3.1. Academic works

Automatic programming and software generation are two fields where the scientists and ICT technician have worked in the past years.
These techniques are applied to different fields in real life and they do not only refer to user interface and web application generators.

3.1.1. Web engineering

Some works in the late 90' and early 2000 addressed the problem of website generation and user interface design, including HDM [XXX-HDM - A Model-Based Approach to Hypertext Application Design.], HDM-Lite [XXX-A Conceptual Model and a Tool Environment for Developing More Scalable], RMM [XXX-RMM: A Methodology for Structured Hypermedia Design],OOHDM [XXX-Web Application Models are More than Conceptual Models], Araneus [XXX-Design and Maintenance of Data-Intensive Web Sites], Strudel [XXX-M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. Strudel: A web-site management system. In Proc. ACM SIGMOD, pages 549{552, Tucson, Arizona, May 1997], Hera [XXX- A Model-driven Approach for Designing Distributed Web Information Systems] , UWE [XXXX-Model-Driven Generation of Web Applications in UWE], MIDAS [XXX-A MDA-Based Approach for Web Information System Development] and WebTE [XXX-WebTE: MDA Transformation Engine for Web Applications]

[XXX-HDM - A Model-Based Approach to Hypertext Application Design.] define a method for the description of complex applications without much concern in implementation details. It makes use of the notions of perspective, the identification of different categories of links and the possibility of easily integrating the structure of a hypertext application with its browsing semantics. A Number of links can be derived automatically from a conceptual-design level description.
With [HDM-Lite] has been introduced a new method for the development of web applications, defining a notation that supports the structural, navigational and presentation layer of the

application. Conceptually it is an evolution of HDM. Three different layers are formalized in three schema, hyperbase, access and presentation schema respectively.

Relationship Management [XXXX-] focuses on a methodology for design hypermedia applications and it is defined by three steps. The first step is about the ER diagram design, a model that represents the domain of the application. During the second phase the entities are divided in slices that group together similar information: developer decide how information will be shown to the users. The last phase of RMM is dedicated to navigational design: each associative relationship In the ER diagram is analyzed and eventually considered as navigational path.

OOHDM [XXX-] is a model-based approach for building large hypermedia applications and includes four different activities:

- Conceptual design
- Navigational design
- Abstract interface design
- Implementation

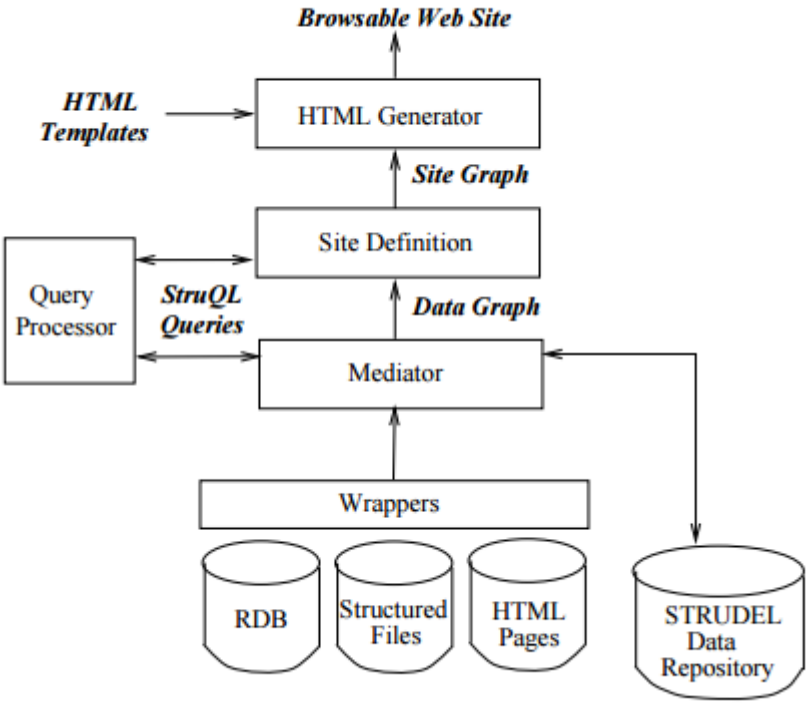
Conceptual design is made using UML.

