VIDE Final Report
www.vide-ist.eu

Author: PJIIT

Table of Contents

1 Acknowledgements

To European Commission for supporting the project, particularly to Project Officer Jorge Gasos and DG INFSO staff that were very helpful during our 2,5 years of cooperation.

Special thanks to all the members of project teams that worked to successfully complete the tasks from PJIIT, Bournemouth University, Fraunhofer Institute (FIRST and IESE), Iwi, SAP, Rodan, TNM, Softeam and Altec, especially to the Leaders of Work Packages.

Reviewers Mr Michael Lemoine and Mr Frantisek Plasil

Special gratitude to all partners from the business community that supported the project answering our surveys and giving guidance, and also to academics who evaluated our work and were the first to use the VIDE tool.

2 The VIDE Project in a nutshell:

Contract Number	FP6-IST-2005-033606
Project Acronym	VIDE
Project Name	Visualize all model driven programming
Priority	IST-2005-2.5.5 Software and Services
Duration, start date	30 months, 1 July 2006
Instrument	STREP
Total Cost	3 963 930 Euro
Commission Funding	2 296 614 Euro

VIDE team

Coordinator - Polish-Japanese Institute of Information Technology (PJIIT) – expertise in language semantics, database languages and visual interfaces to databases as well as object-oriented programming and modelling.

Rodan Systems S.A. – specialist in workflow systems.

Institute for Information Systems at the German Research Center for Artificial Intelligence -involved in Computer Independent Modelling & Procedure Models.

Fraunhofer – (IESE) expertise in Software Quality Assurance, (FIRST) expertise in aspect orientation semantics and decomposition techniques.

Bournemouth University – with expertise in software engineering, systems, business process modelling, requirements, modelling and user interface design.

SOFTEAM – major UML modelling tool vendor, active participant of OMG UML standard development.

TNM Software GmbH – specialized in Inter-/Intranet, mobile Devices, E-Learning systems.

SAP AG – is the world's leading provider of business software solutions that help enterprises of all sizes around the world to improve their business operations.

ALTEC – developer of business information systems and services ERP systems and financial systems for SMEs.

3 Executive Summary

Background

The foundation of the VIDE project was based on the ambition to bridge the gap between the enormous potential of visualisation in software development and the proven lack of truly successful application of visual notation in the development process.

The research was also motivated by the concept of Model Driven Architecture (MDA) from the Object Management Group (OMG) and its vision of creating further layers of software development solutions based on modelling languages thus automating construction of today's programming language code. This goal implies providing a modelling language with the complete means of behaviour specification and with precise executable semantics over them. However there are still many challenges ahead. For example the question remains on how to assess new requirements and change requests and moreover how:

- do they affect the code, the PSM, or the PIM, or CIM?
- to locate the model/code/transformation rule that is affected by a change request?
- to overcome problems resulting from the tenuous connection between CIM and PIM and the muddy path from requirements to design ?

There have been several attempts to overcome the issues that are inherent in visual programming and have been a barrier to executable modelling. In the VIDE Consortium we believed that there was still a lot to be done for the benefit not only for novice users learning how to program, but also in the IT community in general in terms of overall software development productivity.

Aims, objectives and innovativeness

The key feature of the VIDE language and toolset is the shift of programming activities towards the fully platform independent level grounded in the UML notation. Whilst the standard already provides the

necessary vision, there are no existing tools which realise it, and hence the need to be addressed by the VIDE project. Thus the following features are unique to the VIDE project:

- A uniform UML-based language allowing for precise behaviour specification. This provides more flexibility for the software development process and reduces the communication challenges.
- OCL-based query language features which will allow the required functionality to be achieved with a significantly smaller amount of code.
- Queries and imperative statements seamlessly integrated into a single language backed by standardised metamodel.
- Both an aspect oriented approach and quality defect discovery techniques which are moved to the platform independent level to increase reuse and to manage the complexity of applications being developed with VIDE.
- Full coverage of the transition process from informal business models to executable software models.
- Platform independent executable model providing connectivity with existing services and allowing process engine interaction as well as GUI prototyping and testing
- An informal “pre-CIM” modelling tool addressed to business users to provide a starting point for the development process
- Extended CIM-level language based on the BPMN notation
- CIM to PIM transition in terms of model exchange as the starting point for application specification and as a blueprint for executable workflow process design

Such a consistent, seamless and tool supported transition will maximize the input of domain experts to the process of application development.

Results

In the course of work a number of assets have been developed. They refer to various levels of modelling and programming that span from the computation independent level down to executable code.

Exploitable result 1: CIM Tool and Process

The CIM Tool and process consists of both a Domain Analysis Tool at the pre-CIM level and the PIM Prototyping Tool which are designed to be capable of exploiting plug-ins to allow for different notations at the CIM level. The VIDE tool uses VCLL, but it is envisaged that this plug-in will use different notations. The tool allows entry to the VIDE tool-set at the business domain level and access to the tool by users not familiar with process notation.

Exploitable result 2: VIDE CIM Level Language tool

The VIDE CIM Level Language (VCLL) provides an enhanced business process modelling notation based on BPMN. The results in detail are:

- The meta model and OCL based definition of the VCLL
- The Modelling tool for the VCLL which will consist of four sub-components:

- the modelling component for organizational structures of enterprises
- the modelling for data objects
- the graphical editor for Business Rules and
- the modelling component for business processes, which will reference modelling elements created by the three other sub-components above.

Exploitable result 3: Workflow Management
OfficeObjects(R)WorkFlow is the standard-based commercial product of Rodan Systems S.A. for business process management. In the VIDE project it has been extended to support XPDL 2.0 at the definition level and direct web services calls at both the definition and execution levels.

Exploitable result 4: VIDE Core
The solution is based on the integration of OCL and UML Actions and a concrete syntax of textual VIDE language based on it, implemented in the form of a UML-compliant model repository, language editor and compiler to UML metamodel, as well as the model browser, UDDI registry browser for Web service development and Topcased integration. The component also includes the Visual Expression Builder editor for visual specification of OCL queries.

Exploitable result 5: State Visualisation modelling component
Will remain unpublished; only for internal use

Exploitable result 6,7,12: ODRA development environment with model execution capability (Incorporates ODRA Model Compiler, ODRA Execution Engine and Wrappers for Relational and XML data)
The result includes ODRA object database management system incorporating the model compiler from UML/VIDE to ODRA source code, and utilities for direct execution of models on the engine. Moreover, the environment includes wrappers for relational and XML data that assure representations of data sources using UML and runtime components for integrating them with ODRA.

The product is the prototype software and language concept for the ODRA platform. ODRA will constitute a complete programming environment that promotes the concept of a seamlessly integrated query and programming language over an object-oriented data model. The increased productivity is achieved through removing the so-called impedance mismatch at the joint of query language embedded inside a general-purpose programming language and through a flexible solution for virtual updatable data views. As such, ODRA may play important role in data integration. Wrappers and adapters for relational data and for Web service interfaces are to be further developed. Data interchange interfaces for XML are also under development. The development of ODRA schema and application code is to be supported by model-driven solutions based on UML.

Exploitable result 8: Java Model Compiler (Execution Engine)
The Java Model compiler consists of an ECLIPSE plug-in running on the Ganymede version. It exploits VIDE model files in EMF (CORE) XMI format to produce full Java code with JDO statements and

deployment files for persistence part and JAX-WS statements and deployment files for Web Services production and consumption. This plug-in is based on oAW framework and is available freely.

Exploitable result 9: PIM Level AO support

The AO support at platform independent model level consists of the concepts for representing aspect-oriented constructs as well as the concepts for weaving aspect-oriented platform independent models. The approaches were implemented and integrated into the main VIDE development process and into the VIDE Eclipse based tool chain. The representation of the aspect-oriented constructs is realized using UML Profiles. The Aspect weaver is developed using the Atlas Transformation Language (ATL), which is a model-to-model transformation approach. The aspect weaver is integrated into the eclipse based development environment as an Eclipse plug-in.

Exploitable result 10: Quality Defect detection (VIDE-DD)

The VIDE Quality Defect Detector (VIDE-DD) is responsible for the automatic diagnosis and presentation of quality defects within platform independent models. This encompasses structural as well as behavioral diagrams used to visually model a VIDE-based system. It uses information about the software model, stored within the VIDE PIM repository, to analyze the model, diagnose quality defects, and annotate the model using UML-Annotations. The information stored within these Annotations is then used by the quality defect presenting mechanism (an extension to the diagrams presentation mechanisms) to visually enrich the diagrams (within the VIDE visual editor) with information on these defects. These defects are symbolized using icons at the defective elements on the diagram, such as a class, a method, a relation or the diagram itself.

Exploitable result 11: ONAR Integration

The VIDE-ONAR toolset is employed for the development of sophisticated application logic that manipulates the published resources, available as web services built upon ONAR ontologies mapping conceptually the problem domain, in an orchestrated manner defined in PIM business logic level. ONAR services are invoked by the VIDE repository through the import functionality providing the services' URLs. As a result the PIM repository creates based on the retrieved ONAR wsdl files, the model representations of the services, to register them in the VIDE service registry.

Exploitable result 13: WebFace Integration

TNM:WebFace is a development platform for graphical user interface. The visual editor allows to develop graphical user interface based on Java without writing Java code. The GUI is composed with predefined graphical and logical components which are connected with logical event connections. There are graphical editors for all components of the GUI which allow to configure the parameters (for example colours, size, fonts,...). The integration of the TNM:WebFace editor in the VIDE platform is based on the web-service standard. It's possible to develop and test graphical user interfaces for VIDE applications.

Exploitable result 14: e-Learning System

The VIDE e-learning system provides an introduction in the VIDE development process. The course is accessible from the VIDE web-site at http://www.vide-ist.eu.
The originally planned scope of the project has been increased significantly in the course of investigation of the state of the art and requirements analysis; to eventually involve all major layers of the MDA. The analysis of the project results shows that the potential for improving the productivity lies not only in visualisation but also in assuring a more seamless, uniform and traceable integration of models spanning from informal domain models down to executable code. Taking into account the precise semantics of the PIM language, model execution and code generation can be assured and this way a significant part of complexity and irregularity of the platform-specific solution can be mitigated.

Evaluation

The evaluation validates the measurable results outlined in the Description of Work and further requirements specified in the course of the project, and consists of two main views.
The first is a "vertical" evalauation, as it is focused on comparing whole toolchains across the modelling layers and development process phase against the VIDE tool set. 27 essential features of VIDE were identified, classified into modelling layers:

- pre-CIM
- CIM
- CIM-to-PIM
- PIM
- PIM-to-PSM and code

and compared with 10 commercially available toolchains of similar area of application. The comparison has shown that the VIDE features identified are indeed unique – either completely, or at least in combination inside particular toolchains.
The second was termed a "horizontal" evaluation, as it was aimed at investigating selected features of VIDE inside particular layers of the framework. This part complements the vertical evaluation by checking if the VIDE unique features identified there actually bring an advantage. In other words, the workshops, competitions and study were performed with the purpose of verifying the expectations expressed in the project's description of work Measurable Results and the other intended features of the VIDE tooling. The possibilities for experimentation and comparison were limited by the fact that VIDE constitutes a proof of concept prototype implementation, which makes it problematic to compete with mature industrial tools in terms of usability. Nevertheless, most of the horizontal evaluation work was performed in the form of workshop using actual VIDE functionality and comparing it against traditional solutions.

Industrial adoption and future work

The concepts and products developed in the course of projects are to be further developed in different areas by all partners – in terms of industrial applications or further research work.

SAP is exploring a number of VIDE related exploitation items or activities, within their research process. These involve the modelling infrastructure, behavioural aspects in the design and maintenance of business processes and business components, aspect Oriented Modelling and Model Visualisation techniques. SOFTEAM will enhance their code generation capacity with behavioural generation to keep their competative advantage over free tools. In addition they have improved their integration with the Eclipse IDE thus lowering the roundtrip effort from the model to the compiler. Rodan Systems S.A. has provided its OfficeObjects®WorkFlow product (with VIDE extensions) to support the VIDE process view at PIM and PSM levels. RODAN has an exploitation strategy which features further investigation of the pre-CIM and CIM level tools provided by BU and IWi which provide a natural business domain entry level to their tooling. ALTEC aims the VIDE toolkit as an add-on module to the ONAR platform. The existing marketing policy of ONAR, provides the ONAR platform as an additional module to customers with existing ERP, CRM, and Accounting products offered to market. VIDE will provide a natural extension to TNM's exploitation activities and they will introduce VIDE at selected customers, whilst continuing to disseminate information on the product. TNM will continue to host the VIDE web site.

Although the industrialization of the tool is mostly assigned to the abovementioned partners the remaining members of Consortium have also undertaken some steps to advance the production of the tool that may finally be adopted by the commercial users. PJIIT have decided to release a beta version of their software in the next few years that will be useful for SME companies in Poland. BU, IWi and Fraunhofer are also open to the potential development of their tools to make them commercially viable.

The areas that will be researched further include: model visualisation techniques with focus on target users, runtime and executable models, the verification of model correctness, aspect oriented modelling, database technology and object oriented programming environments, domain-specific languages, model-based testing, and business rules modelling.

Conclusions

The results should be considered promising. They suggest a significant productivity gain resulting from the simplicity and seamless design of the VIDE PIM language. The visual syntax was found intuitive for novice users, while the visual expression solution substitute to OCL textual coding allowed the users to realise non trivial queries. The comparative analysis of the VIDE AO features confirm they offer analogous expressive power as the AO solutions of programming languages at the PIM level. The workshop on VIDE Defect Detection shows that the tool allows a significantly higher quality defect detection rate than a manual inspection. Also the non-IT user oriented VIDE solutions offered by the "pre-CIM" tooling have been found, in the course of workshop experimentation, much more approachable than traditional modelling solutions that would constitute a communication barrier for domain experts. Thus we can conclude that the project has, and will continue to contribute to the European Research Area, has commercial value and is a success.

4 Background

4.1 Background to the project and how it builds on previous work

> *""There are not many Visual Languages that would be generally agreed are 'successful', and there is little in the way of formal experiments or informal experience that shows that Visual Languages are good. It would be interesting to see experimental results that demonstrated that visual programming techniques or iconic languages were better than good textual methods for performing the same tasks. Metrics might include learning time, execution speed, retention, etc. Fortunately, preliminary results are appearing for the advantages of using graphics for teaching students how to program."*
>
> Myers, B. A. 1990. Taxonomies of Visual Programming and Program Visualization. *Journal of Visual Languages and Computing* 1(1): 97-123. p. 117.

> *"I do not think I ever reflect in words: I employ visual diagrams, firstly because this way of thinking is my natural language of self-communion, and secondly, because I am convinced that it is the best system for the purpose"*
>
> Roberts, D. D. 1973. *The Existential Graphs of Charles S. Peirce*. The Hague, Mouton & Co. N. V.

The foundation of the VIDE project was based on the ambition to bridge the gap between the immense potential of visualisation in software development and the proven lack of truly successful application of visual notation in the development process .

The work on visual programming languages which started with the beginning of computer graphics, when William Sutherland in 1965 developed the simple visual data flow language continued with David Smith's Pygmalion in 1975 that incorporated icon language and programming by example. These basic concepts were further developed more recently into many languages that can be classified into:

- Purely visual languages
- Hybrid text and visual systems
- Programming-by-example systems
- Constraint-oriented systems
- Form-based systems

The field of visual programming languages is starting to mature, although there are only a few areas where it has gained broader acceptance in industry as there are still issues that remain unsolved. While some solutions are successful in making programming more approachable to novice users, there is a common concern about the productivity of fully visual languages in industrial applications.

> *"I don't believe that graphical programming will succeed just because it's graphical. Indeed I've seen (and worked with) several graphical programming environments that failed - primarily because it was slower to use than writing code. (Compare coding an algorithm to drawing a flow chart for it)."*
>
> M. Fowler: UML as Programming Language, 2003.[1]

These observations have drawn attention towards hybrid solutions that preserve some of the advantages of textual programming languages. Another observation has been that visual programming is more successful in domain-specific solutions (e.g. BALSA, ARK, LabVIEW). On the other hand, success of visualisation in high level UML models, motivates the search for a generic purpose syntax that would at least partially extend them towards precise behaviour modelling. This may be essential another prominent concept of software engineering that is raising the level of abstraction and platform independence in software development. The Object Management Group (OMG) created an initiative of Model Driven Architecture (MDA) which has established the vision of creating a further layer of software development solutions based on modelling languages thus automating construction of today's programming language code. This goal implies providing a modelling language with the complete means of behaviour specification and with precise executable semantics over them. However there are still many challenges ahead. For example the question remains on how to assess new requirements and change requests and moreover:

- how do they affect the code, the PSM, or the PIM, or CIM?
- how to locate the model/code/transformation rule that is affected by a change request?
- how to overcome problems resulting from the tenuous connection between CIM and PIM and the muddy path from requirements to design ?

