

Turning a Standard UML Modeling Tool into a SysML/ModelicaML Modeling Tool

by use of **MetaModelAgent**

Thomas Wiman

thomas.wiman@adocus.com

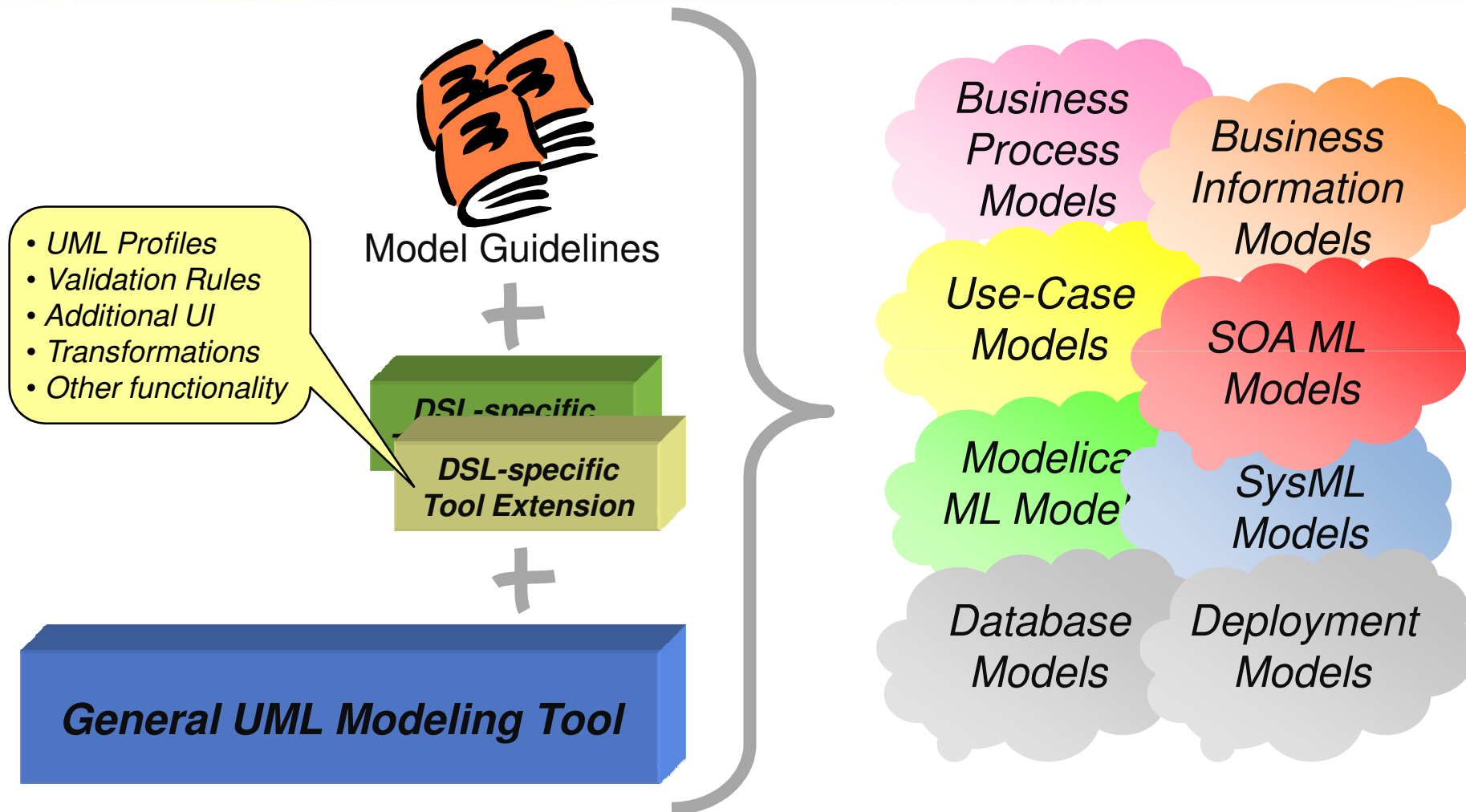
Unified Modeling Language (UML)