The most important feature of this method is the inequality between the objects the user navigates and the conceptual objects: the firsts are built, with a view mechanism, from one or more conceptual objects. In the third phase the appearance of the view is defined.

Araneus [XXX-] is a prohect of “Università di Roma Tre” and addresses the problem of defining an integrated environment for unstructured and structured web content, base on a WBMS (Web Base Management System), a database technology for storing both data and metadata descriping the structure of website.

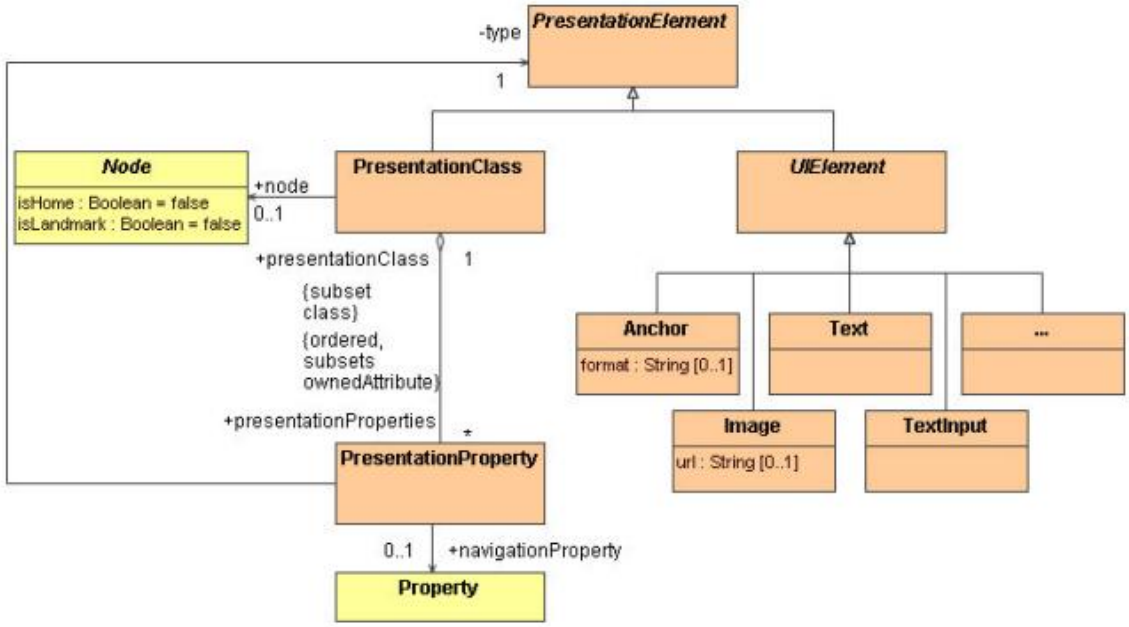
Designer 2000 [XXX-M. Gwyer. Oracle Designer/2000 WebServer Generator Technical Overview (version 1.3.2). Technical report, Oracle Corporation, Sept. 1996] was part of the Oracle Web Development Suite and offered a Web Generator to produce HTML pages by taking in input a model by Designer 2000. It stores metadata in the DMBS.

Strudel [XXX-M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu. Strudel: A web-site management system. In Proc. ACM SIGMOD, pages 549{552, Tucson, Arizona, May 1997] defines a graph structure where each page is represented as a node. This project offers data integration but does not provide any user-friendly interface.