In general, writing transformation rules is very difficult and software architects must understand what transformation the different tools perform. Transformation languages are not a standard yet – innovative prototype experiments such as the VIDE project and it's empirical evaluations are challenging.

Software Architects are not used to creating a PIM-level design that is complete and consistent. There is still a challenge as to how to proceed with those phases to get high quality outcomes and how to validate and verify design and introduce quality attributes into different layers and the required transformations.

There have been several attempts to overcome the issues that are inherent in visual programming and have been a barrier to executable modelling. In the VIDE Consortium we believed that there was still a lot to be done for the benefit not only for novice users who learn how to program, but also in the IT community in general in terms of overall software development productivity.

[1] http://www.martinfowler.com/bliki/UmlAsProgrammingLanguage.html

4.2 The need for the project and why it's important

"I thought that there is especially a lot to be invented in area of code creation. It seemed that RAD tools made a progress here: Much of the code has been generated automatically - especially from points 1-3 above. I loved it. It was 70% of the code or so. But then I realized that specifying the remaining 30% of code took 70% of my time. What was the progress, I asked myself. Before RAD tools, I spent 100% of my programming-time in hand coding, now it is 70%. And this is not much less than 100% !"

Grzegorz Falda – the project initiator "The Vide Story"

This was what Grzegorz Falda, the initiator of this project, experienced working on the software for the large Insurance Company. The struggle with time consuming hand coding made him to look for solutions. *"I realized that the proper understanding of the system could be supported by visual representation of the system code. I came to a conclusion, that a new language should be developed, to visually cover the logic of data intense applications - not only the design of data structures, end user screens, printouts and events. I investigated table-driven programming, which offers an interactive specification of the whole application logic without textual code and with a very strong contextual type checking. I loved it, but I saw that my users do not follow me. They were not able to see the code in the tables. Moreover, the tables were also a problem for me. The code in the tables was not visual, it even did not have a textual representation. Creating the code was a breeze, but I had serious problems with understanding and extending the code, when it became larger".*

The project was a one man vision, but the following quotations demonstrate that the idea is equally important to industry leaders as well as smaller companies and can be deployed in various software development environments. *"The expressivity and the precision of the VIDE language will guarantee that the capability of the Objecteering modelling tool in generating the dynamic part of model will undoubtedly be extended. Moreover the automatic conversion between VIDE language to Activity diagram will result in a significant productivity gain in the area of application development" – Softeam*
"Due to market demands to build more flexible and adaptable solutions. Especially solutions for small and midsize companies that implement processes as complicated as those processes of larger companies, but in addition require adaptations to the specific businesses they implement. In summary the VIDE project helps to investigate the potential (also the economic benefit) of behavioural models to increase the level of modelling in product development in order to increase adaptability and flexibility of software solutions." – SAP

During development of the Project we have realized that there is more to be done than originally thought and there are several potential benefits to software development coming from the improvement offered by VIDE. The project was aimed to be important to a whole range of stakeholders from business users to whole communities such as the Software Engineering (SE) and MDA community.

One of the areas of improvement VIDE has given in modelling tools is the support for business users. Perusal of the MDA literature reveals a focus on the PIM and PSM levels, particularly with regard to automating the generation of executable code. Even though the MDA is supported by a wide range of tools, few offer explicit CIM support, and what support is provided arguably falls short in addressing the needs of business users. For example, some existing toolsets adopt notations such as BPMN. Whilst these provide highly expressive models of business processes and may be suitable for an experienced analyst, they are complex and not readily accessible to general business users unfamiliar with modelling, and, therefore, do not support effective communication. Further, there is little or no consideration of requirements capture and validation, or the development of a specification against which later design and implementation (PIM and PSM) models may be verified. IT projects frequently fail due to an inability to capture effectively user requirements and to support business processes, resulting in software systems that do not have the desired effect in the domain. Hence, the lack of support for these 'upstream' aspects of the development process is particularly worrying.

IT specialists seek for productive tools for software development and, especially its maintenance and adaptation to various target platforms. It is necessary to preserve the work results on specifying the application logic, which requires code generation technology. Modelling and programming need to handle the raising complexity of the code and should reduce the communication barriers for non-IT stakeholders. Overall maintainability of the code and related artifacts brings the need for a possibly seamless tooling support throughout all the MDA-defined modelling layers: from CIM to code.

Realising those requirements i.e. maximizing the reuse of work performed and reducing the communication overhead match the needs of software companies. Faster and more flexible development allows the production of software that is better aligned to the needs of the organisation. The software development technologies need to assure integration with existing software in the context of Service Oriented Architecture (SOA) for service implementation as well as orchestration. From the software maintenance point of view, the quality control solutions are essential. This is going to be important in the current financial climate where CIO's are expecting to introduce efficiencies into their software acquisition and development.

The practical research and prototypes provided by VIDE have potential to make impact - they are important for the future of MDA. VIDE may also impact SE culture by adding syntactic and semantic discipline not only to source code but also to models. From the business perspective VIDE has the

potential to demonstrate how to prevent “throwaway” models by potential reuse of platform independent models. In effect the VIDE approach anticipates the reduction of life-cycle costs of systems. On the more generic level the Project is also important for Consortium members who entered the project with the aim of enhancing their product lines, services or purely to gain knowledge and advance their research in topics of interest. This organization centred motivation did not hamper the creation of a consortium toolset that due to much being available using open sourcing can be used and further developed by software companies.

5 Aims and Objectives

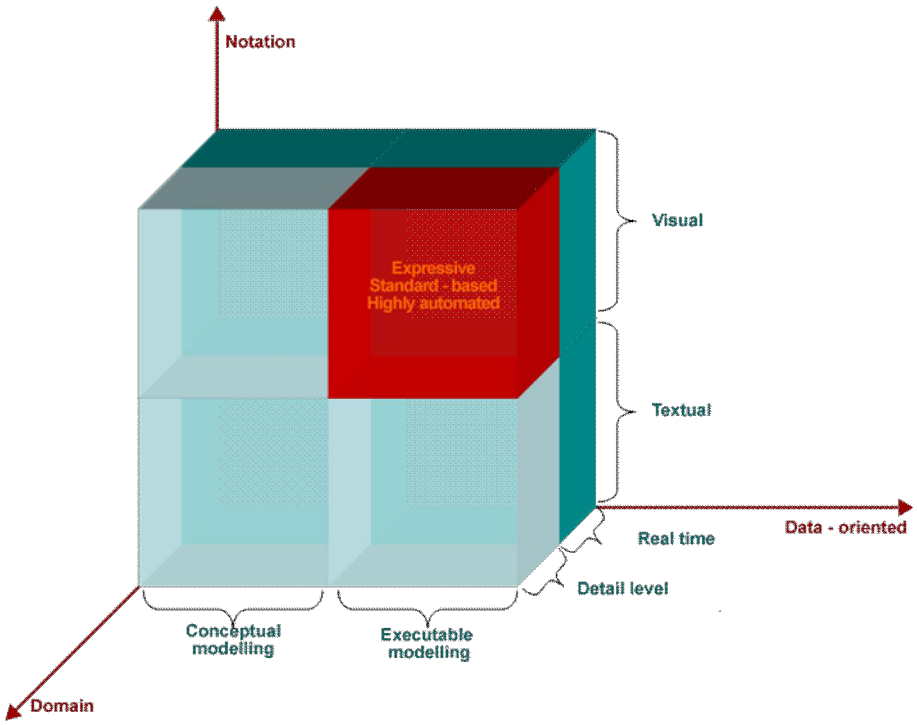
5.1 Aims

There are two main strategic goals of the VIDE Project:

Goal 1: To improve methodologies and tools for application development focusing on data-intense business applications in order to make creation of new systems easier, faster and more accessible to business people.

Goal 2: Introduce the new methodologies and tools to the business world and assure that they will be adopted by industrial leaders

5.2 Vision and uniqueness of VIDE system



The project assumes design and implementation of a system which will enable easy and fast creation of new systems and make the process more accessible to business people. It is aimed at strong standard compliance (especially to the Unified Modelling Language (UML)) in order to build upon existing toolsets and to simplify language adoption. However, although UML is a visual language, it lacks a visual means for specifying the logic of data intense business applications. VIDE is going to fill this gap, and potentially contribute to the further development of the standard. The main perceived benefits of the toolset based on such a language are:

- Expressiveness and user friendliness resulted from the visual syntax and seamless connection with higher level model artifacts,
- High automation of the coding process by following the executable modelling approach,
- Simplification of the development process and increased reuse by shifting most of the development activities to the platform independent model level.
- Streamlining the communication with stakeholders by making the modelling environment more accessible to non-IT professionals.

The key feature of the VIDE language and toolset is the shift of programming activities towards the fully platform independent level grounded in the UML notation. Whilst the standard already provides the necessary vision, there are no existing tools which realise it, and hence the need to be addressed by the VIDE project. Thus the following features are unique to the VIDE project:

- A uniform UML-based language allowing for precise behaviour specification. This provides more flexibility for the software development process and reduces the communication challenges.
- OCL-based query language features which will allow the required functionality to be achieved with a significantly smaller amount of code.
- Queries and imperative statements seamlessly integrated into a single language backed by standardised metamodel.
- Both an aspect oriented approach and quality defect discovery techniques which are moved to the platform independent level to increase reuse and to manage the complexity of applications being developed with VIDE.
- Full coverage of the transition process from informal business models to executable software models.
- Platform independent executable model providing connectivity with existing services and allowing process engine interaction as well as GUI prototyping and testing
- An informal “pre-CIM” modelling tool addressed to business users to provide a starting point for the development process
- Extended CIM-level language based on the BPMN notation
- CIM to PIM transition in terms of model exchange as the starting point for application specification and as a blueprint for executable workflow process design

Such a consistent, seamless and tool supported transition will maximize the input of domain experts to the process of application development.

5.3 Objectives

5.3.1 Relevance to strategic objectives

The project is relevant to strategic objective “2.5.5 Software and Services”. It is intended support the competitive position of the European software industry by providing an extremely productive and easy to use, fully visual toolset for the emerging Executable UML standard. Though the industrial partners SAP and ALTEC are not SMEs, their products are used by several thousand SMEs (notably VARs) for use in the installation, customization and parameterisation of their ERP products and business software solution suites. In this respect, VIDE tools enhance the potential of these SMEs by means of empowering them with additional functions that improve the way they serve customer needs and increase their value creation capacities. VIDE is an open and interoperable platform, that is compliant and builds upon standards (UML/XMI) and successful open source platforms such as ECLIPSE and the Eclipse Modelling Framework (EMF) for tool interoperability.

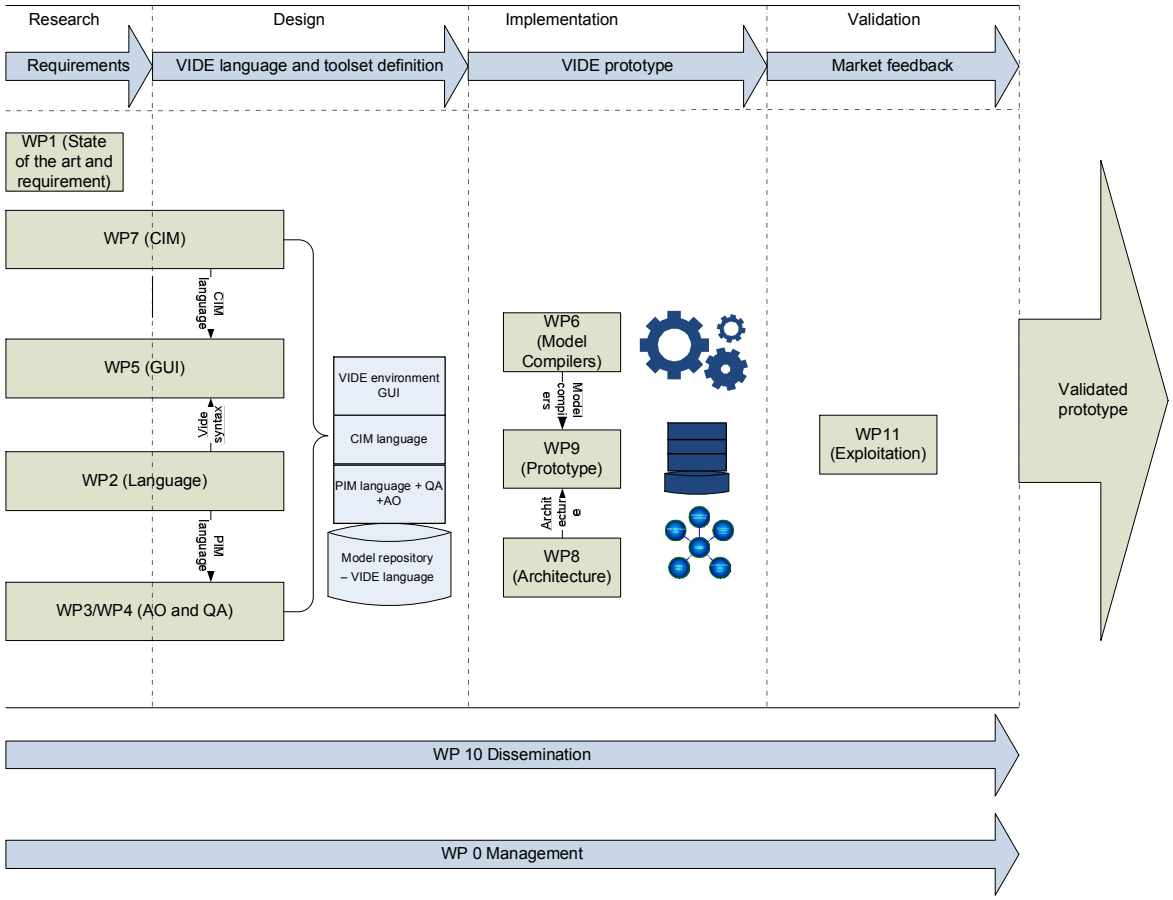
5.3.2 Project objectives

The projects technological objectives were many, but started with several research activities. The first included evaluation of the existing standards, techniques and methods relevant and applicable within the VIDE project. It included standards appraisal and investigation of candidate development environments. The results were to provide the theoretical architectural base for the project. The research activities based on this work had the objective of providing the formal definition of the VIDE language complying to UML action semantics, enabling both visual and textual code representation and resulted in the provision of an early prototype code editor.

To maximise the advantage of complete platform-independent behaviour specification, selected techniques from programming languages were transferred to the PIM level. This included AO modelling and composition techniques and quality defect detection in terms of smells, anti-patterns and design flaws. The mapping of VIDE to the Java platform and to the experimental object-oriented database programming environment ODRA were to be designed and implemented based on the specification of the VIDE to UML Action Semantics and UML Action Semantics to respective target platforms. The Visual User Interface was to be explored; to find out the ways of supporting the developer tasks by applying various visualisations. The CIM level was to be supported as the entry point for application development, and as the source of business process specification to be refined into workflow process definition. The language design and its prototype was to complement respective parts of industrial partners tools, including behaviour specification in UML modelling tool, GUI prototyping tool, legacy data integration tool, Web service engineering tool and workflow management system.
The resulting research findings and software prototype were to be evaluated and complemented with the prospective users addressed resources including a Web library of sample models and an e-Learning system.

6 Approach

6.1 Summary of planned approach



The original plan for the project was research-oriented and strongly focused on the PIM level. The main source of productivity increase was associated with visual syntax at the PIM level and the synergy with Executable UML approach, therefore the plan involved extensive design and implementation work on the visual language syntax and editor, while skipping the implementation of several other research work package outcomes. From the management point of view the project was to be managed like any other industrial project.

There were several reasons for this approach which was shaped by the following priorities:

- The project must adopt best research results, best practices and standards to avoid any rework and to assure future adoption.
- For better understanding and communication with future users the software should be placed at the level of abstraction business people can understand. Better communication between IT- and

business- specialists will lead to avoidance of misunderstanding, loss of time and resources and lead to systems that better address the needs of the end users.

