American Journal of Science, Engineering and Technology
2018; 3(2): 29-33
http://www.sciencepublishinggroup.com/j/ajset
doi: 10.11648/j.ajset.20180302.11
ISSN: 2578-8345 (Print); ISSN: 2578-8353 (Online)

Model Driven Engineering (MDE) Tools: A Survey

Moujtahid Soukaina, Belangour Abdessamad, Marzak Abdelaziz

Faculty of Science Ben M'Sik, University Hassan II, Casablanca, Morocco

Email address:
soukainamoujtahid@gmail.com (M. Soukaina)

To cite this article:
Moujtahid Soukaina, Belangour Abdessamad, Marzak Abdelaziz. Model Driven Engineering (MDE) Tools: A Survey. *American Journal of Science, Engineering and Technology*. Vol. 3, No. 2, 2018, pp. 29-33. doi: 10.11648/j.ajset.20180302.11

Received: July 19, 2018; **Accepted**: August 20, 2018; **Published**: September 13, 2018

Abstract: Model Driven Engineering (MDE) is a new discipline in software engineering that advocates the massive use of models throughout the software development process. The emergence of this discipline has been accompanied by the prosperity of the tools that support it. On the long run, switching to MDE can be beneficial in case the process works, and the process itself depends on the tools. Since the transition is expensive, it is important to invest wisely, and choose the right tool. However, only recently tool creators have started considering metamodeling as an important issue in their list of concerns and university prototypes are sometimes difficult to download and test, so these tools remain little known overall and need to be listed. The aim of this article is to determine the strengths and weaknesses of the support that each of these MDE tools offer to the developer's tasks, in order to learn to identify the right tool that meets the specific needs of the software engineer, without recommending any particular tool or vendors: It will present a significant number of the most popular MDE tools, in order to keep this paper simple, list some criteria for comparing these tools and evaluate them against those criteria.

Keywords: MDE, MDA, Model, Transformation, Tool

1. Introduction

The pressure to reduce the time-to-market and the ever growing design difficulties require new research efforts to adopt languages with high abstraction level or/and new approaches to cope with that. Model Driven Engineering (MDE) is the current betting to raise the design abstraction level and to provide mechanisms to improve the portability, interoperability, maintainability, and reusability of models [1]. In addition, MDE helps to abstract platform complexity and to represent different concerns of the system [2]. Model Driven Engineering (MDE) is a new discipline in software development that has emerged following the emergence of the Model Driven Architecture (MDA) initiative [7]. If the MDA's goal is limited to modeling a system in the form of a PIM and then transforming it into a PSM [14], the MDE goes beyond that goal and generalizes the use and transformation of models throughout the software development cycle [13]. From the developer's point of view [12], a key issue for acceptance of any approach is good tool support so that software programs can be created in an easy and efficient manner. However, only recently have tool creators started considering metamodeling as an important issue in their list of concerns [4]. It is still rare to find a development tool that has explicit support for meta-model creation and/or configuration, which can be surprising if that meta-modeling, is considered one of the founding principles of MDE [10]. This means that, until recently, a developer who wanted to use a certain meta-model would probably have to either create a new modeling tool, which is not reasonable at all or settle on a CASE tool (with a hard-coded meta-model) that allows to perform the desired task with the least possible hassle [2]. However, adding metamodeling support to a tool does bring some practical issues that should be mentioned, such as separating the object-oriented programming languages class-instance relation from the meta-model-model relation [11], deciding whether the number of logical levels should be limited or potentially unbounded, and deciding whether the tool should support model transformation and/or code generation [3]. The final purpose of this article is to determine the strengths and weaknesses of the support that each of these MDE tools offer to the developer's tasks without recommending any particular tool or vendors. This article is divided as follows. Section two describes some

MDE tools. Section three presents some criteria for comparing these tools and gives the results of their evaluation against these criteria. The last section summarizes this paper.