- General purpose visual modeling language
 - Business Modeling
 - Software Modeling
 - System Modeling
 - ...
- No one needs UML as is
 - Too many and general concepts
 - Too general semantics
- You need a domain specific modeling language (DSL)
 - Domain specific concepts
 - Domain specific precise semantics

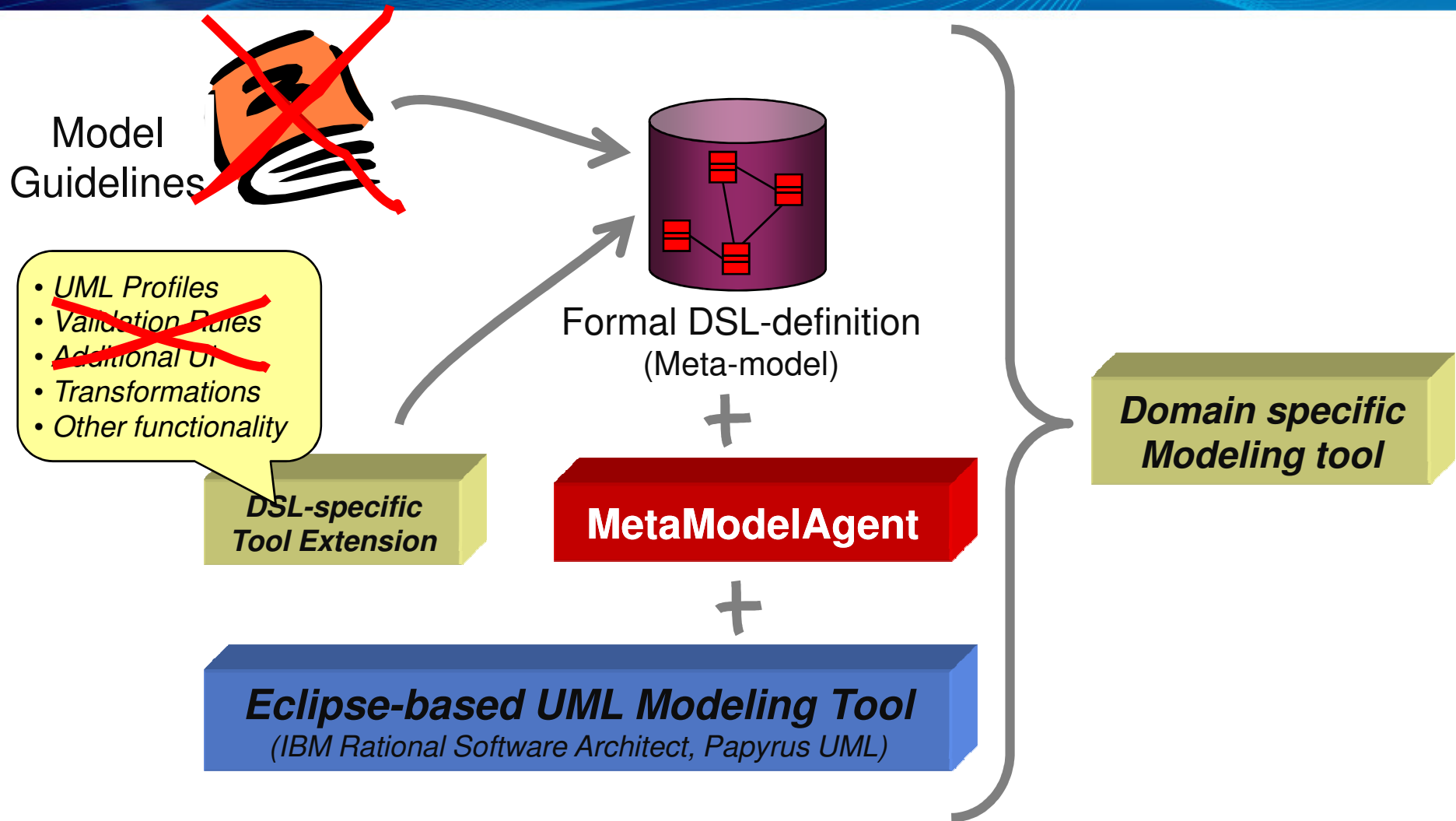
UML and ModelicaML

Abstraction, AcceptCallAction, AcceptEventAction, Action, ActionExecutionSpecification, ActionInputPin, **Activity**, ActivityEdge, ActivityFinalNode, ActivityGroup, ActivityNode, ActivityParameterNode, ActivityPartition, Actor, AddStructuralFeatureValueAction, AddVariableValueAction, AnyReceiveEvent, Artifact, Association, AssociationClass, Behavior, BehaviorExecutionSpecification, BroadcastSignalAction, CallAction, **CallBehaviorAction**, CallEvent, CallOperationAction, CentralBufferNode, ChangeEvent, **Class**, ClassifierTemplateParameter, Clause, ClearAssociationAction, ClearStructuralFeatureAction, ClearVariableAction, Collaboration, CollaborationUse, CombinedFragment, **Comment**, CommunicationPath, Component, ComponentRealization, ConditionalNode, ConnectableElement, ConnectableElementTemplateParameter, **ConnectionPointReference**, **Connector**, ConnectorEnd, ConsiderIgnoreFragment, **Constraint**, Continuation, **ControlFlow**, ControlNode, CreateLinkAction, CreateLinkObjectAction, CreateObjectAction, CreationEvent, DataStoreNode, **DataType**, **DecisionNode**, Dependency, DeployedArtifact, Deployment, DeploymentSpecification, DeploymentTarget, DestroyLinkAction, DestroyObjectAction, DestructionEvent, Device, Duration, DurationConstraint, DurationInterval, DurationObservation, Element, ElementImport, EncapsulatedClassifier, **Enumeration**, EnumerationLiteral, Event, ExceptionHandler, ExecutableNode, ExecutionEnvironment, ExecutionEvent, ExecutionOccurrenceSpecification, ExecutionSpecification, ExpansionNode, ExpansionRegion, Expression, Extend, Extension, ExtensionEnd, ExtensionPoint, Feature, FinalNode, FinalState, FlowFinalNode, ForkNode, **FunctionBehavior**, Gate, **Generalization**, GeneralizationSet, GeneralOrdering, Image, Include, InformationFlow, InformationItem, **InitialNode**, InputPin, InstanceSpecification, InstanceValue, Interaction, InteractionConstraint, InteractionFragment, InteractionOperand, InteractionUse, Interface, InterfaceRealization, InterruptibleActivityRegion, Interval, IntervalConstraint, InvocationAction, JoinNode, Lifeline, LinkAction, LinkEndCreationData, LinkEndData, LinkEndDestructionData, LiteralBoolean, LiteralInteger, LiteralNull, LiteralSpecification, LiteralString, LiteralUnlimitedNatural, LoopNode, Manifestation, MergeNode, Message, MessageEnd, MessageEvent, MessageOccurrenceSpecification, Model, Node, ObjectFlow, ObjectNode, Observation, OccurrenceSpecification, **OpaqueAction**, **OpaqueBehavior**, OpaqueExpression, Operation, OperationTemplateParameter, OutputPin, **Package**, PackageImport, PackageMerge, **Parameter**, ParameterSet, PartDecomposition, Pin, **Port**, PrimitiveType, **Property**, ProtocolConformance, ProtocolStateMachine, ProtocolTransition, **Pseudostate**, QualifierValue, RaiseExceptionAction, ReadExtentAction, ReadIsClassifiedObjectAction, ReadLinkAction, ReadLinkObjectEndAction, ReadLinkObjectEndQualifierAction, ReadSelfAction, ReadStructuralFeatureAction, ReadVariableAction, Realization, ReceiveOperationEvent, ReceiveSignalEvent, Reception, ReclassifyObjectAction, RedefinableTemplateSignature, ReduceAction, **Region**, RemoveStructuralFeatureValueAction, RemoveVariableValueAction, ReplyAction, SendObjectAction, SendOperationEvent, SendSignalAction, SendSignalEvent, SequenceNode, Signal, SignalEvent, Slot, StartClassifierBehaviorAction, **State**, StateInvariant, **StateMachine**, StringExpression, StructuralFeature, StructuralFeatureAction, StructuredActivityNode, StructuredClassifier, Substitution, TemplateableElement, TemplateBinding, TemplateParameter, TemplateParameterSubstitution, TemplateSignature, TestIdentityAction, TimeConstraint, TimeEvent, TimeExpression, TimeInterval, TimeObservation, **Transition**, Trigger, Type, UnmarshallAction, Usage, UseCase, ValuePin, ValueSpecification, ValueSpecificationAction, Variable, VariableAction, Vertex, WriteLinkAction, WriteStructuralFeatureAction, WriteVariableAction