To enable the creation of reliable UML-based tools for software development based at the PIM level, the project will combine and extend the following core approaches and technologies:

- Action programming – which is the core of Executable UML
- Aspect Oriented Composition – which is a popular modern approach to programming which has the power to be shifted to the more abstract level of PIM-based programming
- Quality Assurance which decreases the number of defects and should be applied to the platform-independent level to become even more effective
- Visual user interfaces – which makes programming more user friendly and accessible to business people and should be applied to the platform-independent level at all stages of application creation and maintenance: prototyping, programming, debugging and documenting.

Although the four core technologies described above do not include all areas to be moved to the PIM level, they feature important trends vital and sufficient for user-friendly creation of reliable business applications.

The project also aimed to research the areas of creation of software involving business process centric software development. This would enable business processes already described in CIMs to be easily supported by the VIDE technology. Existing Workflow systems were to be augmented by the capability of creating prototype applications supporting the related businesses.

To enable the creation of applications for verification platforms, model compilers must be created. A model compiler is a tool for automated creation of executable applications out of platform independent models – removing need to change the specification before changing the target platform. It would lead to creation of applications for several platforms in a fully automated way, which brings flexibility to system development. This would help companies to become vendor–independent and more flexible in changing software platforms responding to market development.

For the deployment of the system to the business world, VIDE has partners who are industrial leaders. VIDE puts strong emphasis on disseminating the outcomes of the project amongst business and IT students as they are the main vehicle to bring the knowledge of VIDE to companies later on. Moreover, VIDE has been receiving feedback from the business community throughout the duration of the project to assure compliance with the needs of the business world. The early prototype and the industrial use cases are examples of these actions.

The central role assumed for PIM level visual notation influenced the scope of the early prototype of the VIDE code editor intended to validate the visual syntax. This approach has been soon verified and amended.

6.2 The final approach and findings

Alterations to the scope were introduced through the duration of the project to accommodate findings revealed by the research and prototyping work, while the constant element were the strategic goals. Three Annex amendments have been prepared, submitted and accepted in the course of project. The amendments can be classified within the following categories:

Approach to scope of work

In the course of the state of the art analysis it was discovered that certain parts of the underlying UML metamodel and its executable semantics remain relatively immature. Hence it was necessary to perform more in-depth work on the language design and validating its results by creating an early prototype. For similar reasons, the complexity of work and the expectations of the Industrial Partners towards the VIDE tool, that were not originally planned but were necessary for final adoption of the tool, have lead to additional tasks and research work such as the integration between UML and OCL metamodel in order to use the latter as a query language in VIDE. This choice allowed the scope of the VIDE language to retain the abstract syntax covered by OMG standard specifications. The aim of the early prototype became the implementation of the language in terms of its actual integration into UML and its execution engine realisation. Eventually, two visual syntax proposals were designed and evaluated in the further phase of the project. The work on visual interface and business modelling revealed the need for a more approachable modelling solution for business people, allowing their direct involvement in development work. To address this, an additional layer called pre-CIM supporting domain analysis was designed and implemented in the final prototype.

The implementation work has been increased by including the WP's that were not initially planned to have their findings implemented. As a result the following new features were delivered:

- A Domain Analysis Tool prototype covering the initial interaction of a user with the VIDE system at the CIM level. A business user or a business analyst will be able to describe the problem domain in terms that they understand. In subsequent steps it supports a user in refining their ideas from the problem domain towards a business process model. The implementation of this functionality was originally outside of the project scope but in the course of research and investigation the need for supporting non-IT users this way was revealed.
- A PIM Prototyping Tool providing wizards that support the developer in creating the transition from CIM models to application elements specified at PIM. This branch is intended to protect the modelling effort done at CIM level by reusing and tracing its contents to respective elements at

PIM. Originally the PIM prototyping concept assumed only a top-down approach to designing behaviour elements at PIM level but the final version assumes integrating the CIM model as a base for PIM level application development.

- New functionality for GUI prototyping. An important purpose the partners identified of executable PIM is the ability to use it as a mean of communication with final users. In this case the ability to provide at least a rudimentary GUI prototype is essential to make the model tangible to the user. While originally the TNM:WebFace tool was planned to be merely a consumer of the VIDE PIM language (to ease producing Java code for its purposes), the resulting idea, is to use it for PIM level GUI prototyping. The tool makes it possible to design GUI forms and to execute them with interaction with the executable PIM.
- An Aspect Oriented Modelling tool. This functionality is to realise the WP3 results basing on the Visual Editor. This involves language syntax extensions in the editor and realisation of the Aspect Weaver component. In the original plan no resources were assigned to implementing the research results in AO.
- A Quality Defects discovery and visualisation tool providing defect discovery working over the PIM repository and the respective visualisation of quality defects in the Visual Editor. In the original plan no implementation was allocated for the quality defect discovery research results.
- A standard based transition from the CIM model to orchestration at the PIM level (realised by the export from the CIM Editor to Rodan's OOWF) allowing any workflow tool using XPDL2 to benefit from the VIDE CIM editor. The functionality broadens the future usability of the project work.
- Tighter integration between the Visual Editor and Textual Editor. The research work and subsequent experimentation with the visual and textual editors, lead to the solution consisting of a visual editor, textual editor (both dealing with the same underlying metamodel constructs) and the Visual Expression Builder, that may be used in combination.
- Providing a Model Execution Engine with Web Services and RDBMS connectivity. In the course of research work two basic ways of integrating with externally developed software have been identified: consuming and providing Web services from the code created from VIDE models (this involves also the interaction with GUI prototyping module) and consuming relational database data sources by the code created from VIDE PIMs. Apart from handling those connections in the model and in editors, it was desirable to have this covered with the Model Execution Engine functionality. This was realised by adapting ODRA which originally lacked that functionality.
- An open source Java model compiler. Initially, it had been planned to implement the Java compiler as an Objecteering MDAC (MDA Component or plug-in). But the reviewer's recommendation as well as Partners' requirements revealed in the course of the architectural design, show the need of an open source availability for the Java compiler. Because it is not possible to produce an open source implementation on top of Objecteering, Softeam decided to implement it as an Open source Eclipse plug-in.

To sum up, the above listed features contribute to the following top-level benefits. Firstly, the toolset is much more than initially planned and oriented towards non-IT users (by provision of informal means for describing the problem domain and providing assistance for gradual refinement and more complete integration of the CIM perspective with PIM modelling tools). Secondly, the plan update allows better use of the research results in the prototype being developed which contributes to better support for developers. The new features of the tool are likely to lead to better market adoption.

Approach to technology

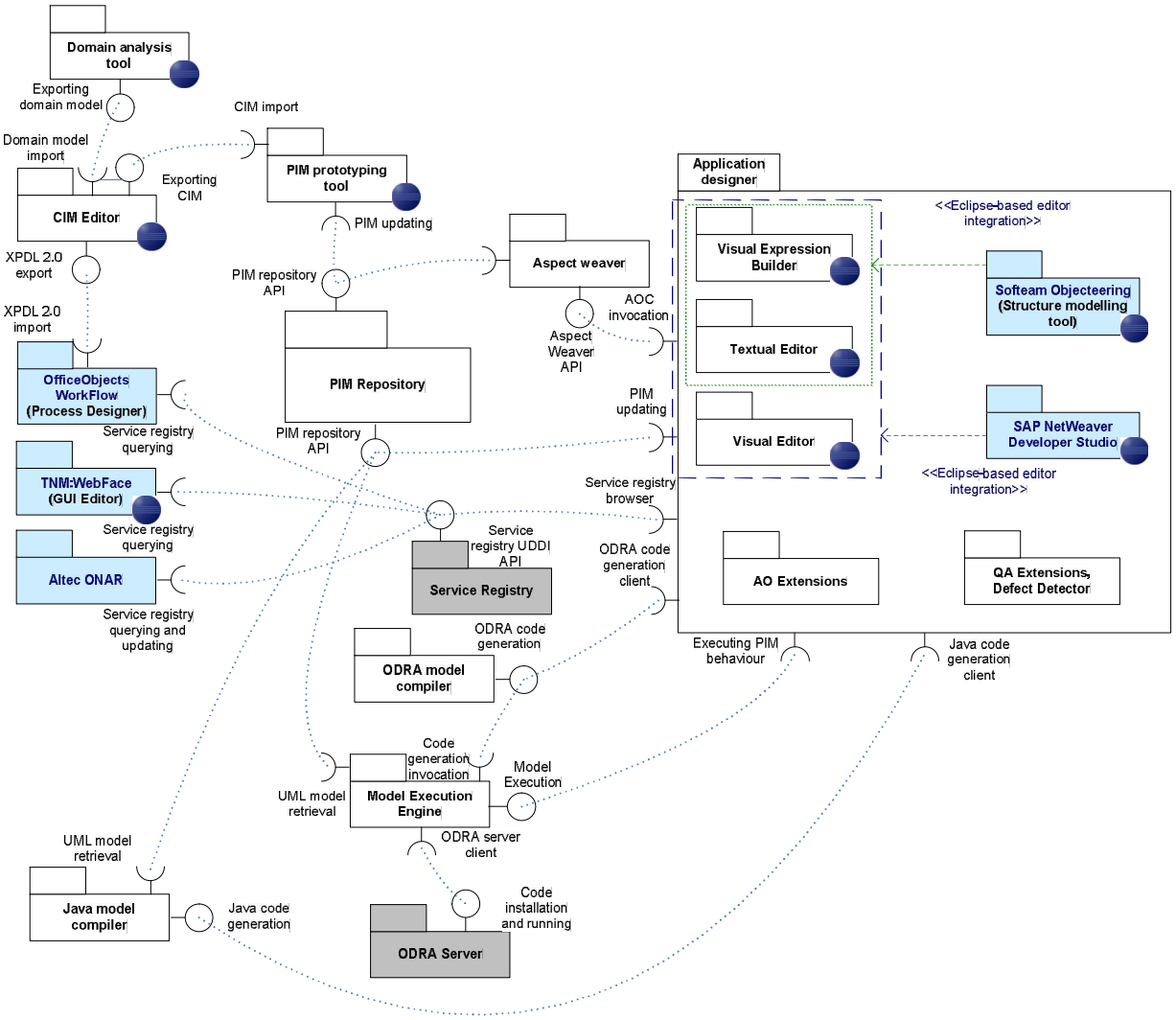
The evaluation of programming tools available confirmed the selection of Eclipse as the development environment to host the VIDE prototype. The MDT Eclipse-based implementation of UML and OCL metamodels was found out to be optimum from the standard compliance and dissemination potential point of view for the VIDE language implementation. For the visual editor design, the GMF Eclipse framework was adopted and for the Java model compiler implementation the openArchiectureware framework was chosen. These tools brought significant advantage in terms of productivity for solution prototyping, however their steep learning curve had to be accommodated in the updated work plan.

As most of the integration of VIDE is based on Web Service technology, the PIM level tooling was provided with Web service import and export functionality, including a browser for UDDI registries. The selection of Eclipse and popular modelling frameworks eased the development of the functionality handling the Web service description to UML transformations.

Approach to management

The techniques that were initially adopted, which were sufficient for managing industrial projects were challenged with issues resulting from communication in the multileveled, multinational project which was conducted in a semi- virtual environment. It had to take account of the varying needs of particular partners, some ambiguities of the workplan and simultaneous interdependent research at several levels. The initially planned communication pattern proved to be insufficient, hence a new one was applied based on tools including the use of a Wiki and frequent teleconferencing, which were enhanced when needed by a formal decision making process. Much attention was given to constant risk monitoring and change management. The initial plan, tasks and budget assignment were changed several times during the project. The flexible managerial approach developed during the first year of the project resulted in resolving most of the technical issues and simplifying the procedures without hampering the quality of the project work. Even though some new challenges appeared (namely quite high turnover of project participants and general resource problems experienced by some of the partners) the management went much smoother than during the first year of the project and the consortium were able to finish the project on time and within budget.

7 Technical design and standards



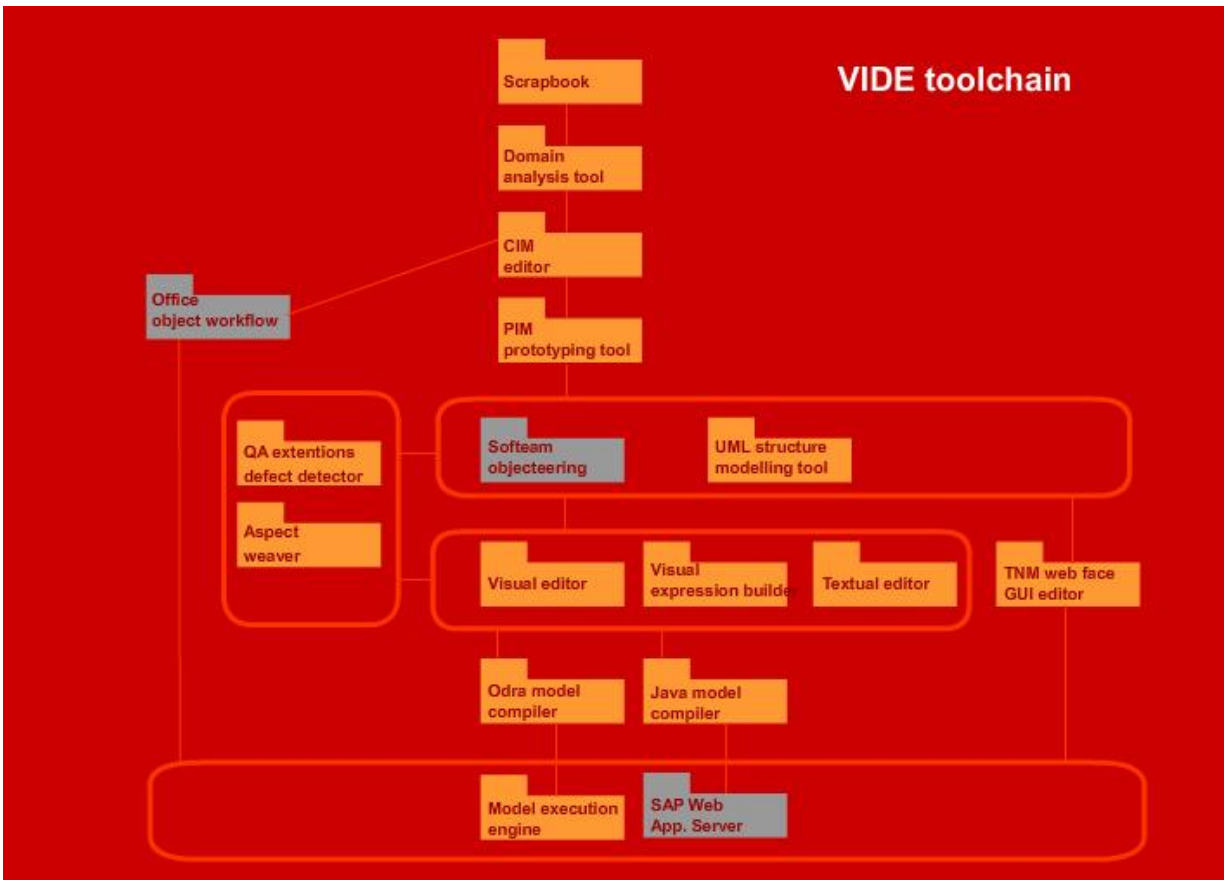
VIDE assumes model interchange across toolset components and layers and aims at integration within the common development environment platform – Eclipse. All the components constituting the VIDE prototype are developed in Java and are based on Eclipse frameworks. Components marked with the icon involve editors that integrate with Eclipse and hence offer a uniform look & feel. Most of the inter-component integration is based on the model interchange, which uses standard formats mostly based on XMI. The interactions of the executable model or of the code generated from the model with other applications is realised through Web services.

VIDE has extensively reused existing standards in order to make the adoption of its contributions easier and to investigate the suitability of certain standards by extending their areas of application. Particularly, VIDE tools support or use the following standards:

- Business Process Modeling Notation
- XML Process Definition Language
- Unified Modeling Language
- Object Constraint Language
- XML Metadata Interchange
- SOAP and Web Services
- Web Service Definition Language
- Universal Description, Discovery and Integration

8 Ensuring adoption

From the very beginning of the project a wide dissemination plan has been implemented in order to disseminate the developments of VIDE to industry and more importantly gain the feedback that would allow us to shape our tool. We have also researched the market to monitor the trends that may be important for project and analyzed the commercial tools available in the market. We have realized that the tools existing on the market expose a significant gap where executable modelling of data intensive applications is considered.



