

Tool Comparison Worksheet

Automating development with Domain-Specific Modeling (DSM) involves evaluating existing tools and their features, related costs and availability of supporting services. The list below aims to help you in comparing MetaEdit+ with alternative tools that you may be considering.

With this form you can make your own comparison. Just add any tool you would like to compare MetaEdit+ with into the right column.

Language development

Tool features	MetaEdit+	Tool: _____
• Several integrated modeling languages	✓	
• Multiple simultaneous modelers	✓	
• Language definition tools (metamodeling)	✓	<ul style="list-style-type: none"> • Graphical • Form-based
• Integrated metamodeling and modeling (test the language while defining it simultaneously)	✓	Integrated tools
• Automated language deployment to modelers	✓	
• Metamodel coverage for rules and constraints of the language	✓	<ul style="list-style-type: none"> • Rule and constraint definition tools • Generator-based constraints • Regular expressions
• Metamodel checking during language definition	✓	
• Traceability between elements of a metamodel	✓	Bi-directional: <ul style="list-style-type: none"> • Who is using the element • What elements are used
• Metamodel evolution	✓	Deployed language can be safely changed
• A metamodel specifies one or multiple languages	✓	Both possible
• Automatic models-updating upon metamodel change	✓	Models will always open in editors
• Support for old language versions	✓	
• A repository to handle various metamodels	✓	Object Repository
• Import and export of metamodels	✓	<ul style="list-style-type: none"> • XML • Binary formats These can cover also <ul style="list-style-type: none"> • Notation (symbols) • Constraints • Generators
• Library of metamodels	✓	70+ languages available
• Library of metamodels	✓	70+ languages available
• Extend UML with stereotypes, tagged values and constraints	✓	Possible, but why? <ul style="list-style-type: none"> • Modifying a metamodel gives more power • Can also remove unnecessary elements, add new types, constraints etc.

Symbol and representation definition

Tool features	MetaEdit+	Tool: _____
• Specifying notation for the language	✓ Draw in Symbol Editor or import symbols	
• Rich notation (more than just an icon for nodes and a line for arcs)	✓ Symbols for objects, relationships, roles and properties	
• Symbols can change based on model data	✓ <ul style="list-style-type: none"> • Generator-based symbol elements • Conditional symbols • Regular expressions • Ports in symbol to connect with 	
• Library of symbols available	✓ Several hundred notational elements	
• Import (and export) symbols	✓ <ul style="list-style-type: none"> • Scalable vector graphics (SVG) • Bitmaps 	
• Access and modify notation with any programming language/platform	✓ SOAP/.NET/Web services API	
• Modeling tool development automated	✓ Editors immediately available with <ul style="list-style-type: none"> • icons, menus, toolbars • Copy & paste, replace, trace • Auto-layout, undo & redo etc. 	

Generator definition

• Generator and metamodel are integrated	✓	
• Supports straight model-to-code transformations	✓	No intermediary formats needed
• Supports model-to-model transformations	✓	Between the same or different metamodels
• Support for templates, visitor pattern, crawlers and multiple streams	✓	
• Syntax highlighting and output filtering	✓	Generator Editor
• Generator debugging	✓	Generator Debugger
• Generated and hand-written code can be separated	✓	
• Protected blocks can be defined into the generated result	✓	
• Many to many mappings from models to files	✓	
• Read and parse external files	✓	

Modeling with your language

• Support for different kind of editors	✓	<ul style="list-style-type: none"> • Diagram • Matrix • Table
• Support for modeling operations (copy/paste, reuse, replace, group, layout, grids, zooms, model hierarchies etc)	✓	
• Support for multiple concurrent modelers	✓	<ul style="list-style-type: none"> • Multi-user • Single-user functionality
• Support for large number of models	✓	4 billion model objects per project
• Automatic validation of models based on the metamodel	✓	
• Constraints can be checked at modeling time or when wanted	✓	Automatically available based on either <ul style="list-style-type: none"> • Metamodel • Model-checking reports
• Link and reuse among models	✓	As defined in the metamodels
• View changes graphically	✓	Changes highlighted directly in the models

Modeling tool features (continue)	MetaEdit+	Tool: _____
• Models update automatically if a new language version is available	✓ If desired	
• Models can be saved and stored using the older language version	✓ If desired	
• Model browsing tools	✓ Various browsers with filtering and different hierarchies	
• Document generation	✓ Predefined, customizable generators <ul style="list-style-type: none"> • RTF, Word, HTML 	
• Metrics and model checking	✓	
• Traceability between model elements	✓	
• Automated trace on modeling history and change comparison	✓ Change and Version Tool with <ul style="list-style-type: none"> • Tree view, graphical, textual 	
• Import and export of models	✓ <ul style="list-style-type: none"> • XML • Binary format • Includes also representations 	
• Model-to-model transformation	✓ <ul style="list-style-type: none"> • With Generator Editor using XML as intermediary format • Directly using the API 	
• Access and modify models with any programming language/platform	✓ SOAP/Web services API available for almost any programming language and platform	
• Reverse engineering	✓ <ul style="list-style-type: none"> • With XML import • API • Generator language (MERL) 	
• Trace back from generated code to models	✓ Live-code function: click the generated code to see the model	
• Animate models during program execution	✓ Animate with API	
• UML support	✓ UML metamodel is one of the many included	

Other

• IDE integration	✓ <ul style="list-style-type: none"> • Eclipse plug-in • Visual Studio extension 	
• Version control system integration	✓ <ul style="list-style-type: none"> • SVN, GIT, etc. 	
• Can be integrated with other tools (simulators, versioning, emulator, etc).	✓ Easy as based on most commonly applied integration techniques <ul style="list-style-type: none"> • Web services/SOAP/.NET • XML • Command line interface 	
• Easy to use and learn	✓	
• Other technologies required	⊘	
• Runs on different platforms	✓ <ul style="list-style-type: none"> • Multiple platforms • Heterogeneous platform set-up 	
• Manuals and tutorial available	✓ <ul style="list-style-type: none"> • Modeling • Metamodeling • Sysadmin 	

Supporting services

• Maintenance	✓ Covering minor and major tool upgrades	
• Support and helpdesk	✓ On-site, forum, email, phone, fax	
• Training services	✓ <ul style="list-style-type: none"> • DSM creation for experts • DSM use for modelers • Organizational change for process, management 	
• Consulting services	✓	