daarnaast is door de gerealiseerde productiesystemen voor en in de praktijk bewezen dat applicatiegeneratie nu en in de nabije toekomst een belangrijke bijdrage levert aan de versnelling, kwaliteitsverbetering en kostenreductie en beheersing van systeemontwikkeling.

De basisfunctionaliteit die IMAGine genereert biedt alle raadpleegfuncties (browse) op het informatiemodel en een default schermpresentatie met panels (informatieclusters), waarin alle CRUD-functies standaard zijn opgenomen met validaties en stamtabelkoppelingen in combo boxes. IMAGine biedt een universele zoekfunctie op alle informatievelden over elke master-detail relatie en een navigatiefunctionaliteit via de tabel-relaties tot op elk niveau. De standaard auditfunctie voorziet in rollen voor gebruikersgroepen met bewaking op toegang en bewerking op veldniveau van elke tabel. Elke recordmutatie wordt geregistreerd en behouden in historietabellen met een timestamp, gebruiker en machineregistratie als universele mutatielogging samen met de registratie van de gebruikersraadpleging per raadpleegfunctie. Standaard-rapportages en -export zijn uiteraard

beschikbaar zoals ook geoptimaliseerde personalisatie van de gebruikersinterface door registratie en opslag van de laatste keuzes en instellingen in het gebruik.

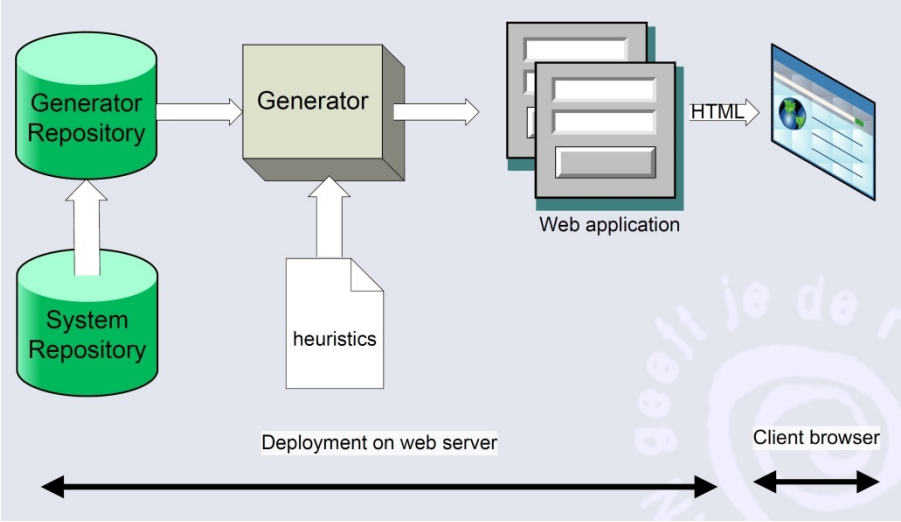
**Gebruik van IMAGine @ HAN**

De applicatiegenerator IMAGine wordt aan alle studenten en docenten voor educatief gebruik beschikbaar gesteld. In het semester "Develop an Information System" (DIS) hebben inmiddels honderden studenten de fundamenten van applicatiegeneratie in courses en projecten ervaren en gepraktiseerd. In projecten in dit semester worden naast realisatie van systemen voor prototyping en proof of concepts voor bijzondere realisaties ook aanpassingen en mogelijke vernieuwingen onderzocht en gerealiseerd. IMAGine wordt in lectoraatsprojecten gebruikt als tool voor business requirements specificatie en validatie.

In ontwikkeling en beheer van productiesystemen voor de HAN (Studievolg), Faculteit Techniek(Facultaire informatie), ICA (stage- afstuderen, inzet) en individueel deeltijdonderwijs (inschrijving, voortgang en begeleiding) wordt voortdurend het praktijkonderzoek gevoedt en versterkt. De laatste jaren is IMAGine ook in afstudeerprojecten ingezet om in bedrijfsopdrachten onderzoek te doen in verbinding met commerciële producten (o.a. Thinkwise, Mendix, PNA).