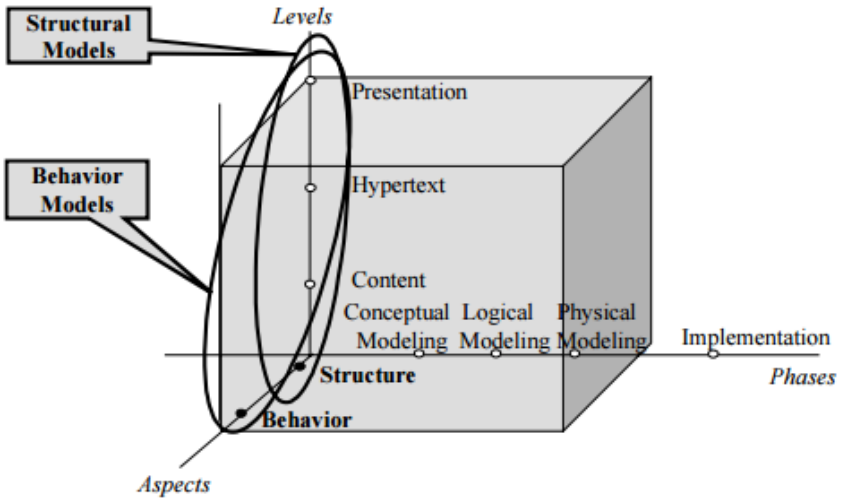


Even Hera [XXX- A Model-driven Approach for Designing Distributed Web Information Systems] (a WIS design framework) perspective consists in three layers: semantic, application (navigational view over data) and presentation layer (dealing with concrete rendering like HTML). The Hera framework is based on the WWW standard and has a model-based approach since WIS are data-intensive applications. It is mainly focused on data integration model.

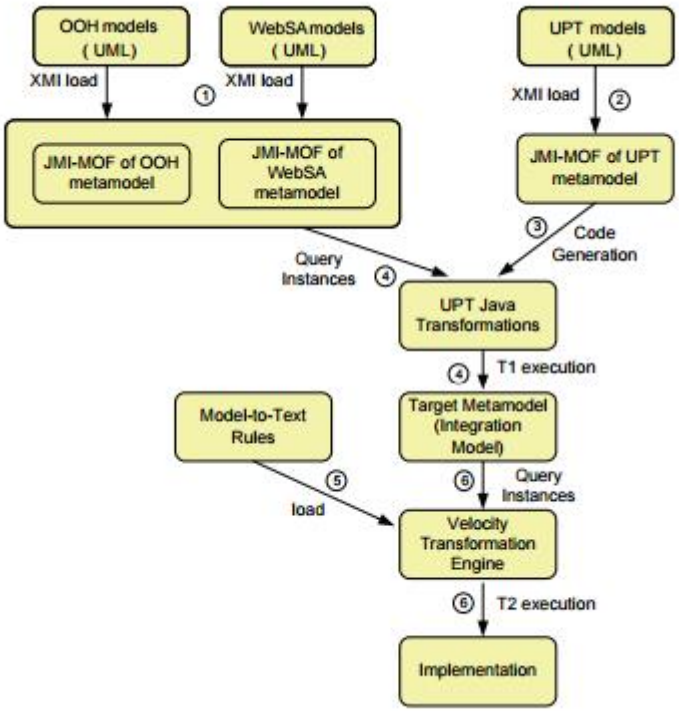
UWE [XXXX-Model-Driven Generation of Web Applications in UWE] is another important project in the MDE techniques, addressing the problem of model abstraction and transformations in web engineering. Model is transformed to web pages through ATL transformations. UWE makes use of Content model, Navigational model and Process Flow model before arriving at the target Presentation model. UWE metamodel is composed of presentation element categorized in presentation classes and UI elements.



MIDAS [XXX-A MDA-Based Approach for Web Information System Development] is a model-driven methodology based on MDA for the development of Web Information Systems. It categorizes a WIS by means of three orthogonal dimensions: levels, aspects and phases. PIMs and PSMs are represented with UML.