There are a number of key points which show how VIDE may be differentiated from the opposition. The VIDE toolchain was compared to a number of toolchains available in the market place, which were selected for their similarity in scope to VIDE. (see following tables). The features which were evaluated reveal that there is considerable value added in the VIDE toolset. These features can then be further summarised to show how the final product can be placed. These include:

- Explicit concern for accessibility to business users, through various ‘pre-CIM’ tools.
- Integration of requirements, specification and CIM modelling; business critical parts of the MDA that are often overlooked.
- Visual notations for specifying system behaviours, supporting the automatic generation of more code than is currently possible based on just class models.
- Enhanced traceability, from informal domain information, through various pre-CIM and CIM models, to PIM.
- Integration with existing leading software tools.
- Improved support for the development of data intensive business applications.
- Novel visual and textual code editors, improving efficiency of development processes and enhancing communication.
- Inclusion of an aspect-oriented approach on the PIM level.
- Incorporation of approaches for quality defect discovery and visualisation.

	Evaluation Criteria	IBM	Borland	Visual Paradigm	Telelogic	Eclipse	Artisan	Open Ames	Magic Draw	Select	MID
Pre-CIM	• Creating and storing descriptions of the domain (including descriptions of business requirements and operational procedures)										
	• Selection of desired scrap text from the domain description to create a visual informal scrap model.										
	• Selection of scrap model elements for further analysis and creation of an informal analysis model										
	• Refinement of domain model elements into roles, activities or data objects.										
	• Selection of analysis model elements for use in creating an initial business process model										

Key:

No cover	
Partial	
Complete	

	Evaluation Criteria	IBM	Borland	Visual Paradigm	Telelogic	Eclipse	Artisan	Open Ameos	Magic Draw	Select	MID
CIM	• Allows the import of un-structured requirements information										
	• Allows the creation of Business Rules from both data structure and organisational structure model elements in addition to natural language										

	• Enhances business process modelling using extended BPMN
	  • Creating such things as multi-media artifact information
	  • Creating data elements, organisational data and business rules inside the process. (These can be used singly or combined.)										

	• Allows the creation of organisational and data structures using elements helpful to the non-IT person										
	• Exports information from the process model as XPDL (for a workflow engine.)										
	• Uses the export of a process model as a blueprint to allow creation of UML2 diagrams										

Key:

No cover	
Partial	
Complete	

	Evaluation Criteria	IBM	Borland	Visual Paradigm	Telelogic	Eclipse	Artisan	Open Ameos	Magic Draw	Select	MID
CIM TO PIM transition	• Automatic creation of class model from business process model (by use of heuristics for BPM-to-Class mapping).										
	• Automatic creation of activity chart from business process model (by use of heuristics for BPM-to-Activity mapping).										

Key:

No cover	
Partial	
Complete	

	Evaluation Criteria	IBM	Borland	Visual Paradigm	Telelogic	Eclipse	Artisan	Open Ameos	Magic Draw	Select	MID
PIM level	• Model execution at the PIM level									?	
	• Textual language and editor for UML action modelling									?	
	• Visual language and editor for UML action modelling										
	• User friendly action modelling										
	• OCL used as a high-level expression and query language									?	
	• OCL query expressions visualized									?	
	• PIM-level aspect-oriented modelling										
	• PIM-level quality defect detection										
	• PIM-level publication of web services - modelling and execution										
	• PIM-level consumption of web services - modelling and execution										
	• Platform independent means for GUI development										
	• Availability of a logical structure of GUI										

Key:

No cover	
Partial	
Complete	

	Evaluation Criteria	IBM	Borland	Visual Paradigm	Telelogic	Eclipse	Artisan	Open Ameos	Magic Draw	Select	MID
PIM/PSM level	• Code generation of behaviour from PIM (method/constructor bodies)										
	• Complete and compile-ready code generation (including web services)										

Key:

No cover	
Partial	
Complete	

In addition several “enablers” that include an online training course and comprehensive “cookbook” were created. The open-sourcing of software combined with the creation of an on-line models library will encourage the SME community to experiment with the toolset.

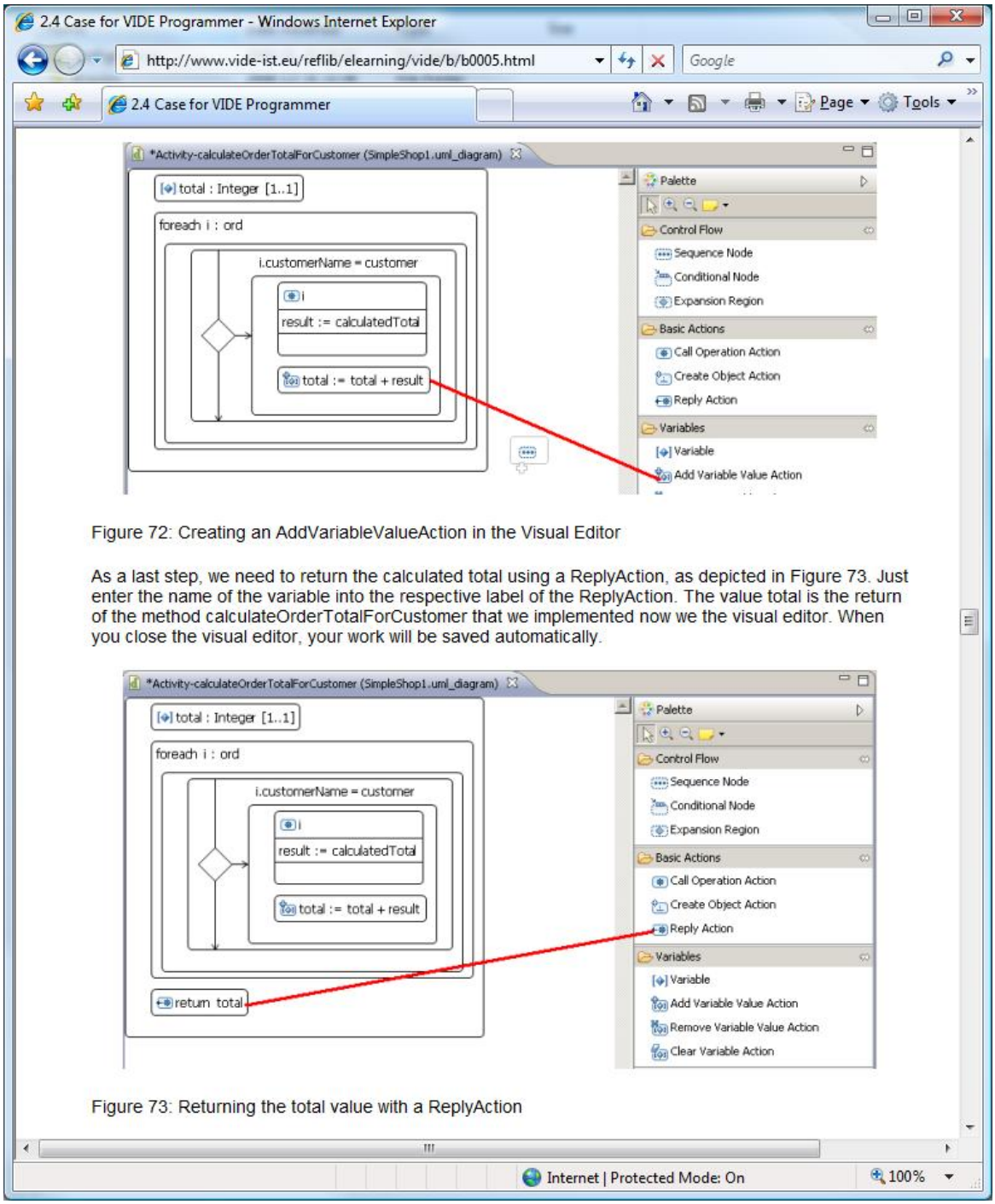


Figure 1: VIDE e-learning system screenshot

Figure 2: VIDE industrial use cases in the reference library

Some partners (PJIIT, for example) have made the source code of the VIDE 'core' components available whilst others have implementations available and are prepared to discuss future exploitation with interested parties on an individual basis.

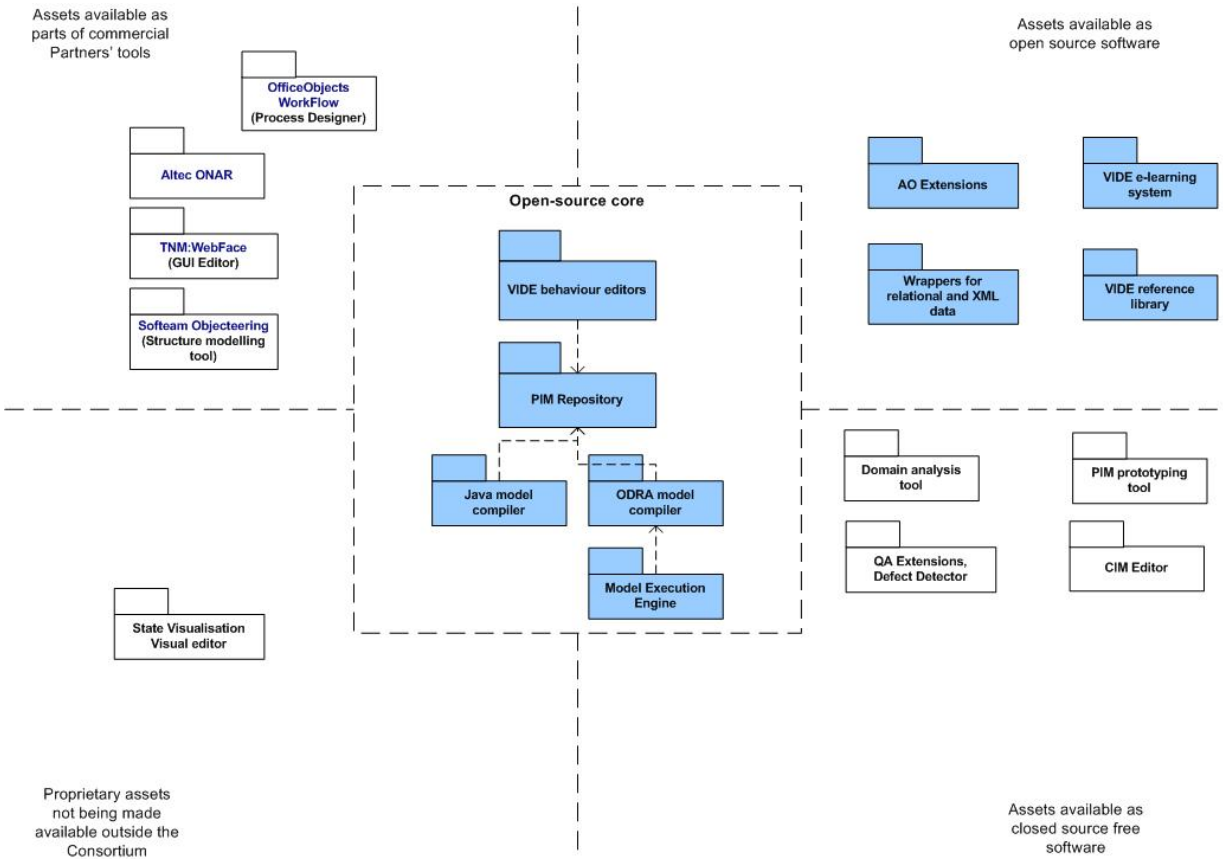


Figure 3: VIDE software components availability

The potential benefits of open source are well known, as is the potential for partners to build a service based income stream on the basis of expert knowledge, through distribution, support, customisation and enhancement, and branding.

The Open source part of the tooling supports the basic PIM level behaviour specification performed in the context of UML class model and involving textual VIDE programming and Visual Expression Builder. The models produced this way can be executed using the Model Execution Engine component or transformed into generated Java source code.

This open-source core can be further extended by several freely available closed-source VIDE components. This way the development process can incorporate the CIM level and its transitions towards PIM as well as the PIM-level quality defect discovery.

8.1 Example

To present the above features of the VIDE software an example is discssued:

The Sales Scenario focuses on sales processes of enterprises selling one or more products. It involves different aspects, ranging from opportunity management to quotations to customers, sales order processing and invoice processing.

From a customers point of view the functionality of the Sales Scenario is described as follows:

1. A field representative of a manufacturer of computer hardware receives a message on his PDA, telling him that company X is planning to replace its complete system in the next quarter. The company has budgeted substantial financial resources for the replacement.

2. He immediately enters this information in the system, i.e., master data of the potential customer, including budget estimation, description of sales opportunity, sales volume, and timeframe of the opportunity.

3. The Opportunity object is created in the system and evolved by the assigned employee until it reaches a go/no go decision by sales management.

4. Another employee of the sales office creates an offer using the Quotation module, which automatically generates a quotation template based on the sales opportunity.

5. Based on the categorisation of the prospect in a Customer Group, estimated sales volume, and sales probability, the Individual Prices module is used to calculate a discount for the customer, which is included in the quotation.

6. After the sales office has contacted the customer and received an order, the system automatically converts the quotation into an order upon mouse click using the module Sales Processing.

7. To check the creditworthiness of the customer, a Credit Check is performed during sales processing by interacting with the Payment module.

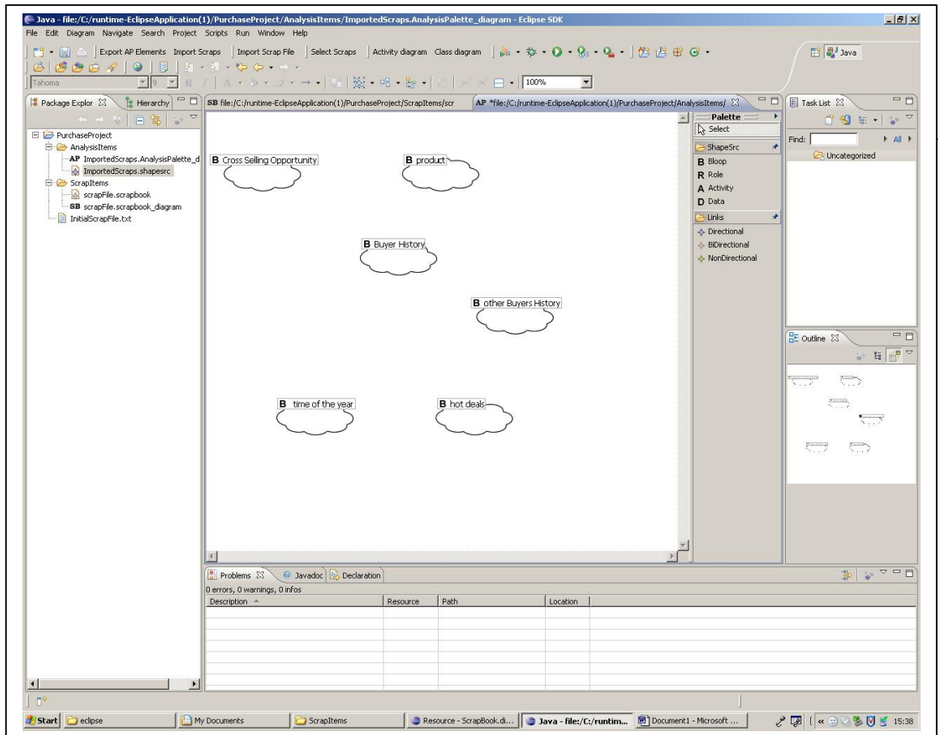
8. An (optional) Availability Check is performed to check warehouse stock for required capacities.

9. The availability-to-promise check requires interaction with warehouse management (Stock). In case of Multiple Stocks only those warehouses sufficiently close to the shipping address are included.