**Uitgebreide IMAGine functionaliteiten**

Voor de applicatiefunctionaliteit die IMAGine standaard genereert, wordt in eerste instantie de systeemrepository van het relationele databasemanagementsysteem gebruikt. Uitbreiding van die functionaliteiten wordt gerealiseerd door een extra specifieke generator-repository in te richten.




Figuur 6. Uitbreiding van IMAGine functionaliteit.

In de generatorrepository kunnen geavanceerde functionaliteiten in de gegenereerde applicaties worden gespecificeerd (figuur 6). Datatypen die niet standaard in systeemrepository van een relationeel systeem databasemanagement voorkomen worden in de generatorrepository opgeslagen. Daarmee worden o.a. datatypen voor email, picture en calender opgenomen in de functionaliteit van de gegenereerde applicaties. Ook iconen, panels (informatieclusters), schermposities, titels, medelingen en teksten voor meertaligheid worden in de generator-repository opgenomen. Voor de toegangs- en auditfunctie worden per gebruiker kolom- en rij-rechten opgeslagen. Ook taken, rapportages en export-functies zijn voor rollen of gebruikers persoonlijk te specificeren. Bedrijfsregels in deze generator repository bieden detailinformatie voor gebruikers en

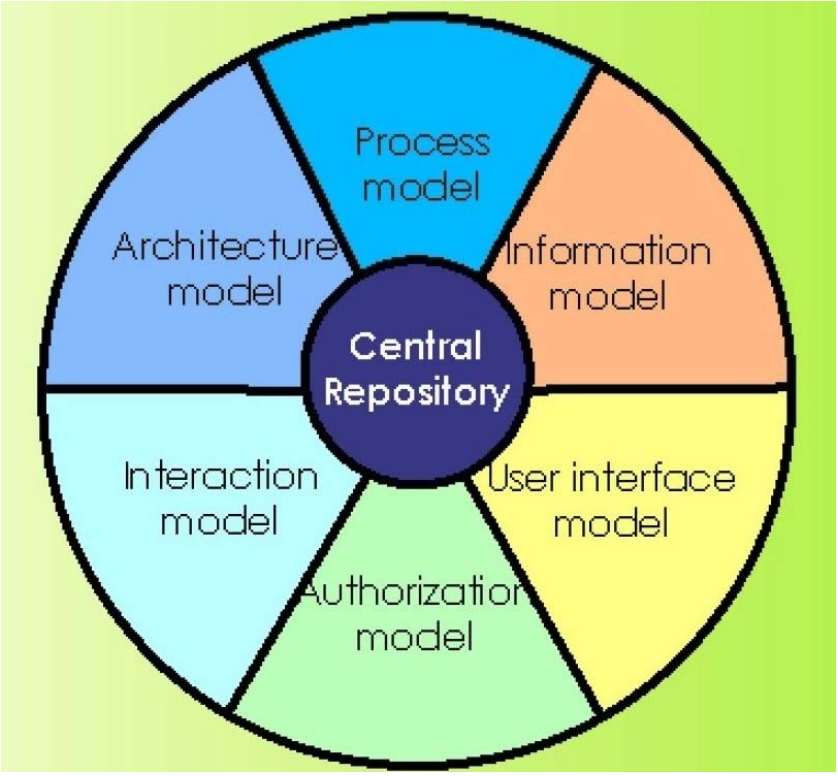
de optimalisatie van gebruikersinteractie door afscherming of visuele identificatie van ontoegankelijke deelfuncties.

**Toekomst van applicatiegeneratie**

De functionaliteit van door IMAGine gegenereerde en gefaciliteerde applicaties is volledig gemodelleerd in enerzijds de applicatierepository en anderzijds in de domeindatabase (figuur 6). Voor de specificatie van de domeinapplicatie worden zes groepen functies in de modellering onderscheiden: informatiemodel, procesmodel, architectuurmodel, user-interface model, interactiemodel, authorisatiemodel. De inhoudelijke domeinspecificatie en requirements worden door informatieanalyse volgens de FCO-IM methode vertaald en automatisch geconverteerd in een relationeel databaseontwerp, waarvoor de datadefinitiecode in SQL wordt gegenereerd. Het domein specifieke informatiemodel wordt opgeslagen in de database en de andere vijf modellen in de (applicatie)repository. IMAGine ontwikkelt zich verder als feit-gebaseerd-generatie-gereedschap (fact-based-CASE-tool).