WebTE [XXX-WebTE: MDA Transformation Engine for Web Applications] is a tool based on the MDWE approach and WebSA[XXX-Applying Model Driven Engineering to Web Applications] that takes in input a model and return the final implementation of a web application by means of model-to-model transformations and model-to-text transformations. The advantage of this method is the complete interoperability with other tools as the models are represented with UML in XMI format readable by any UML tool. WebTE transformation engine is developed using the J2EE platform.



These projects lead to the invention of WebML.

3.1.2. WebML

Web Modeling Language[XXX-Web Modeling Language (WebML): a modeling language for designing Websites] is a notation for specifying complex websites at the conceptual level, enabling high level description of a website considering its structural model, composition model, presentation model and personalization model. It is associated with a graphic notation and a textual XML syntax, platform independent and guarantees a model-driven construction of complex sites.
It was invented at Politecnico di Milano by Stefano Ceri and Piero Fraternali.
A first CASE environment, Autoweb System [XXX-Model-Driven Development of Web Applications: the Autoweb System] has been detailed in the year 2000. It is based on HTM-Lite and it consists in two phases, the generation of the database and meta-database and the implementation of the web application.

3.1.3. IFML

Interaction Flow Modeling Language[XXX-IFML- Model Drive UI Engineering of Web and Mobile Apps with IFML] is the successor of WebML[XXX- Designing data-intensive web applications] and has been built by a team at Politecnico di Milano, including Roberto Acerbis, Aldo Bongio, Marco Brambilla, Sara Comai, Stefano Butti and Maristella Matera.

It has arrived at the beta2 version and in March 2013 has been adopted by the Object Management Group.

It is supported by WebRatio [XXXX- www.webratio.com] solution and by IFML-Editor [XXX- http://ifml.github.io/] , an open source editor based on Sirius and Eclipse.

The goal is to provide a graphic tool that permit to define the behavior of the UI with respect to the final user. The tool will then generate the code to fulfill that goal.

IFML supports the platform independent description of graphical UI for web applications as the language allows the description of the view layer.

User interface is described by different ViewContainers that can contain SubContainers and ViewComponents. ViewComponents can have input and output parameters and can be associated with Events, user action mapped in the backend software.

There exist other tools that support MDE development on similar principles of IFML, like Mendix [XXX- www.mendix.com], Outsystems [XXX- www.outsystems.com], OrangeScape [XXXX- www.orangescape.com], LongJump/AgileApps Live (www.softwareag.com/special/longjump/), Tersus (www.tersus.com) and Softfluent Entities (www.softfluent.com).

3.1.4. User interface generation

A common problem in software developing is represented by the user interface. It must be standard through all the different views of the application and can be automatized. HTML or XUIL are perfect examples for the standardization of the user interfaces, each component of the same type is rendered in the same way, based on the operating system and the browser version.
There are studies that addresses this problem which propose a different approach based on software mining.
Software mining is a branch of data mining focused on mining software artefacts such as source files and database schema for useful information related to the characteristics of a system. The main idea consists in analyzing the software metadata and inspect the back-end architecture before creating a native UI suite.
It has been implemented with the project Metawidget, a framework that inspects the backend software of different technologies and allows to create user interfaces for a variety of client applications and technologies.

This topic has been addressed even some years ago for the UI of desktop applications. This guaranteed a faster developing in software engineering.
Early examples of model-based tools include Cousin [XXX-- “Design Alternatives for User Interface Management Systems Based on Experience with COUSIN] and HP/Apollo’s Open-Dialogue [“ADM-A Dialogue Manager,” in Proceedings SIGCHI'85: Human Factors in Computing Systems] that provided a declarative language in which the designer listed the input and output of the user interface. The system then generated the dialogs to display and request the data. These evolved into model-based systems, such as Mike [XXX- “Mike: The Menu Interaction Kontrol Environment], Jade [XXX- Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces], UIDE [XXX-- A Second Generation User Interface Design Environment: The Model and The Runtime Architecture], ITS [XXX- ITS: A Tool for Rapidly Developing Interactive Applications.], and Humanoid [XXX-- Beyond Interface Builders: Model-Based Interface Tools].