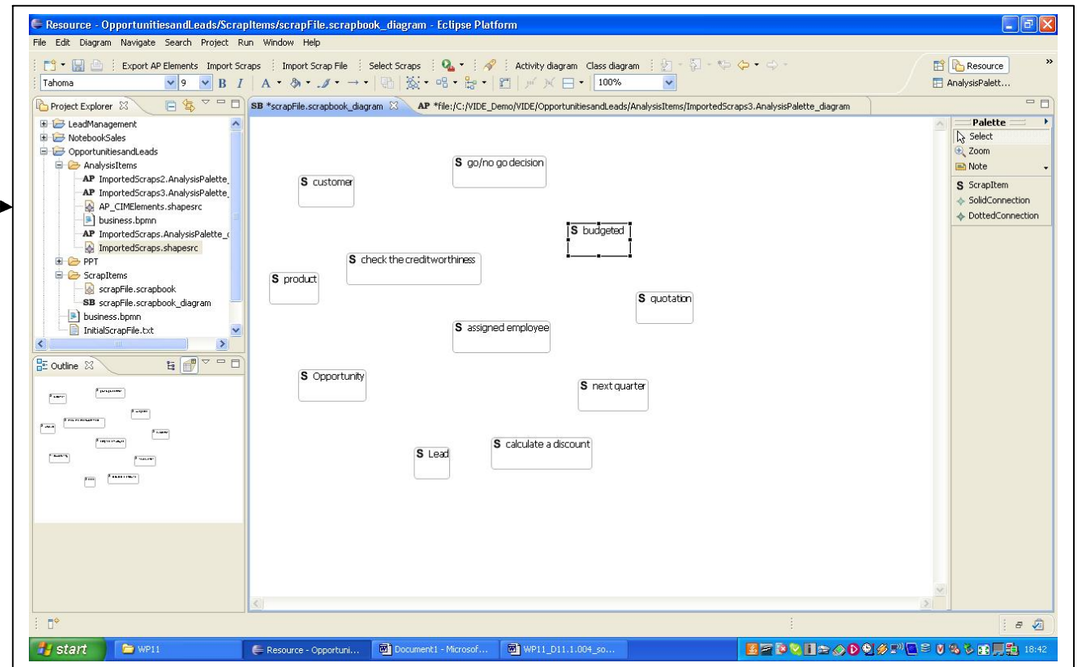
10. In cooperation with the Delivery module, the order is split into separate orders for each involved warehouse, which have to be scheduled appropriately.

11. If Payment is to be integrated into the process, it would be activated automatically upon creating a binding sales order. Depending on the method of payment offered by the system and selected by the customer, an automatic debit transfer from the customer's account can be triggered (Payment Card), an invoicing document can be attached to the delivery (Cash On Delivery), or Invoicing is activated for later settlement.

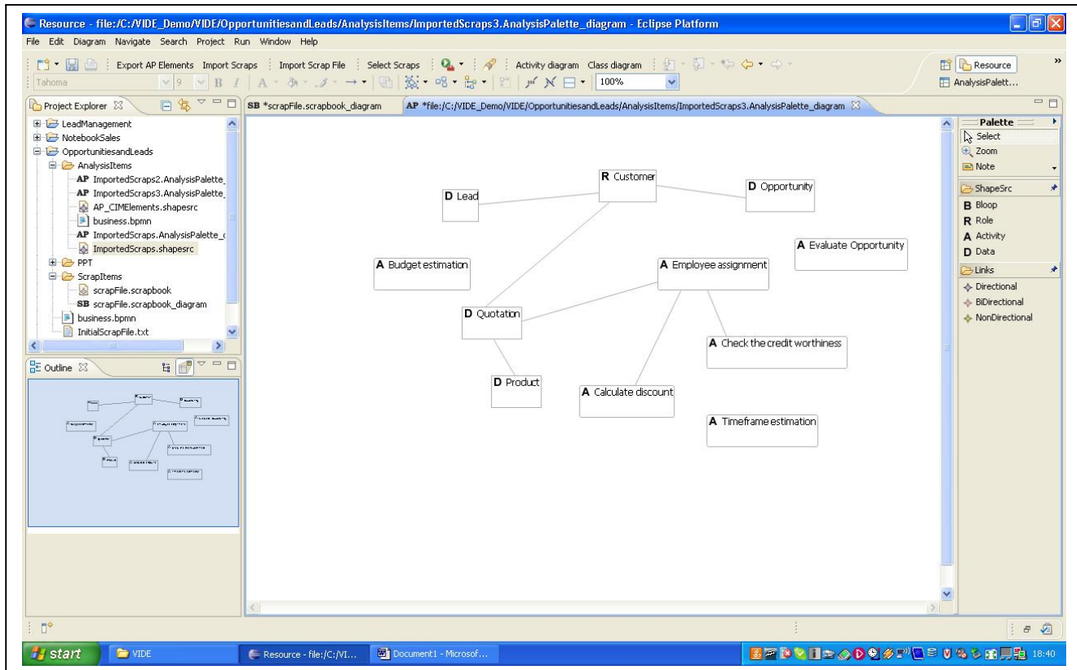
12. The order status is set to "completed" by an employee as soon as it is delivered to the customer.



Initial cross selling 'bloop' model within the domain analysis tool

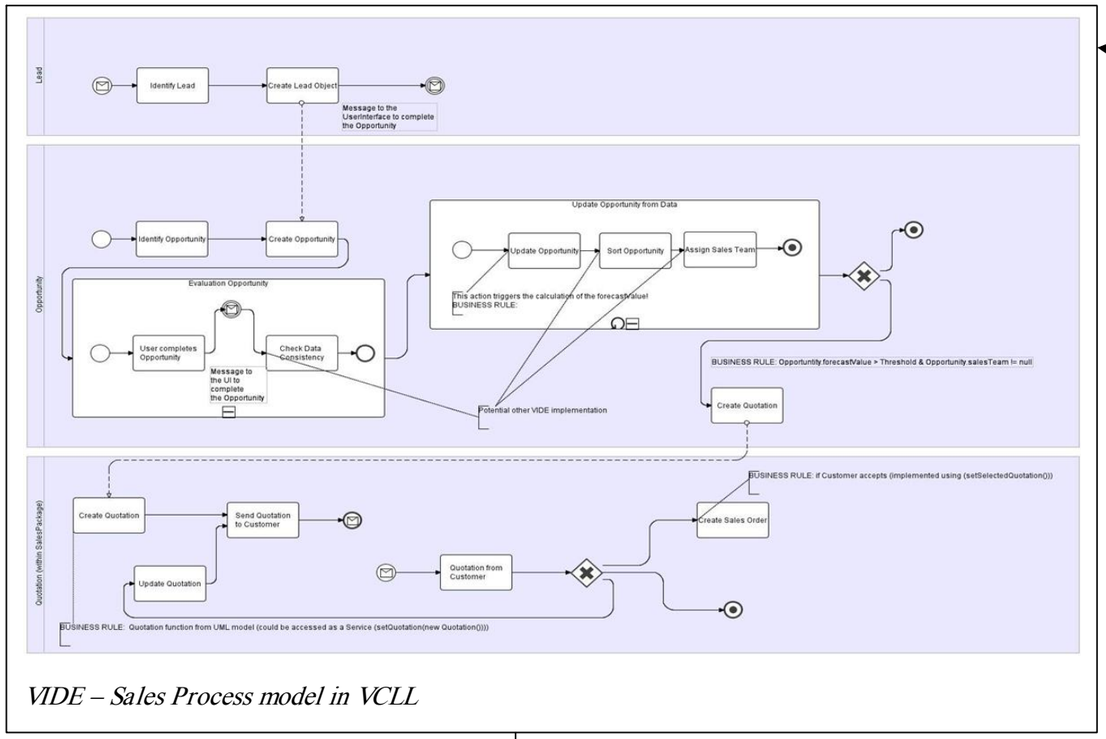


Selected Scraps in Scrap book editor

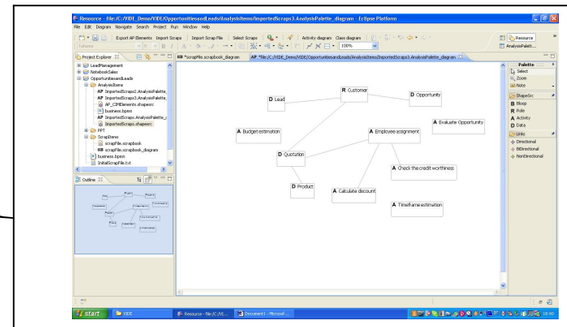


VIDE-Informal model in Analysis Palette

Development can be started with an informal model of domain items called *bloops*. The items can be identified by extracting them manually from a natural language description. Next, the concepts can be refined by classifying them into roles, activities and data and by setting relationships between them.
The simplicity of this model makes it approachable for business users who can actively collaborate with business analysts here.

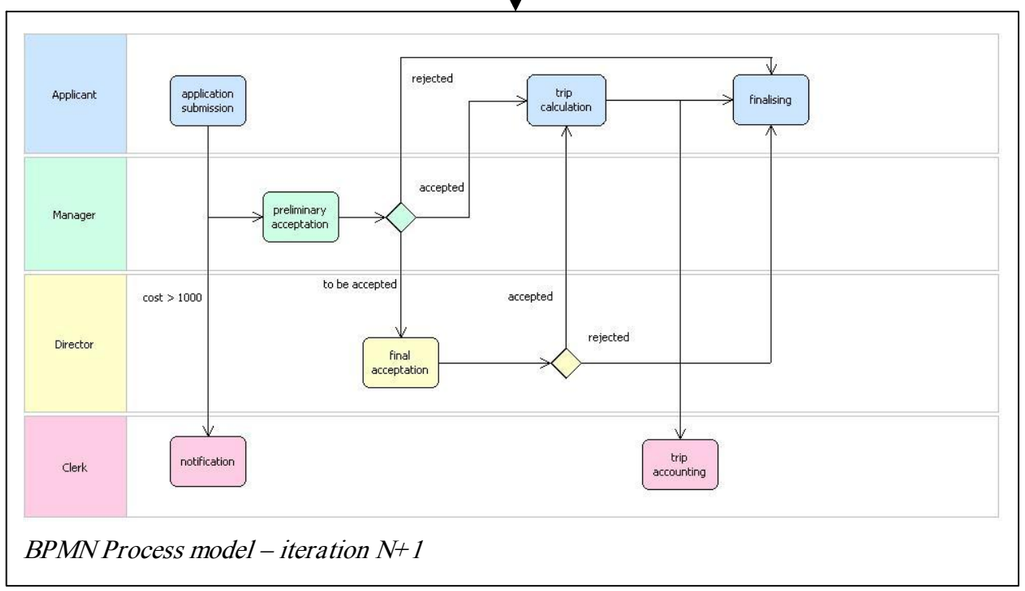


VIDE – Sales Process model in VCLL

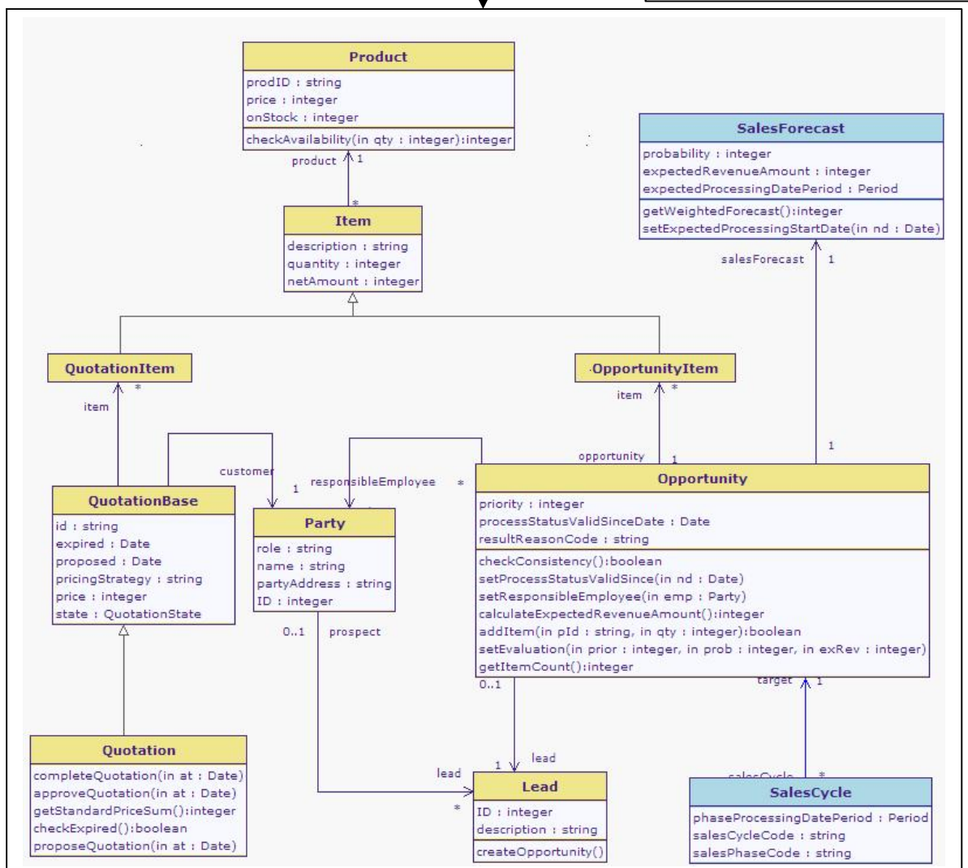


VIDE-Informal model in Analysis Palette

The informal analysis model can be subsequently refined by a business analyst into a Computation Independent Model. The process view that is one of its constituents provides a base for specifying executable workflow processes.
Another path from the analysis model is a refinement towards application structure described using class diagram. Selected items from the analysis model constitute a “first cut” PIM of the application being developed.

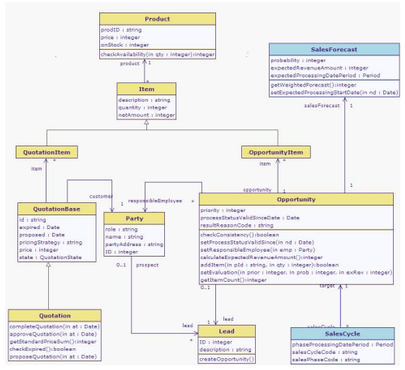


BPMN Process model – iteration N+1

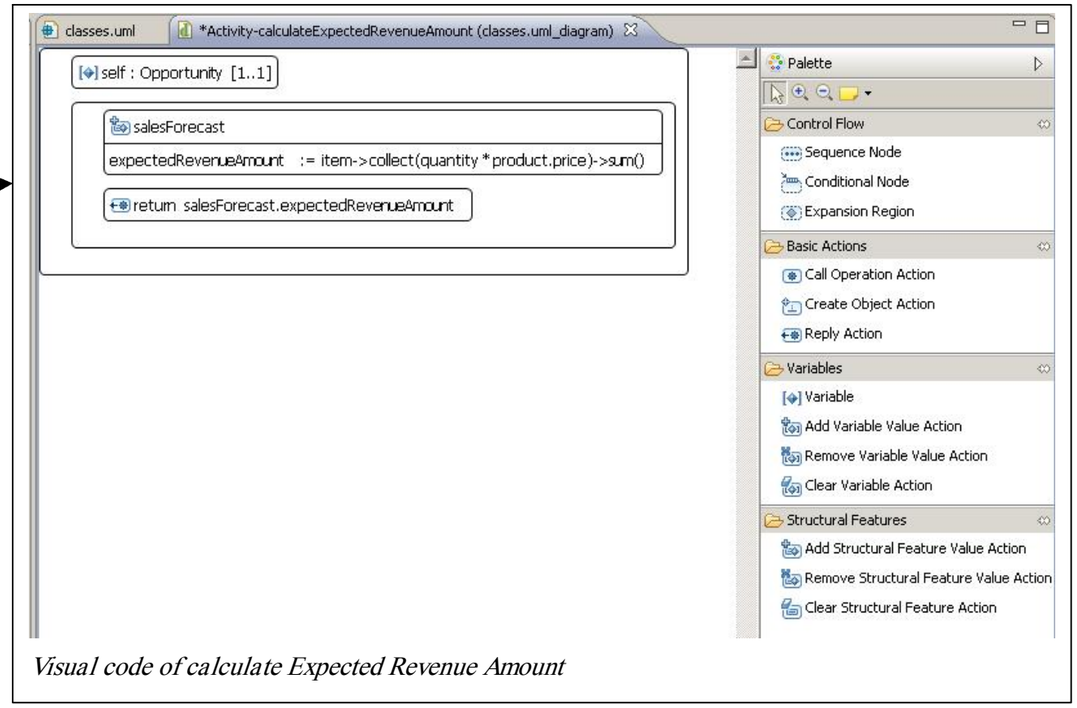


Class diagram (focus on `Opportunity`)

Class diagram is the foundation of VIDE PIM. For the operations included in the classes, a complete behaviour can be specified using VIDE visual or textual syntax covering both imperative statements (that represent UML actions and activities) and powerful query expressions.