2. Tools Supporting Model Driven Engineering

Tools that proclaimed MDE are distributed among powerful commercial tools and university prototypes sometimes difficult to download and test. The following list shows some of the most popular among these tools [5]:

Table 1. *Most popular MDE Tools [5, 8, 9, 15].*

Tools	Description
MetaEdit+	a tool for domain-specific modeling and development
QVTo-Eclipse	an Eclipse implementation of Borland Together based on QVTo
Kermeta2	based on a model-oriented language optimized for meta-models and DSLs
Modelio	the succr. of Objecteering is based on UML and BPMN
Umple	a programming language family to enable model-oriented programming
MDWorkbench	Eclipse-based IDE for code generation and model transformations
Melange	the succr. of Kermeta2 supports the semantics of the modeling languages
MagicDraw	a visual UML, SysML, BPMN, and UPDM modeling tool
JAMDA	creates Java code from a model of the business domain
Ente. Arch.	a UML design and business analysis tool
OpenCanarias	a virtual machine implementation of the QVTo mappings
SmartQVT	a partial implementation of the QVTo language
JQVT	based on a compiled QVT engine for Java
Together	a set of Eclipse plugins which partially implements the QVTo language
Merlin	based on EMF Java Emitter Templates (JET) templates and mapping model
MOFScript	the succr. of UMT which implements the OMG MOFM2T specification
ATL	uses textual syntax and parts of the QVT specification to define transf. rules
Fujaba	a story-driven modeling and graph transf. platform
GrGen.NET	a programming productivity tool for graph transformations
Rational	consists of a set of UML modeling tools for software design

3. Evaluation of MDE Tools

3.1. Criteria for Comparing

There have been various studies characterizing and/or comparing development tools oriented models. Most of these are focused toward MDA tools that are without discussion, the most popular. To compare the tools, it is necessary to choose the characteristics that are going to assess each of them. There are several criteria that can be adopted for comparing MDE tools [6] [7]. These criteria differ depending on the type of the interest in these tools. The following features are selected from the related works in recent years:

1) Update time (UP): The tool is updated regularly (a), sometimes (b) or never (c). Information not available (d).
2) Operating System (OS): The tool has been run on Windows (a), Linux/Unix (b), Mac (c), all of the them (d). Information not available (e).
3) Technological Platforms (TP): Vendor specific platform types (a), technology specific platform types (b), both vendor and technology specific platform types (c) or no support (d). Information not available (e).
4) Availability (A): The tool is open source (a), freely available for research in binary form (b), the tool is commercially available (c) or there is a free evaluation license (d). Information not available (e).
5) Execution Environment (EM): The tool is a plug-in for Eclipse (a), integrated/dependent in other IDE (b), no IDE support (c) or the tool has a standalone APP (d).
6) Domain Application (DA): The tool is a general tool (a), can be used for web applications (b), can be used for management information systems (c) or the tool can be used for real-time/embedded applications (d). Others/Information not available (e)
7) Compatibility with Standards (CS): The tool supports XMI standard (a), CWM standard (b), is an implementation of QVTo (c), is an implementation of QVTr (d), is an implementation of QVTc (e), is an implementation of QVT-Like (f), supports OCL expression (g), supports DD specification (h), supports MOFM2T standard (i), supports HUTN standard (j), supports JMI standard (k) or the tool supports CMI standard (l). Information not available (m).
8) Modeling Languages (ML): UML 2.x (a), Before UML 2.x (b), xUML (c), BPMN (d), Programming languages (e), SysML (f), Petri nets (g) or all of the above (h). Information not available (i)
9) Development Dimension (DD): The tool has graphical concrete syntax (a), has textual concrete syntax (b), has both graphical and textual concrete syntax (c), its abstract syntax/meta-modeling language is EMOF (d), its abstract syntax/meta-modeling language is CMOF (e), supports both

EMOF and CMOF meta-modeling languages (f), its abstract syntax/meta-modeling language is Ecore/EMF (g), its abstract syntax/meta-modeling language is KM3 (h), other meta-modeling languages (i) or the tool support semantic of modeling language (j). Information not available (k).