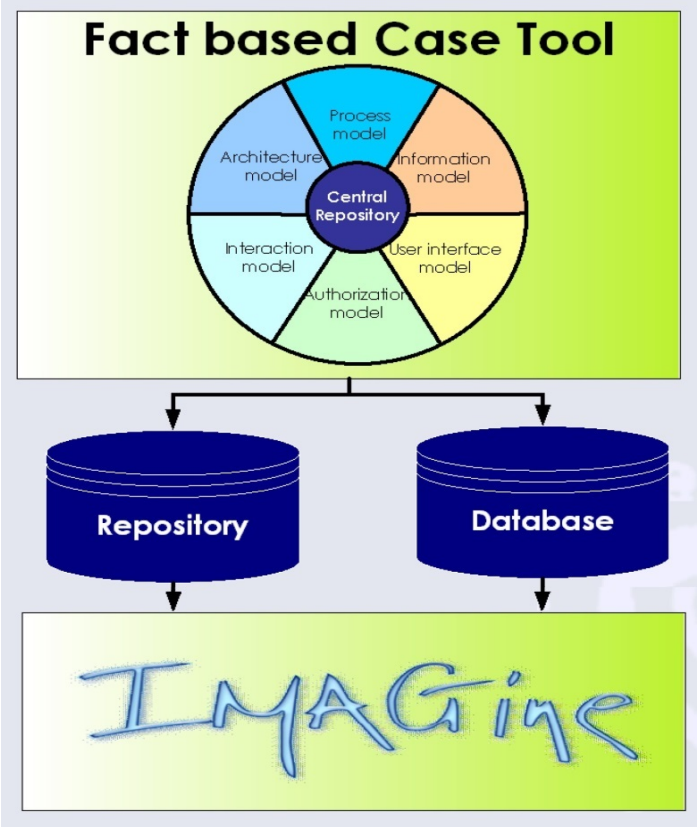
Door versterking van de verbinding en toenemende integratie tussen de informatiemodellering én de applicatiespecificatie kan een functioneel nog rijkere applicatie worden gegenereerd (figuur 6).

De applicatie-functionaliteit kan ook zelf nog worden uitgebreid. De gerealiseerde praktijktoepassingen laten zien dat er mogelijkheden voor additionele functionaliteiten liggen bij koppeling van schermformulieren, integratie van grafische data, een reductie van de interpreter(engine) functies door nog meer in de database te registreren. Daarnaast wordt XML als uitwisselingsformat toegepast. Meer flexibele gebruikersinterface bv door panels in panels, grids in panels en buttons achter controls. Tenslotte liggen er kansen en plannen om de bedrijfsprocessen nog meer te integreren in de applicatie-specificatie en generatie.




Figuur 7. Modelaspecten van een feit-gebaseerd Case Tool.

Aspecten van de applicatie die in de generatorrepository worden opgenomen zijn nader te definiëren en per applicatie te specificeren in diverse submodellen zoals: het Informatiemodel; het Procesmodel; het User Interface model; het Authorisatiemodel; het Interactiemodel; het (technische) Architectuurmodel. Het CaseTool realiseert dan samen met de applicatiegenerator een zeer flexibele specificatie, systeemontwikkeling en realisatie van (web)applicaties (figuur 8).




Figuur 8. Toekomst flexibele systeemontwikkeling en applicatiegeneratie met IMAGine.

Wilt u reageren op deze presentatie? Neem dan contact op met:
Arnoud van Bers; hoofddocent / onderzoeker; Lectoraat Model Based Information Systems; Hogeschool van Arnhem en Nijmegen.
arnoud.bers@han.nl