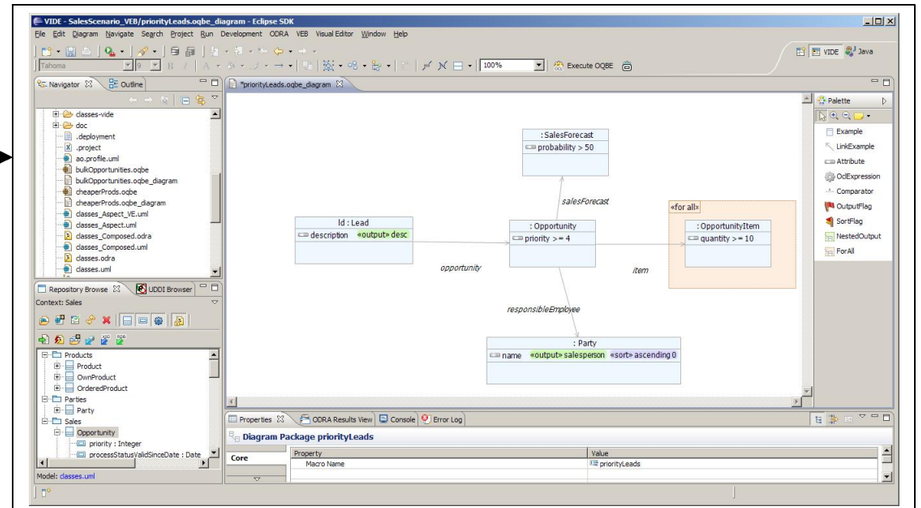
Class diagram (focus on `Opportunity`)



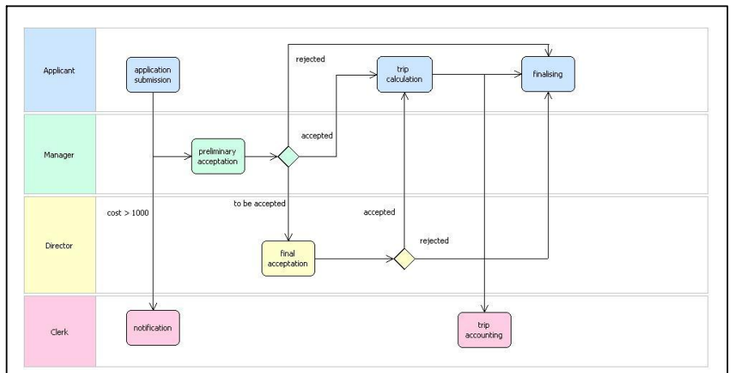
Visual code of calculate Expected Revenue Amount

```
context Sales::Opportunity.calculateExpectedRevenueAmount body {
    salesForecast.expectedRevenueAmount :=
            item->collect(quantity * product.price)->sum();
    return salesForecast.expectedRevenueAmount;
}
```

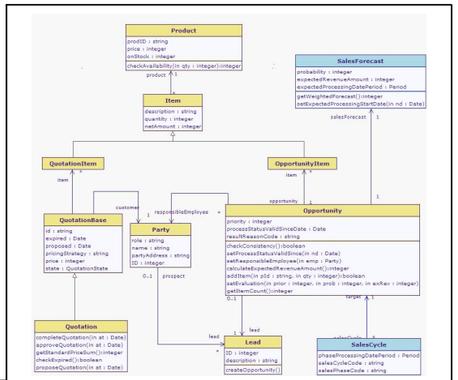
Textual code for `calculateExpectedRevenueAmount`



Visual expression builder

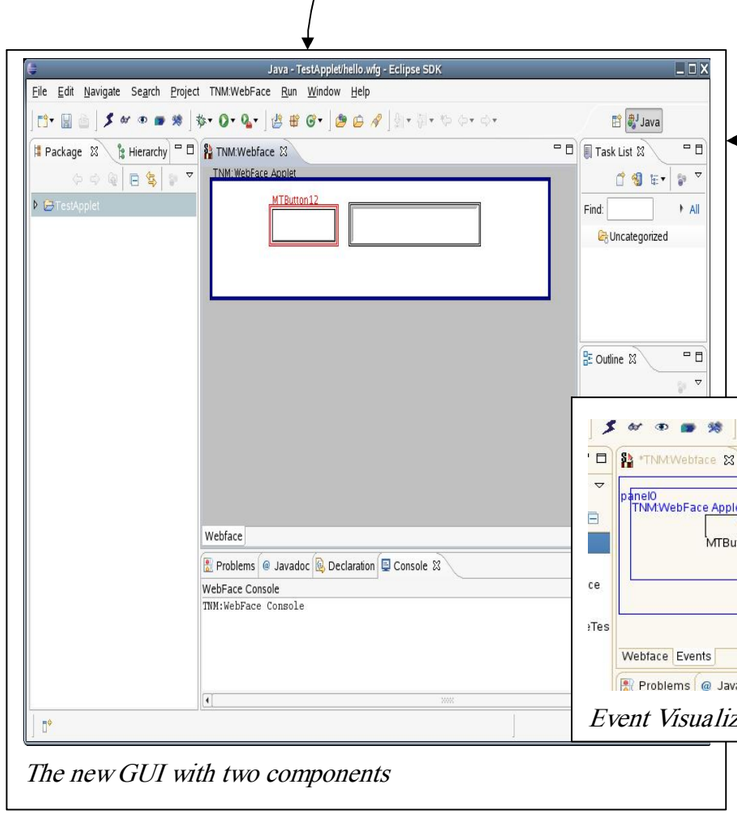


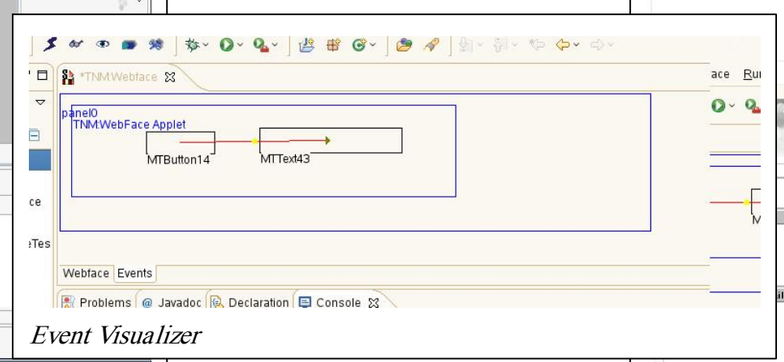
BPMN Process model – iteration N+1



Class diagram (focus on `Opportunity`)

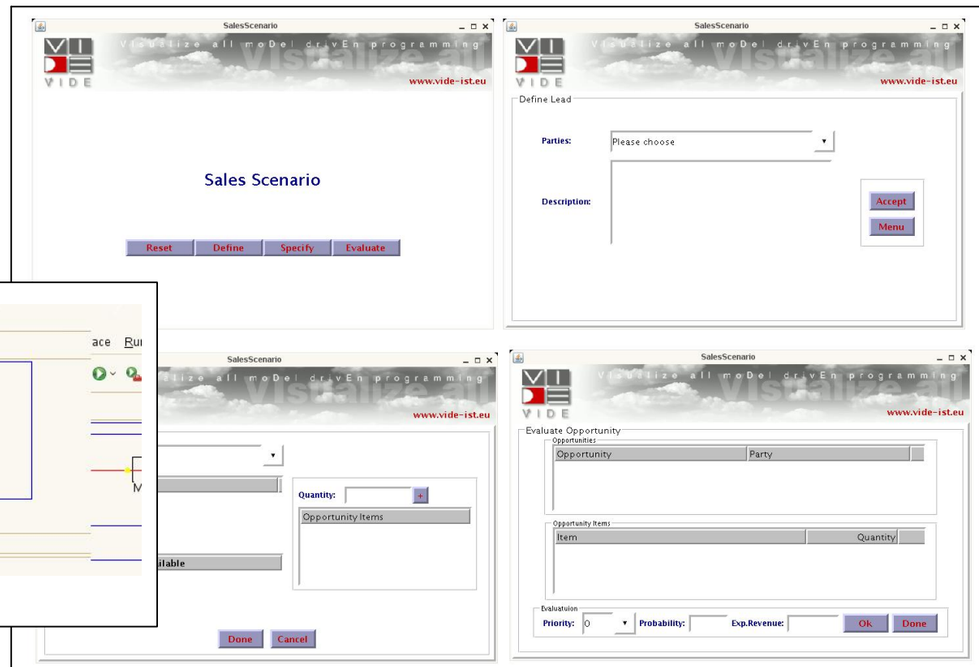
With the GUI builder component by TNM a presentation layer for application as well as for the human tasks defined in the workflow process can be developed declaratively without the need for coding in a platform-specific language.
Not only the GUI elements but also the behaviour logic behind them can be specified visually.





Event Visualizer

The new GUI with two components



Some screens of the Sales Scenario application GUI

9 Outputs and Results

The Project results may be divided into the following categories:

- Research results
- Evaluated software chain
- Influencing standardisation.
- Integration of the methods and concepts into the partner's tools
- Enablers
- Several areas for further development specified.

9.1 Research results

The research work resulted in 19 technical contractual deliverables including 4 tooling prototypes, 4 internal deliverables, several conferences, publications and whitepapers.

The research results include:

- The extension of the CIM-level modelling framework,
- A UML standard compliant PIM-level executable modelling language including textual and visual syntax,
- A PIM-level aspect-oriented modelling solution,
- A PIM-level defect-discovery tool,
- A UML model execution engine and Java code generator.

The design solutions are backed by working prototypes that use the common Eclipse environment with its frameworks for graphical modelling, Web service support and metadata management.

The research and implementation work produced a number of observations and findings that can inform OMG standardisation efforts in modelling and in object-oriented databases.

9.2 Evaluated software chain

The project recognises the potential of storing system behaviour specification in the form of abstract, platform independent, standardised models. Hence it is argued and demonstrated in the prototypes that the tooling should provide a traceable support through all the modelling layers – i.e. a tool-chain from CIM to executable code. From this point of view a number of unique features designed for and demonstrated in the VIDE prototype were identified in the course of evaluation against 10 analogous toolchains available on the market.

Another area of evaluation, performed using workshops and comparative analysis was the investigation of individual features of the VIDE considered central for this project. This included the new modelling features at the CIM level, the language for specifying behaviour at the PIM level, and PIM-level functionality for defect detection and aspect-oriented modelling. Although the scope of the evaluation was

too small to provide strict evidence, the experimentation suggests significant benefits in terms of productivity compared to traditional platform-specific coding.

9.3 Influencing standardisation

VIDE extensively reuses existing standards, but also applies them in a novel way. This refers especially to PIM level and the core standards underlying this layer of VIDE – namely UML 2.1 and OCL 2.0. The combination of several factors allows VIDE to provide new insight into those standards:

- Structured Activities and Actions are relatively new units of UML specification. Hence, although they are already standardised, they are not always present in UML tools and – if so – are rather seldom used. Hence, the intense use of those elements in the VIDE language, will provide important feedback and may indentify shortcomings or bugs of the current specification. Those units have only abstract syntax specified. This may be an obstacle for broader adoption of that part of UML, and this fact has already been observed by OMG. Hence a work on standardisation of a concrete syntax has been recently initiated and a “Concrete Syntax for a UML Action Language” RFP, ad/08-08-01 has been issued. The implemented prototype of the VIDE language provides a valuable input directly relevant to that initiative.
- The assumed area of application for VIDE – that is, data intensive applications – brought the need of going beyond UML Actions for representing expressions. Otherwise the language would be too low-level compared to the means offered by its target platforms. To address this, but at the same time keep the language standard-compliant, OCL was introduced as an expression language in VIDE. This constitutes a novel application of the OCL language. Development of VIDE provides important observations on both OCL expressiveness and usability in this role, as well as its interoperability with UML Actions. It was observed that integration of those two languages would allow the significant reduction in redundancy – e.g. by replacing all data read actions by respective UML constructs.
- VIDE is a UML-based language that deals with behaviour that includes processing of database data. This provides some experiences and concepts relevant for the object database standardisation initiative recently initiated by OMG. Particularly, a seamless integration of imperative constructs and powerful query expressions, as well as the postulate of making the prospective standard possibly platform-independent in the sense the MDA approach are noticeable.

Hence the general observations on UML from the point of view of executable modelling and a list of issues encountered in the specification (already outlined in several papers) have been collected into a working document and cooperation with an OMG member planning a response to “Concrete Syntax for a UML Action Language” RFP has been initiated. VIDE solutions can also indirectly inform the OMG Object Database Technology Working group activities.

9.4 Integration of methods and concepts into partners' tools

The research and implementation results of the project have been directly or indirectly contributing to VIDE industrial partners' tools development:

- Softeam has ported its Objecteering UML tool onto the Eclipse platform to ease its integration. The experience gained during the development of the VIDE Java code generator will be used in the development of model compilers for Objecteering.
- TNM:WebFace has been integrated with VIDE in terms of the Web service connectivity. This solution, incorporating Java code generator can be used to extend the WebFace automated code construction from GUI towards the application logic.
- Rodan has extended its workflow system OfficeObjects(R)WorkFlow with the XPDL2.0 import functionality to be able to consume process models developed with other tools. This includes pre-CIM and CIM solutions provided by VIDE as the mean of extending the tool's business modelling capability.
- Altec has made its tool interoperable through Web services with VIDE executable models. This paves the way for the ONAR tool to incorporate platform-independent behaviour specification into the legacy data integration programming.
- SAP achieved interoperability of VIDE with NetWeaver in terms of Web services and generated java code. The project results in terms of visual programming, aspect oriented modelling, defect detection and code generation are likely to inform future development of that tool.

9.5 Enablers

In the course of dissemination and exploitation work several means of enabling learning the VIDE concepts and prototype have been developed:

- VIDE tool prototype – freeware core components covering the path from CIM to executable code are available on the VIDE website.
- Selection of Open Source for many VIDE tool core components available on the VIDE website.
- VIDE Cookbook document and the VIDE e-learning system developed from it – available in the online form and as Eclipse help system plug-in.
- Web library of VIDE samples illustrating the industrial use cases developed for VIDE.

9.6 Areas for further development

The main directions of future research and development include:

- Model visualisation at CIM and PIM level with the focus on expressiveness and approachability for different kinds of users.
- Code generation from platform independent models – optimisation and handling persistency.
- Design of high-level query / programming languages and environments for database applications.
- Model simulation and model-based testing.
- Model validation.

9.7 Measurable results

Domain	Planned value	Achieved value	How it was verified
Visual coding on the PIM level	With VIDE, at least 90% of the code of applications covered by Executable UML will be coded visually on the PIM level.	Complete evaluation model behaviour implemented using VIDE Visual Editor solely (with textual OCL inclusions though)	Evaluation workshop involving 3 teams of undergraduate and graduate students, involving sample application implementation assignment performed using Visual VIDE language.
Application development time	Applications covered by Executable UML will be developed with VIDE in 1/2 of the time needed now using both Executable UML tools or more traditional Rapid Application Development (RAD) tools.	1) Up to 50% gain in development time when performing a complex assignment in VIDE compared to traditional programming in Java and SQL. 2) 30%-50% gain in development time using Visual VIDE language when performing simple programming tasks compared a traditional textual language.	1) Evaluation workshop involving a group of 14 undergraduate students and a group of 10 graduate students, involving sample application implementation assignment performed by particular teams in Java, Textual VIDE language and Visual VIDE language. 2) Evaluation workshop involving 16 participants performing 3 simple programming tasks in VIDE Visual language and in SmallTalk.
AOP composition at PIM-level	Expressiveness in terms of joinpoint models and possibilities of structural and behavioural adaptation compared to existing AOP approaches. The (visual) aspect composition at PIM-level will be sufficient to cover at least 90% of typical AOP-scenarios in those parts of an	The approach, developed in the VIDE project provides features, which are analogous to the features provided by the AspectJ programming language.	Feature comparison analysis against the selected known AOP approaches.

	application, that can be defined in the VIDE language.		
Quality of Platform Independent Models	With the discovery of quality defects and following model refactorings the qualitative and quantitative quality as measured against a specific quality model (probably based on ISO 9126/25000 but for MDA) will increase by at least 20%.	80% more defects are found using VIDE-DD than manual inspections.	Workshop - comparison of the quality of PIM developed with and without the quality defects diagnosis approach realized within VIDE-Defect Detector.
Skill level needed to use the programming tool	The required skill level is the basic knowledge of a UML-tool. Individuals who have the knowledge from a 1 week preliminary UML course will need 1 week of training to be able to create small Executable UML applications.	> 60% success rate throughout all student groups and assignment elements when programming in visual and textual VIDE language. The assignment was time constrained to 150 minutes and was presented to students having UML knowledge but only a few hours training in VIDE.	Evaluation workshop involving a group of 14 undergraduate students and a group of 10 graduate students, involving sample application implementation assignment performed by particular teams in Java, Textual VIDE language and Visual VIDE language.