These systems used techniques such as heuristic rules to automatically select interactive components, layouts, and other details of the interface, leading to some difficulties in control.

Developers were required to learn a new language for defining models, not helping the technique-s diffusion.

3.1.5. Web-based generation

Another branch of the automatic software generation is related to web based applications.

The main idea is to analyze the model of the application in order to be able to generate the server-side software and the related views: the software must be allow to insert, editing, validating and managing entities.

Data-entry applications are typical examples of this pattern.

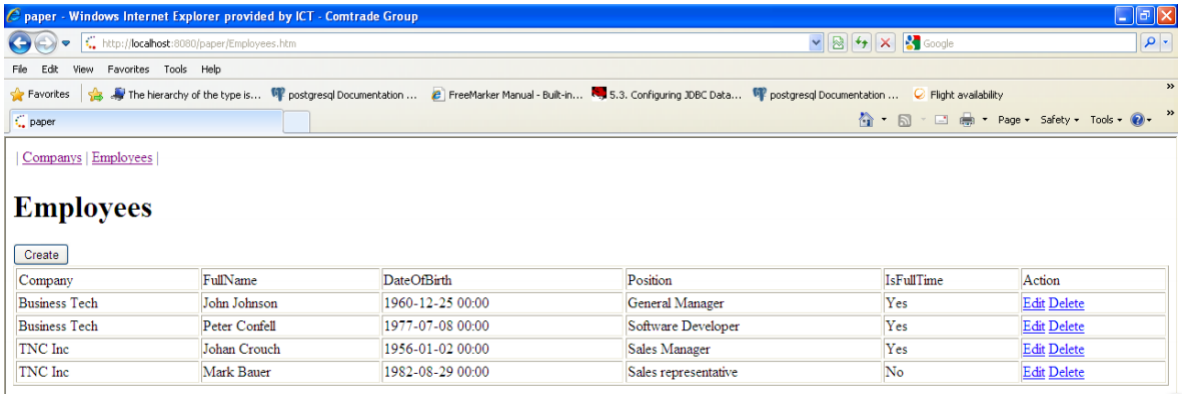
Business logic and presentation layer should be completely separated, that's why these kinds of application make use of different exchanging format for communications.

The difference between the various projects is the metamodel's specification used.

Some makes use of XML for defining elements but most are based on the Eclipse EMF framework.

EMF project is an Eclipse modeling framework and code generation facility that operates on source XMI models and produces Java classes. The input of the model may be a set of UML diagrams on top of which the framework applies model transformation in order to obtain the desired result.

A different approach [XXXX- A Generator of MVC-based Web Applications] is given by HibernateTools toolset to analyze a database's schema metadata. By obtaining the meta information it is possible to generate the required output. It uses Freemarker ad rendering technology.



The technologies adopted are different as some researches have been developed.