10) Level of Abstraction (LA): The tool supports dynamic models (a), supports static models (b) or the tool supports both static and dynamic models (c). Information not available (d).
11) Execution Mode (EXM): The tool is interpreter-based (a), is compiler-based/code generator (b) or the tool is both interpreter-based and compiler-based (c). Information not available (d).
12) Model Handlers (MH): The tool supports EMF (a), supports MDR (b), supports both EMF and MDR (c) or the tool does not support model handlers (d). Others/Information not available (e).
13) MDA Model-Levels (MML): CIM (a), PIM (b), PSM (c) or all of the above (d). Information not available (e).
14) Type (T): Exogenous transformations (a), endogenous transformations (b) or both exogenous and endogenous transformations (c). Information not available (d).
15) Level (L): Vertical transformations (a), horizontal transformations (b) or both vertical and horizontal transformations (c). Information not available (d).
16) Direction (D): multi-directional transformations (a), bi-directional transformation (b), uni-directional transformation (c) or all of the above (d). Information not available (e).
17) Scope (S): CIM-CIM (computational independent models can be refine) (a), CIM-PIM (computational independent models are transformed into platform independent models) (b), PIM-PIM (abstract or refine models without binding to any platform-specific information) (c), PIM-PSM (a platform-independent model, with enough transformation definitions can be transformed to a platform-specific model) (d), PSM-PIM (use for reverse engineering and are rather dicult to drive) (e), PSM-PSM (abstract or refine platform-specific models during the component realization and deployment) (f), PSM-Code (PSMs are translated into software artifacts) (g), Code-PSM (used for reverse engineering) (h) or all of the above (i). Information not available (j).
18) Cardinality (C): 1-to-1 (there is one source model and one target model) (a), 1-to-N (can produce several target models e.g., model merging) (b), N-to-1 (several source models are combined into a single model) (c), N-to-N (one or more input model(s) is transformed into one or more target model(s)) (d) or all of the above (e). Information not available (f).
19) Maturity of Tool (MT): The tool has been used in academic (a), industry (b) or both academic and industrial world (c). Information not available (d).
20) Maintenance Support (MS): The tool provides complete support (a) or limited-support (b). No support (c). Information not available (d).
21) Concurrent Transformations (CT): The tool provides concurrent transformations (a). No support (b). Information not available (c).
22) Live/active Transformations (LT): The tool provides live/active Transformations (a). No support (b). Information not available (c).
23) Model Comparison (MC): The tool compares homogeneous models (a), compares heterogeneous models (b), results are in visual/model forms (e.g., UML) (c), results are in textual forms (d), all of the above (e) or no support (f). Information not available (g).
24) Interoperability (IN): VCS (to control concurrent development of the source code by multiple developers) (a), the tool provides with automatic import/export mechanisms for meta-models/models developed with other tools (b) or both VCS and import/export mechanisms (c). No support (d). Information not available (e).
25) Automatic Report (AR): The tool generates report/documentation (a). No support (b). Information not available (c).
26) Security (SEC): Obfuscate (to delete sensitive information from a confidential model) (a), read-only/Locked models (b), code blocks (c) or all of the above (d). No support (e). Information not available (f).
27) Editor (EDI): Graphical (a), command-line (b) or both graphical and command-line (c). Information not available (d)
28) Workspace and Proj. Mngmt (WPM): The tool has workspace and project management (a). No support (b). Information not available (c).
29) Teamwork Support (TS): The tool is multi-users (a), multi-projects (b), or both multi-users and multi-projects (c). No explicit teamwork support (d). Information not available (e).
30) Syntax Editor (SYE): The tool has syntax highlighting (a), auto formatting (b), code completion (c), code navigation (d), code folding (e) or all of the above (f). No support (g). Information not available (h).
31) Semantic Editor (SE): The tool has re-factoring (a), error and warning detection (b), quick fixes (c), debugger (d), reference resolution (e), build systems (f), profiler (g) or all of the above (h). No support (i). Information not available (j).
32) Verification (V): Correctness for syntactic and semantic (the correct models of source language result in the correct models of target language) (a), termination (a transformation always stops executing after a finite number of steps and leads to a result) (b), consistency (models are consistent with each other) (c), completeness (A forward transformation is

called complete if each element of the source model can be transformed to an element of the target model, and vice versa) (d), determinism/Uniqueness/Confluence (different executions of the transformation always produce the same result) (e), comprehensibility (the developed model is comprehensible by the user(s)) (f), robustness (the ability to manage invalid models) (g), definedness (the transformation can be applied to every model of the source language) (h) or all of the above (i). No support (j). Information not available (k).

33)Validation (VA): The tool provides a testing environment (a), the tool provides a simulation environment (b) or both simulation and testing environments (c). No support (d). Information not available (e)

34)Input (I): User-defined models (created manually by the user(s)) (a), derived models (created automatically by the program(s)) (b) or both user-defined and derived models (c). Information not available (d).

35)Output (O): In-place (a), out-place (b), textual artifacts (c), source code (d), database artifacts (e), query (f) or all of the above (g). Information not available (h)

36)Editing Tasks (ET): Access transformations (a), add transformations (b), update transformations (c), delete transformations (d) or all of the above (e). Information not available (f).

37)Meta-Programming (MP): A meta-programming tool (a). No support (b). Information not available (c)

38)Reverse Engineering (RE): A reverse engineering tool (a). No support (b). Information not available (c).

39)Round-trip Engineering (RT): A round-trip engineering tool (a). No support (b). Information not available (c)

3.2. The Evaluation Results

This evaluation is limited to popular tools in order to keep the evaluation and this article simple. These tools were chosen also because this set is a good representative of the current status of MDE-supporting tools currently available, they also presented enough differences amongst themselves to justify their inclusion in this evaluation [5]. Although these tools do not reflect everything that is currently available in MDE tools, they address the MDE based approaches defined earlier by providing the features that can often be found in typical tools of their corresponding approach.

Table 2. *Group 1 [5].*

Tool	UP	OS	TP	A	EM	DA	CS	ML	DD	LA	EXM	MH	MML	L	D	S	C	V	VA
MetaEdit+	b	d	e	cd	abd	a	a	h	ai	c	b	e	d	c	c	i	e	c	b
QVTo-Eclipse	a	d	b	a	a	a	cg	i	bg	c	a	a	d	c	c	abcdef	e	j	d
Kermeta2	c	d	b	a	ad	a	abg	af	bgj	c	b	a	d	c	c	i	e	c	a
Modelio	a	d	b	acd	abd	cd	ag	adf	g	c	b	a	d	c	c	i	e	c	a
Umple	a	d	b	a	ad	a	ag	a	ci	c	b	e	bc	c	d	cdefgh	abd	acg	c
MDWorkbench	b	d	b	bcd	a	a	ag	i	bg	c	b	a	d	c	c	i	e	j	a
Melange	a	d	b	a	a	a	ag	h	bgj	c	b	a	d	c	c	i	e	c	d
MagicDraw	a	d	b	cd	abd	a	acg	abdf	ad	c	b	a	d	c	c	i	e	acd	c
JAMDA	c	d	b	a	cd	b	agk	b	k	d	b	e	e	c	c	j	ad	j	d
Ente. Arch.	a	d	d	bc	ab	a	ag	adf	g	c	b	a	d	c	c	cdefgh	abd	j	c
OpenCanarias	d	d	e	e	a	e	acg	a	cg	d	b	a	e	d	c	j	f	j	d
SmartQVT	c	d	e	a	a	a	acg	a	bg	d	b	a	e	d	c	j	f	j	d
JQVT	c	d	d	a	a	e	f	a	g	b	b	d	b	b	c	cf	ab	j	d
Together	a	d	b	cd	a	e	aceg	abde	bd	c	b	a	d	d	c	i	e	k	c
Merlin	c	d	e	a	a	a	afg	a	cg	b	b	a	bc	c	c	cdfg	abd	j	d
MOFScript	c	d	e	a	ad	e	agi	i	bg	d	b	a	e	c	c	j	f	k	e
ATL	b	d	e	a	a	a	afg	a	bg	d	b	c	d	c	c	i	e	j	d
Fujaba	b	d	e	a	ad	a	a	a	ad	c	b	e	d	c	c	i	a	acf	c
GrGen.NET	b	d	a	a	cd	a	A	i	bgj	d	c	e	d	c	c	abcdefg	a	f	c
Rational	d	d	e	e	abd	e	M	i	k	c	d	e	e	c	e	j	f	k	e