9.8 Exploitable assets

This chapter will outline the various assets that have accrued to the different partners as a result of their participation in the VIDE project. Whilst the VIDE Assets discussed in section 9.8.2 are easily identifiable and quantifiable, along with the asset combinations discussed in Section 9.8.3 the VIDE knowledge is the type of asset which is both difficult to quantify and much harder to identify.

9.8.1 VIDE Knowledge

Whilst the concrete assets derived from the prototype VIDE toolset are described elsewhere, less obvious, but nonetheless valuable knowledge for future developments have emerged from partners' research and development efforts. For example, partners have already identified the following useful outcomes:

- User requirements definition, including usability requirements, user definition and tool evaluation
- Non-technical business user expertise to support VIDE design and accessibility.
- The concept of the VIDE PIM language: UML2 compliant language for business applications specification incorporating imperative statements and query language capability.
- Design of the VIDE textual syntax
- Visual notation of the VIDE language
- Concept and metamodel of the VIDE CIM level language

- Notation of VIDE CIM level language
- VIDE software development process
- CIM-to-PIM transformation
- Design of the editor for VIDE visual language
- Specification and implementation of VIDE language mapping onto underlying standard metamodels
- VIDE textual editor and model execution engine
- The concept, design and specification of the VIDE GUI environment.
- The visual and conceptual metaphors that support VIDE project artefact visualisation, modelling and transformation
- Evaluation of software tools and environments.
- Analysis and evaluation of graphical modelling notations
- VIDE GUI evaluation method
- Aspect-oriented modelling and composition techniques, and specification of aspect oriented composition to be supported by VIDE
- Quality defect discovery methods
- Visual action prototyping approach
- Architecture for VIDE tool integration
- Legacy application modernisation using the VIDE language

9.8.2 VIDE Assets

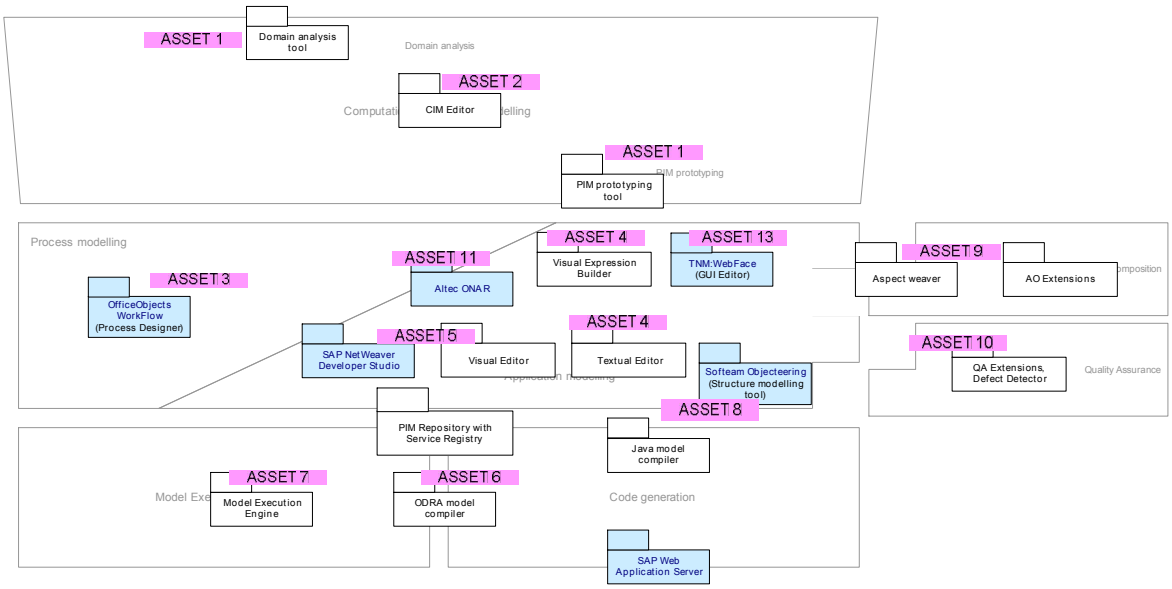


Figure 4: VIDE Assets by functionality

9.8.2.1 CIM Tool and Process – ASSET 1

Partners involved: BU

Open Source: No

Independently exploitable: Yes

The CIM tool supports the 'upstream' development processes, from raw domain information, through informal and formal modelling, to PIM prototyping. Tools support includes:

- 'pre-CIM' analysis with tools to organise and elicit domain information and develop informal analysis models using a bespoke, flexible and accessible, visual notation.
- Business process (CIM) models using formal notations (Role Activity Diagrams (RADs), Entity Relationship (ER) Models, etc.).
- CIM-PIM transformation wizards.
- CIM repository / knowledge base.
- PIM prototyping tool, supporting UML class diagrams and activity diagrams.
- Integrated specification tools, for defining interfaces (at the system boundaries) and interaction behaviours.
- Support for forwards-reverse traceability of model elements at all levels.
- Rich graphical modelling environment to support efficient and effective development processes.

Whilst the pre-CIM notation is flexible and informal, and the PIM notation is based on standard UML, there are many options for CIM models. These vary in terms of expressiveness and, critically, accessibility to non-specialist users. There is a strong case, therefore, to support different CIM notations. Hence, the CIM tool architecture will support plug-in CIM modelling components (see VIDE ASSET 2 for example).

The CIM tool offers a number of benefits over other approaches, such as: introduction of a 'pre-CIM' stage to the MDA; flexible bespoke notation for building initial analysis models; wizards for traceable transformations; and integration of wizards for system specification. These benefits result in potential for various exploitation activities from research and development to potential productisation. It will provide, as a default, CIM models based on RADs and ER models, and could, therefore, be exploited as a stand-alone tool or combined with VIDE ASSET 2 to support the VIDE CIM Level Language.

9.8.2.2 VIDE CIM Level Language (VCLL) Tool – ASSET 2

Partners involved: IWI

Open Source: No

Independently exploitable: Yes

The VIDE CIM Level Language (VCLL) provides an enhanced business process modelling notation based on BPMN. The exploitable assets in detail are:

- The metamodel and OCL based definition of the VCLL
- The Modelling tool for the VCLL which will consist of four sub-components:
    - the modelling component for organizational structures of enterprises
    - the modelling for data objects
    - the graphical editor for Business Rules and
    - the modelling component for business processes, which will allow to refer to modelling elements created by the three components above in business processes.

The VCLL tool as a whole is realised as a self-contained graphical modelling tool, with provision for integration with other tools. Namely workflow management systems (see VIDE ASSET 3) can use VCLL models as input via an XPDL 2.0 export facility. Other VIDE components can benefit from this tool as they can access the VCLL models through a shared EMF repository. Furthermore the VCLL tool can use input created by the pre-CIM Tool (see VIDE ASSET 1)

The VCLL tool may be exploited as a stand-alone component, but also integrated with the CIM Tool (see VIDE ASSET 1), or other parts of the VIDE toolset.

9.8.2.3 Workflow Management – ASSET 3

Partners involved: Rodan

Open Source: No

Independently exploitable: No

This asset comprises the integration of Rodan's existing OfficeObjects®WorkFlow system with the VIDE toolset, in order to support the development of application software to support business processes. This may be integrated with either/both the VCLL tool (see VIDE ASSET 2) or the PIM repository and suitable PIM development components to provide various potential product offerings.

9.8.2.4 VIDE Core – ASSET 4

Partners involved: PJIIT

Open Source: Yes

Independently exploitable: Yes

This asset comprises various components that form a core of VIDE functionality, focussed on PIM level modelling:

- The Visual Expression Builder.
- Customization and integration of UML and OCL metamodels in terms of relevant constructs for VIDE.
- The abstract syntax tree (AST) of the VIDE language.
- Concrete textual syntax of VIDE and textual VIDE editor.
- Interface to EMF-based implementation of UML and OCL models repository (using the Eclipse MDT project) and Repository Browser component providing entry point to VIDE PIM-level functionality.
- Mappings between textual VIDE language and the metamodel.
- UML profile for consumed and published Web Services and a wizard for importing consumed service WSDL definitions.

The Visual Expression Builder (VEB) component is a visual editor implemented using Eclipse GMF framework, realizing an object variant of the Query By Example (QBE) approach, designed in the course of VIDE project to comply with UML syntax and object model, named Object QBE (OQBE). The visual expressions can be specified with the purpose of:

- embedding them (through a hyperlink) into other expressions or imperative statements inside the textual VIDE editor,

- embedding them into selected constructs of visual VIDE editor, or – specifying ad-hoc queries against an executable PIM instance.

The component comprises the following elements:

- Concrete and abstract syntax of the OQBE
- Visual Expression Builder – Eclipse-based OQBE editor
- Contextual hints inside the OQBE editor, realised based on the PIM repository access
- Design and implementation of the OQBE to OCL mapping

The VEB can be used in combination with VIDE Visual Editor (VIDE ASSET 5) and / or Model Execution Engine (VIDE ASSET 7).
The asset constitutes an implementation of UML-compliant action language and provides an entry point for invocation of PIM-level VIDE functionality including model execution. If installed standalone, it allows to edit and store textually specified behaviour into a form of UML metamodel instance, as well as to import Web service descriptions into the model or specify them manually using the provided profile.

9.8.2.5 VIDE PIM-level Visual Editor– ASSET 5

Partners involved: SAP

Open Source: No

Independently exploitable: Yes

Visualisation of models is a common task in the development of business applications, since graphical information is more easily accessible to stakeholders with business backgrounds such as ISVs and customers. Current models only partially contain process or business logic. Therefore, extending static business process and structural models such as component models and class models is a logical step to increase the coverage of modelling and allow modelling complete applications.

The PIM level visual editor is intended for software designers to model the behaviour of their applications visually. The visual notation offered by this component is unique compared to the action notations offered by most state of the art MDA tools. Further, it provides several useful features that simply the creation and manipulation of action models such auto-layouting and content-assists.

9.8.2.6 ODRA Model Compiler – ASSET 6

Partners involved: PJIIT

Open Source: Yes

Independently exploitable: Limited

The ODRA model compiler takes models from a VIDE PIM (EMF-based) repository and compiles them for the ODRA platform. One of the goals of the VIDE project is to map the VIDE language to a target platform that provides a test bed with strong data querying capabilities and offers a semantically clean and powerful approach both to query and programming languages. The target platform is ODRA (Object Database for Rapid Application development) platform from PJIIT, which is based upon the unique Stack-Based Approach (SBA) to database and non-database query language semantics.

This may be combined with the VIDE core (and other components where appropriate) to provide a route from PIM to executable code.

Developers of ODRA applications may hence use this asset to replace textual coding with completely VIDE-based specification of executable (compilable) ODRA program.

9.8.2.7 ODRA Execution Engine – ASSET 7

Partners involved: PJIIT

Open Source: Yes

Independently exploitable: No

This asset allows the execution of VIDE models retrieved from VIDE PIM, using the ODRA model compiler (see VIDE ASSET 6) and the underlying ODRA based execution engine. The goal of that functionality is to take advantage of the precise behaviour specification available at VIDE PIM and to provide a possibly straightforward way of test-running the model using sample data. The functionality is based on the model compilation and execution; however those steps are made as transparent as possible. This may be combined with other components of the VIDE toolset that support the development of VIDE PIM-level models.

9.8.2.8 Java Model Compiler (Code Generation) – ASSET 8

Partners involved: Softeam, SAP

Open Source: Partially

Independently exploitable: Limited

Code generation for a behavioural model from a UML activity model is not an easy task because activity models are used for different purposes. Using the VIDE language adds both the missing precision of an activity model and reinforces the generation capabilities by including OCL expressions. The Java model compiler may be combined with other components of the VIDE toolset to provide a route from PIM to executable Java code.

The Java J2EE code generator consists of two parts: The first part is a "generic" Java/J2EE code generator which will be released as Open Source. The second part uses the "generic" Java/J2EE code generator for generating Java code, but in addition generates configuration file and archives for deploying the J2EE to a SAP WebAS J2EE server as an example for a J2EE server. This part will not be released as OpenSource but exploited separately.

9.8.2.9 PIM Level AOP Support – ASSET 9

Partners involved: Fraunhofer (FIRST)

Open Source: Yes

Independently exploitable: Limited

This asset comprises an approach for aspect oriented modelling (AOM) and aspect oriented composition (AOC) at the PIM level. The approach differs from other AOM/AOC approaches and was inspired by the concepts used in the domain of aspect oriented programming. The approach can be exploited independently, for example, for consulting and education.

The realisation of the developed approach consists of the following software artefacts:

- UML Profiles, which allow aspect oriented constructs to be represented in the VIDE repository
- Transformations for pointcut matching and aspect weaving

The implemented software artefacts may be combined with other components of the VIDE toolset, particularly requiring the VIDE core or other PIM modelling support or other comparable components in different toolsets and MDA frameworks.

9.8.2.10 Quality Defect Detection and Visualisation – ASSET 10

Partners involved: Fraunhofer (IESE)

Open Source: Yes

Independently exploitable: Limited

This asset comprises exploitable knowledge concerning quality defects that exist in model driven software development, especially at the PIM level. This knowledge includes information on the causes, effects, and available treatments, as well as the specific defects. This has been used to develop a plugin based platform for quality defect diagnosis on the PIM level. This can be exploited as a component part of the VIDE toolset, when combined with the VIDE core or some other PIM modelling tool.

9.8.2.11 ONAR Integration – ASSET 11

Partners involved: Altec

Open Source: No

Independently exploitable: No

This asset consists of the integration of Altec's ONAR (Ontologies based Enterprise Application Integration) with the VIDE toolset. ONAR could add value in situations were reusability in SOA applications are needed and, therefore, the merging of Semantic Web and MDA approaches would be a requirement.

9.8.2.12 Wrappers for Relational and XML data – ASSET 12

Partners involved: PJIIT

Open Source: Yes

Independently exploitable: No

This asset comprises wrappers for relation and XML data in order to bring the features of platform independent visual programming into existing data management tools. It is exploitable where combined with components of the VIDE core.

9.8.2.13 WebFace Integration – ASSET 13

Partners involved: TNM

Open Source: No

Independently exploitable: No

This asset comprises the integration of TNM's WebFace platform for GUI development with VIDE applications. The platform allows to create user interfaces for VIDE applications. The communication between the GUI and the business logic is based on the web-service standard. Since the GUI-builder of TNM:WebFace is inetegrated in the Eclipse environment of VIDE, it's possible to develop a complex business application with a graphical user interface using only the VIDE environment.

A GUI created with TNM.WebFace has many advantages over other solutions:

- The application can be published as a Java Applet or a desktop-application. Therefore, the application can be used on all operating systems where a Java runtime environment is available (for example Windows , Linux, Solaris, Mac OS X). The application can be installed on the client computer or the application can be delivered with a web-browser.
- In order to create the GUI, there is no need to write a single line of Java code. The logical connections between the components are created by drag&drop. In contrast to other Java GUI builders, TNM:WebFace doesn't generate new Java code on the fly in order to generate connections between the graphical components. Hence the resulting jar-library with the program

data doesn't grow with the complexity of the GUI. Only if new components are added to the GUI, the corresponding classes are also added to the jar-file. Adding new events to the GUI only adds more data.

- Because the logical representation of the events connecting the GUI components is available, it's possible to implement complex debugging and help systems. For example, in a research project (Smartkom, www.smartkom.org), an animated presentation agent was used to guide the user through an application with a GUI based on TNM:WebFace.