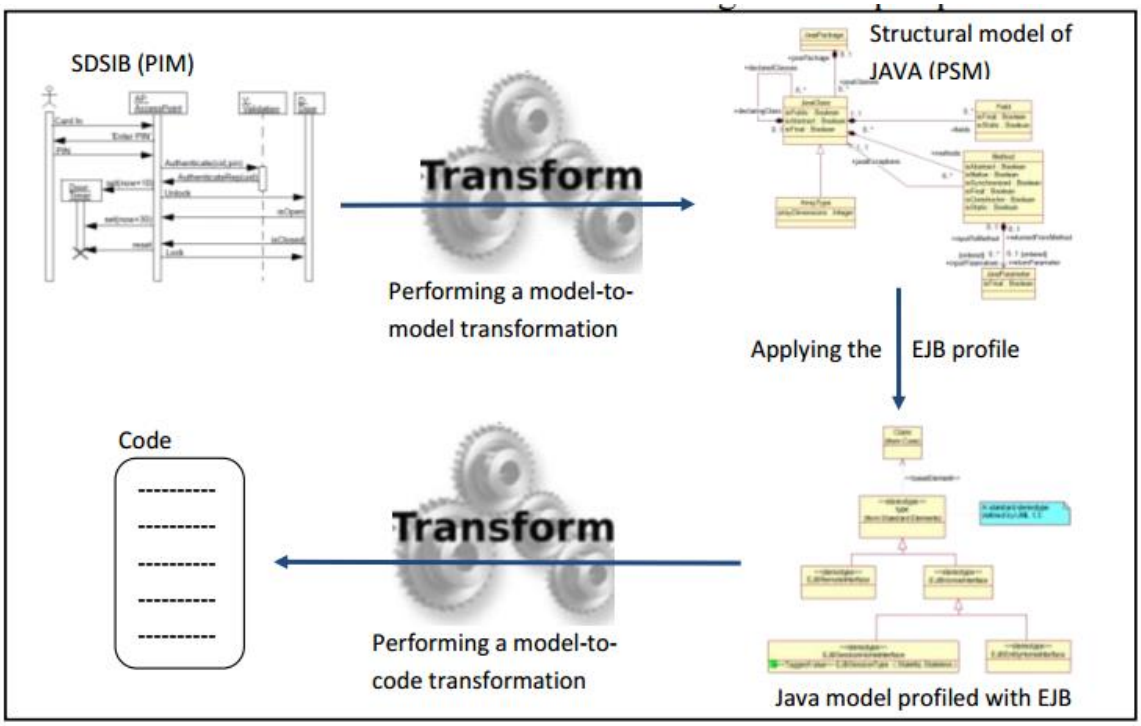
[XXXX- TOWARD AUTOMATIC GENERATION OF MVC2 WEB APPLICATIONS] Is based on a source metamodel built with UML and applies transformation that lead to the target model, which is a simplified version of the relational databases's schema. Implementation of the mapping rules is based on EMF with a programming approach.
The output of the method is an XML file containing all actions, forms and forward jsp pages.

3.1.6. Code generation from sequence diagram

Starting from a UML sequence diagram is a different approach to code generation. Sequence diagram can represents the MVC pattern and it is possible to obtain a model transformation from the source model to the target model, by means of a Java PSM as intermediate model.

Some project follows this concept [XXXXX- Automatic code generation by model transformation from sequence diagram of system's internal behavior

] and are able to generate controllers that satisfy the source use case.



The first phase consists in transforming the source model into a set of Java classes, annotated in order to support relationships. On top of this EJBs it is built an interface to support CRUD operations. The transformations follow the EMF framework.

3.1.7. Other types of code generation

Code generation is used in many application's field above Crud software.

First use of the code generation comes together with the definition of the first programming languages. Programming languges has been developed from the first generation (machine language) till the fifth-generation (constraint-based), passing through assembly language, high level language and declarative languages [XXXX- Programming Languages: History and Fundamentals (Automatic Computation)]. Coding a programming language is the first example of code generation.

The AUTOPASS project (AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly) refers to a system able to translate English-like sentences in software. It uses a geometric database generated prior to compilation and updated during the compilation, that represents the state of the worls at each assembly step.

Other project [XX – Special purpose automatic programming for 3D model-based vision] applies automatic programming to recnognizing software. Given a description of a 3D object it generates a software able to recognize the object in images, without restrictions on the orientation of the object in space.

A generator for interactive voice response applications [XXXX- https://www.google.com/patents/US6456699] has been patent in 2002. The software is able to inspect the menu section of a website and produce a IVR menu.

Automatic programming has been applied also in the robotics field [XXX- Automatic Programming of Robots using Genetic Programming]. The proposal of the project is to generate a computer program that enable an autonomous mobile robot to perform simple tasks. It does not apply reinforcement learning algorithms due to their computational and knowledge issues. It is built with the genetic programming paradigm.