Table 3. *Group 2 [5].*

Tool	T	I	O	ET	MP	RE	RT	EDI	WPM	TS	SYE	SE	MT	MS	CT	LT	MC	IN	AR	SEC
MetaEdit+	c	c	bcd	e	a	a	a	c	a	c	abd	abd	c	a	a	b	abd	c	a	a
QVTo-Eclipse	c	c	ab	e	b	b	b	a	a	b	acde	bdeg	d	c	a	b	f	b	b	b
Kermeta2	c	c	abcf	e	a	b	b	a	a	c	f	abdf	a	b	b	b	f	c	a	e
Modelio	c	a	abcde	e	b	a	a	a	a	c	f	h	c	a	b	b	f	c	a	b
Umple	c	a	abcde	e	b	a	b	c	a	c	ae	bd	c	a	a	b	ad	c	a	e
MDWorkbench	c	a	abce	e	b	a	b	a	a	c	f	bd	c	a	a	b	f	c	a	a
Melange	c	a	abc	e	a	b	b	a	a	e	f	abcdef	a	b	b	b	f	b	b	d
MagicDraw	c	c	abcdf	e	b	a	a	a	a	c	f	abd	c	a	a	b	f	c	a	b
JAMDA	a	a	bd	abd	b	b	b	b	b	d	g	j	a	c	b	b	f	b	b	e
Ente. Arch.	c	a	abde	e	b	a	a	c	a	c	f	bcde	c	a	b	b	ac	c	a	a

Tool	T	I	O	ET	MP	RE	RT	EDI	WPM	TS	SYE	SE	MT	MS	CT	LT	MC	IN	AR	SEC
OpenCanarias	d	a	h	e	b	b	b	d	c	d	h	j	d	d	b	b	f	e	c	f
SmartQVT	d	a	h	e	b	b	b	a	a	d	h	j	a	c	b	b	f	b	b	e
JQVT	a	a	b	abd	b	b	b	a	a	d	ac	bc	a	c	b	b	f	b	b	e
Together	c	c	ab	e	b	a	a	c	a	d	f	h	c	a	a	a	acd	c	a	f
Merlin	a	a	abd	e	b	b	b	a	a	d	acd	bcd	a	c	b	b	f	b	b	e
MOFScript	d	a	abcd	e	b	b	b	d	a	e	h	j	a	c	b	b	f	e	b	f
ATL	c	a	abf	e	b	b	b	a	a	e	f	h	c	a	b	b	f	c	b	b
Fujaba	c	a	abd	e	b	a	a	a	a	c	abcd	bcdefg	a	c	b	b	f	c	b	e
GrGen.NET	c	c	abcf	e	b	b	b	b	b	d	a	bdg	c	b	a	b	f	c	b	a
Rational	d	c	h	e	b	a	a	d	a	e	h	j	c	a	c	c	g	c	a	f