9.8.2.14 E-Learning System – ASSET 14

Partners involved: TNM

Open Source: No

Independently exploitable: No

The VIDE e-learning system provides the documentation for the VIDE system. The course shows how to develop applications using the VIDE technology. The document generator of the e-leaerning system is used to generate also a help system which can be integrated in the Eclipse help system. The course is exploitable where the integrated VIDE toolset is offered.

9.8.3 VIDE asset combinations

VIDE has a number of asset combinations which may be exploited by interested parties. It is thus possible, for example, to create a Business Modelling Toolset by combining Assets 1 and 2. This will involve bi-lateral agreement with the two parties involved (BU and iWi). There are also possible combinations which will include BU/iWi and RODAN or ALTEC for example. Again this will involve similar bilateral agreements.

9.8.3.1 Business Modelling Toolset – ASSET 15

Consisting of the Domain Analysis Tool and the CIM Editor
Partners involved: BU and iWi

Open Source: No

Independently exploitable: No

This Asset combination provides a comprehensive toolset for the business domain modeller. They may be a domain user or expert or a business analyst communicating with the domain user. It provides document storage and entry level notations prior to a detailed business process model view including business rules.

9.8.3.2 Business to Code Toolset – ASSET 16

Consisting of the Domain Analysis Tool, CIM Editor, PIM Prototyping Tool, VIDE Core and Java Code Generator for vertically integrated model driven development
Partners involved: BU, iWi, PJIIT,SOFTEAM

Open Source: Only PIM level components

Independently exploitable: No

This asset combination extends the Business Modelling Toolset with PIM-level VIDE language features and code generation, to provide a top-down realisation of the MDA modelling stack. Combined use of those modules allows analysts and programmers to seamlessly follow the elaboration of domain models towards executable platform-independent application models and eventually – to produce target platform code from them. This exemplary combination of VIDE tools constitutes a frame that can be further extended e.g. with aspect modelling, defect detection, GUI development, workflow development etc.

9.8.3.3 PIM level Application Development – ASSET-17

Consisting of VIDE Core, Model Execution Engine, WebFace and Java Code Generator for PIM-level application prototyping and development
Partners involved: PJIIT, TNM, SOFTEAM

Open Source: No

Independently exploitable: No
This asset combination assures a UML-based platform-independent specification of data-intensive applications, using a single, uniform, object-oriented query and programming language. The toolset can provide a tangible output from the modelling activity by generating complete, executable Java code.

9.9 Further exploitation of the software

The VIDE consortium is made up of five commercial partners. Of these SAP and ALTEC are considered industrial partners, whilst SOFTEAM, RODAN and TNM are developers and suppliers of software development tools. The VIDE project has provided extensions to the toolsets of all the commercial partners and their exploitation plans reflect this. Whilst SAP is a major European company the other partners are all SME's and therefore have different ways of exploitation.

SAP is the recognized leader in providing collaborative business solutions for all types of industries and for every major market. Serving more than 41.200 customers in more than 120 countries, SAP is the world's largest business software company and the world's third-largest independent software provider overall. SAP has a rich history of innovation and growth that has made it a true industry leader so that 75% of the Fortune Global 1,000 companies run SAP software. SAP Research is the research department of SAP, significantly contributing to SAP's product portfolio by identifying breakthrough innovation and conducting co-innovation with external partners and customers. SAP Research acts as SAP's IT trend scout, identifying emerging IT trends, researching and developing in strategically important SAP business areas, and leveraging entrepreneurial inventive talent. SAP is exploring a number of VIDE related exploitation items or activities, within their research process. These involve the modelling infrastructure, behavioural aspects in the design and maintenance of business processes and business components, aspect Oriented Modelling and Model Visualisation techniques.

SOFTEAM has fifteen years experience in object oriented methodology, and has been editing and distributing an object oriented CASE tool for 14 years. SOFTEAM has three main activities:

- CASE tool editing with the "Objecteering" CASE tool, which has been on the market since 1992.
- Consulting; as a provider of object oriented methodology, and as an active OMG contributor
- Training; SOFTEAM provides training in related technical areas (languages, techniques, tools) and in methodological areas.

The Objecteering/UML CASE tool supports software modelling and development automation across the entire software development lifecycle. It is differentiated from competitors by its strong code generation capacities, by its ability to drive entirely application development and by its unique customisation capacities. Objecteering was one of the first tools to provide MDA capacities, being precursor to MDA since 1996 (through a technology called "hypergenericity"), and being the first and most complete implementation of UML Profiles with the "Objecteering/UML Profile builder tool" launched in 2000. The VIDE project has given Softeam the ability to reinforce their strong code generation capacity with behavioural generation to keep their advance on free tools. In addition they have improved their integration with Eclipse IDE thus lowering the roundtrip effort from the model to the compiler. This will allow them to offer their VIDE enhanced Java generator without team capabilities at a special price to allow them to gain in markets against cheap competition, and with team capabilities to ensure they retain market share with existing customers.

Rodan Systems S.A. is one of the major producers of Web based content, knowledge and workflow management systems. RODAN specializes in the development and implementation of application software based on its proprietary OfficeObjects® products and solutions. The systems offered by the company meet current needs of information management, organise multimedia information repositories, allow integration of traditional legacy systems with advanced solutions and directly support realisation as

well as monitoring of business processes. Within the project plan RODAN is responsible for providing its OfficeObjects®WorkFlow product (with VIDE extensions) to support the VIDE process view at PIM and PSM levels. RODAN has an exploitation strategy which features further investigation of the pre-CIM and CIM level tools provided by BU and IWi which provide a natural business domain entry level to their tooling.

ALTEC Information and Communication Systems S.A. is one of the largest ICT companies in Greece, with activities in software, system integration, services and products and telecommunications, in several branches all over Greece, while also operating subsidiary companies in Romania and Bulgaria. ALTEC develops and provides technologically advanced systems and software applications to cover the entire spectrum of computerised and organisational demands in both the public and the private sector. ALTEC has considerable expertise in the domain of business and ERP software in Greece; it is the vendor of ERP ATLANTIS and XLINE. Exploiting this experience and blended with the Semantic Web and Service Oriented Architectures paradigms, ALTEC produced the ONAR tool, which provides a solution in Semantic Web domain competition. The tool has been selected in various European research programmes as well in commercial applications. ALTEC exploits ONAR both in its existing ERP customer base as well as in the SOA market. ALTEC has an exploitation strategy which features further investigation of the pre-CIM and CIM level tools provided by BU and iWi which provide a natural business domain entry level to their tooling.

TNM Software develops applications for Intranet, Internet and mobile devices. TNM is specialised in the development of highly interactive and portable web-applications based on the Java programming language. On the basis of TNM's own development platform, TNM:WebFace, TNM develops a broad range of applications for many areas. One example is the e-learning system E-MIL (Electronic Multimedia Intelligent Learning, available from http://www.emil-learning.de), which provides an environment for teaching and learning via the Internet. Another application is a company wide information system for an advertising company. The system handles all types of business processes from contract and invoice management, manufacturing to maintenance. VIDE will provide a natural extension to TNM's exploitation activities and they will introduce VIDE at selected customers, whilst continuing to disseminate information on the product. TNM will continue to host the VIDE web site.

Although the industrialization of the tool is mostly assigned to the abovementioned partners the remaining members of Consortium have also undertaken some steps to advance the production of the tool that may finally be adopted by the commercial users. PJIIT have decided to release a beta version of their software in the next few years that will be useful for SME companies in Poland. BU , IWi and Fraunhofer are also open to the potential development of their tools to make them commercially valuable.

10 Conclusions

The originally planned scope of the project has been increased significantly in the course of investigation of the state of the art and requirements analysis; to eventually involve all major layers of the MDA. The analysis of the project results shows that the potential for improving the productivity lies not only in visualisation but also in assuring a more seamless, uniform and traceable integration of models spanning from informal domain models down to executable code. Taking into account the precise semantics of the PIM language, model execution and code generation can be assured and this way a significant part of complexity and irregularity of the platform-specific solution can be mitigated.

Evaluation of the VIDE software against the tools available on the market revealed some advantages already visible despite the prototype nature of it. A number of novel features that seem to contribute to that gain have been identified and will be further developed through several planned projects. Thus we can conclude that the project contributed to the European Research Area, has commercial value and can be considered as a success.

11 Current status and further research planned

The toolset is currently available for trial on the VIDE website at http://www.vide-ist.eu/. In addition there is a comprehensive help section and a number of pre-created models to give a new user useful starting points wherever they may start in the MDA stack. In addition there is a helpful e-learning system that will enable a novice to get started. Research and education based partners are planning several activities that tackle at the VIDE Project work. The Consortium members are planning to further knowledge arising from VIDE Project by researching following areas:

SAP

"SAP Research plans to participate in FP7 Call 5 projects. The main assets from the VIDE project expected to be reused are experiences in the model visualisation with focus on target users. Our aim is focussing on runtime and executable models that support a) the understanding of the system behaviour b) verification of correctness. The areas that will be further researched are:

Aspect Oriented Modelling .

The potential benefit of AOP addresses internal as well as external software development. Applied to internal software development and modelling projects AOP may increase development efficiency and software quality (i.e. increased modularisation). Using AOP concepts may also ease the development of software variants built by SAP as well as extensions by ISV partner - that for example build industry solutions upon SAP platforms - and customers during the customizing process (external software development).

Model Visualisation Techniques

Visualisation of models is a common task in the development of business applications, since graphical information is more easily accessible to stakeholders with business background such as ISVs and customers. Graphical frameworks must be accessed and, as in VIDE, several either textual or graphical modelling sources are used to generate a visualisation of a model. Providing an automatic visualisation is a non-trivial task where VIDE results can be exploited for modelling also other kinds of models.

SAP Research plans to participate in FP7 Call 5 projects. The main assets from the VIDE project expected to be reused are experiences in the model visualisation with focus on target users. Our aim is focussing on runtime and executable models that support a) the understanding of the system behaviour b) verification of correctness. However there are further research projects derived from VIDE planned internally"

PJIIT

"The PJIIT development plan assumes participating in at least 5 large research projects in medical area until 2012 and applying for partnership at NoE project. At the moment, the project "Advanced analysis and interactive synthesis of human body motion system and tooling modules" proposal has been created and submitted. The project will involve design of a dedicated data management solution. The experience from VIDE project was one of the chief assets to allow PJIIT to play the role of the coordinator in this project. Two next project in the medical area are being developed and will be ready to submit the mid next year.

In the area of purely technical topics dealing with database technology and object oriented programming environments, PJIIT plans to submit a proposal for project on development of a database solution and programming environment to be developed under open-source license and targeted to SME sector. More projects may be developed based on the industry needs.

The target market for the research projects conducted in the field of medicine are mostly hospitals and polish authorities such as Ministry of Health, as well as medical schools. The medical market is developing rapidly with the private hospitals and medical centres entering the market, however the costs of software is often too high for the underinvested medical industry. Furthermore it is also not adjusted to the hardware the polish hospital posses. Therefore several of hospitals already expressed interest in conducting the research and develop the solutions dedicated for them. The projects of purely database technology can be targeted towards any industry. However the target market for PJIIT are SME companies, that are more flexible in adopting new concepts. The exemption from this rule are the telecom companies."

BU

"Experience from projects such as VIDE is extensively used to underpin new research and development projects, and is one of the most significant, if hard to quantify, impacts of the project on our business. BU

expects to be involved in a number of research projects in the next five years. These are likely to be both UK and EU funded projects. Currently a UK project has been submitted in the area of Domain Specific Languages which has resulted from BU's involvement with the VIDE project. A further two projects are currently under development in the Requirements Engineering and Software Product Lines areas. Further projects have been submitted in partnership with hospital and police services in the area of Business Process Modelling, which further enhances the value of VIDE.

Participation in the VIDE project, and outputs derived from it, are essential for a relatively small research centre to compete for future research and development projects. Again, the impact is hard to quantify, but this has arguably already had an influence on internal support for the centre.

BU is interested in developing practical tools for software development, although it is recognised that productisation is expensive and time-consuming, and within the timescale of the VIDE project, only a limited functionality prototype will be possible. Hence, it would be challenging for BU to develop product offerings based on VIDE assets. However, the potential for spin-off activities was considered during the evaluation of the project. The prototype will be used within the curriculum: as a teaching tool for students to understand the 'upstream' development processes that are not well supported by other CASE tools; in or for student projects; to support empirical studies of analysis and process modelling tasks; and as a focus for further research and development. It has also spawned a new project looking at the transformation of business process models to models further downstream"

FIRST

"During the VIDE project, Fraunhofer FIRST has gained a lot of knowledge about the model-driven software development in the domain of data-intense business applications. Concepts used in the area of aspect-oriented programming were evaluated and adopted in the model-driven software development process.

Since Fraunhofer FIRST has strong background in the domain of embedded systems (software development, quality assurance e.g. testing and deductive verification), the knowledge and the experience gained in the VIDE project will be transferred into various areas such as model-driven software development for embedded systems and model-based testing. Since the Fraunhofer-Gesellschaft has a clearly defined mission of application-oriented research with a focus on key technologies of relevance to the future, we will focus on research activities.

The research priority of Fraunhofer FIRST is the development of new methods for software engineering and quality assurance, especial in the domain of embedded systems. Many activities and projects in this area are processed and also planned for the near future. These activities and ongoing projects provide a platform for introducing the aspect-oriented techniques into a concrete software engineering area, like testing, model-based testing and product line engineering. Due to the excellent position of Fraunhofer FIRST, the area of aspect-oriented concepts within model-driven software engineering will be researched and exploited in various existing and future projects (ITEA, EU, BMBF, industry).

Fraunhofer FIRST participates and will participate in important projects concerning the software engineering for embedded systems in the automotive, railway and aerospace domain. The following topics are planned to be addressed with respect to the aspect-oriented concepts at programming language level or at model level. "

IESE

"For a Fraunhofer Institute, VIDE is an interesting project, which has already resulted in enhancement of our knowledge base on software diagnosis and MDSD that will be transferred into the industry. Furthermore, conducting the research and developing the tools has paved the way for new research that will result in consequent research and industry projects to further mature the MDSD process. The concepts developed in VIDE, will, after a thorough evaluation, be partly integrated into existing services."

Iwi

The use of the VIDE toolset is to improve the efficiency or productivity of other development projects. The software prototype implemented by DFKI has the potential to be used in many other different activities and research projects. In particular, as the VCLL tool consists of four sub-components these can be independently adapted for reuse. For example, the Business Rule design component is a potential candidate to be reused in projects related to Business Intelligence. The added value of VIDE Project for the Institute for Information Systems is:

- *Knowledge regarding the computation independent modelling using process, data, organizational and business rule modelling methods*
- *Transformation of unstructured requirements information*
- *Use of business requirements directly in the process of software creation*
- *Excellent knowledge regarding the modelling, meta modelling and model transformation technologies in the Eclipse context (EMF, GMF, ATL), which is immediately applied to other projects.*
- *Use of CIM models controlling workflow systems.*
- *New starting points for further research*
- *Better understanding of the industry needs and trends in the software engineering field*

The Institute for Information System at the German Research Center for Artificial Intelligence participates and will join new important projects concerning the integration of business models into the software engineering process. A Vide follow up project, the Institute participates in, is the SHAPE project.

12 Further reading

For finding more info about practical usage of the VIDE software we would recommend to read the Cookbook (http://www.vide-ist.eu/reflib/cookbook.html). Elaborated version of that material, extended with screencast demonstrations, can be found in the online e-learning course at: http://www.vide-ist.eu/reflib/elearning/vide/index.html .

For those that are interested in quality assurance interesting info can be found at: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-376/paper3.pdf

An overview of the visualisation in platform-independent modelling is available the paper: http://www.vide-ist.eu/download/VIDE_Visual_Programming.pdf

A number of other papers and whitepapers covering topics of CIM and PIM-level model development are available from the VIDE “Publications” webpage at: http://www.vide-ist.eu/publications.html

A more industrially-oriented view of the VIDE solutions is provided through the descriptions of “VIDE Industrial Use Cases” available at http://www.vide-ist.eu/reflib/usecases.html

The VIDE downloads page http://www.vide-ist.eu/reflib/dist.html contains, the publically available VIDE prototype components together with contact information for each of them, which can be used to get more details on particular topics.

Please visit our website at www.vide-ist.eu to find more information on the various project activities and results.

13 Appendix (multimedia presentation)