4. Conclusion

The introduction of Model-Driven Development means a change in the process and change of development software. Process changes take time and are expensive, while replacing the tools means lower developer productivity under the learning period and sunk costs. On the long run, switching to MDE can be beneficial in case the process works, and the process itself depends on the tools. Since the transition is expensive, it is important to invest wisely, and choose the right tool. One of the main issues in the industry is if a tool scale, or not. There are other quality aspects too. ISO/IEC9126 defines multiple aspects of quality for software as functionality, reliability, usability, maintainability, portability and efficiency. Functionality can be altered or added. Usability and maintainability can be improved. Portability can be addressed with changing the environment for the tool to execute. Scalability is on the other hand different. A solution can be highly functional, reliable, usable, maintainable and portable, if it doesn't scale, it won't be suitable for industrial use. In case the solution scales, the other issues can be solved one way or another. It can be concluded that for industrial needs, the most (and only) important aspect is scalability for MDE, but even generally. The part where scalability comes into picture with MDE is the part where the model transformation takes place. The model transformation should scale, and therefore, a model transformation tool should also scale to be suitable for industrial use. In case the transformation tool would not scale, the tool and therefore the process using it would fail, resulting in a loss of capital. The aim of this article was the evaluation of the most popular MDE tools in order to learn to identify the right tool that meets the specific needs of the software engineer without recommending any particular tool or vendors.

References

[1] Kahani, N., Bagherzadeh, M., Dingel, J., & Cordy, J. R: The problems with Eclipse modeling tools: a topic analysis of Eclipse forums. In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (pp. 227-237). ACM (2016).

[2] Juha-Pekka Tolvanen and Steven Kelly: Model-Driven Development Challenges and Solutions Experiences with Domain-Specific Modelling in Industry. Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016), pages 711-719. SCITEPRESS – Science and Technology Publications (2016).

[3] Manel Ammar, Mouna Baklouti, Maxime Pelcat, Karol Desnos, Mohamed Abid: On Exploiting Energy-Aware Scheduling Algorithms for MDE-based Design Space Exploration of MP2SoC. (2016).

[4] Alberto Rodrigues da Silva: Model-driven engineering: A survey supported by the unified conceptual model. Computer Languages Systems & Structures Elsevier (2015).

[5] Kahani, N., & Cordy, J. R: Comparison and evaluation of model transformation tools. Technical Report 2015-627 (2015).

[6] Saheed Popoola Jeffrey Carver Jeff Gray: Modeling as a Service: A Survey of Existing Tools. Department of Computer Science. University of Alabama (2015).

[7] Model-Driven Engineering of information systems. Principles, techniques and practice. Apple Academic Press, Inc. (2015).

[8] Nicolas Belloir, Jean-Michel Bruel, Raphael Faudou: Modélisation des exigences en UML/SysML, (2014).

[9] Jon Holt andSimon Perry: SysML for Systems Engineering: A Model-Based Approach (2014).

[10] Guillaume Barbier: Contribution de l'Ingénierie Dirigée par les Modèles à la Conception de Modèles Grande Culture. University Blaise Pascal – Clermont II (2013).

[11] Pascal Roques: Modélisation de systèmes complexes avec SysML. Broché (2013).

[12] Sarah CHIOUT: contribution à la modélisation logique et conceptuelle des documents numériques. Journée des Doctorants Heudiasyc (2013).

[13] Sanford Friedenthal, Alan Moore et Rick Steiner: A Practical Guide to SysML. The Systems Modeling Language. Elsevier (2012).

[14] Ahuja A., Simonin J., Nédélec R., « MDA Tool for Telecom Service Functional Design », 4th European Conference on Software Architecture (ECSA), Copenhagen, Denmark (2010).

[15] Mark R. Blackburn: What's Model Driven Engineering (MDE) and How Can it Impact Process, People, Tools and Productivity. Systems and Software Consortium, Inc. SSCI-2008002-MC (2008).