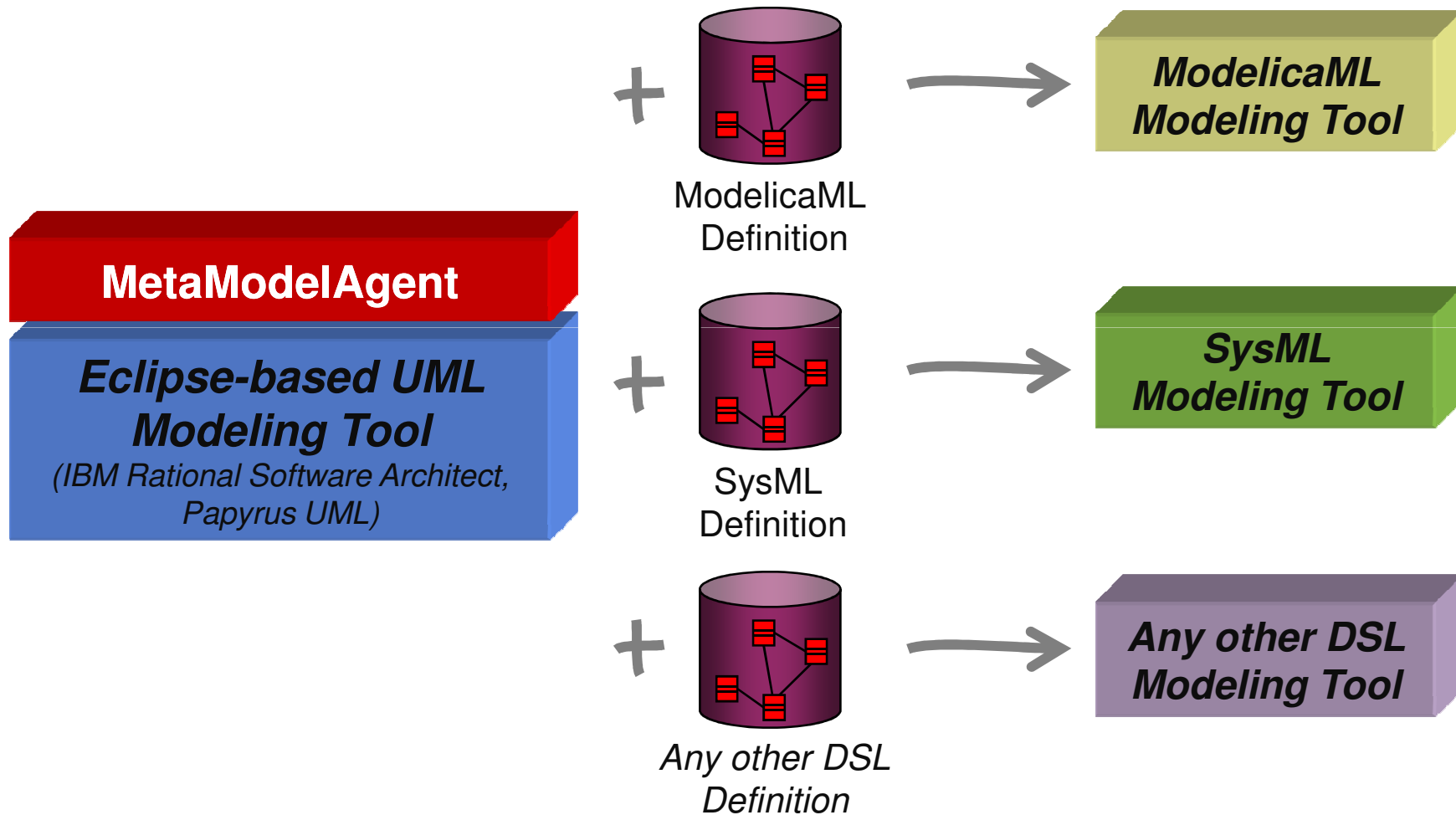
Domain Specific Modeling



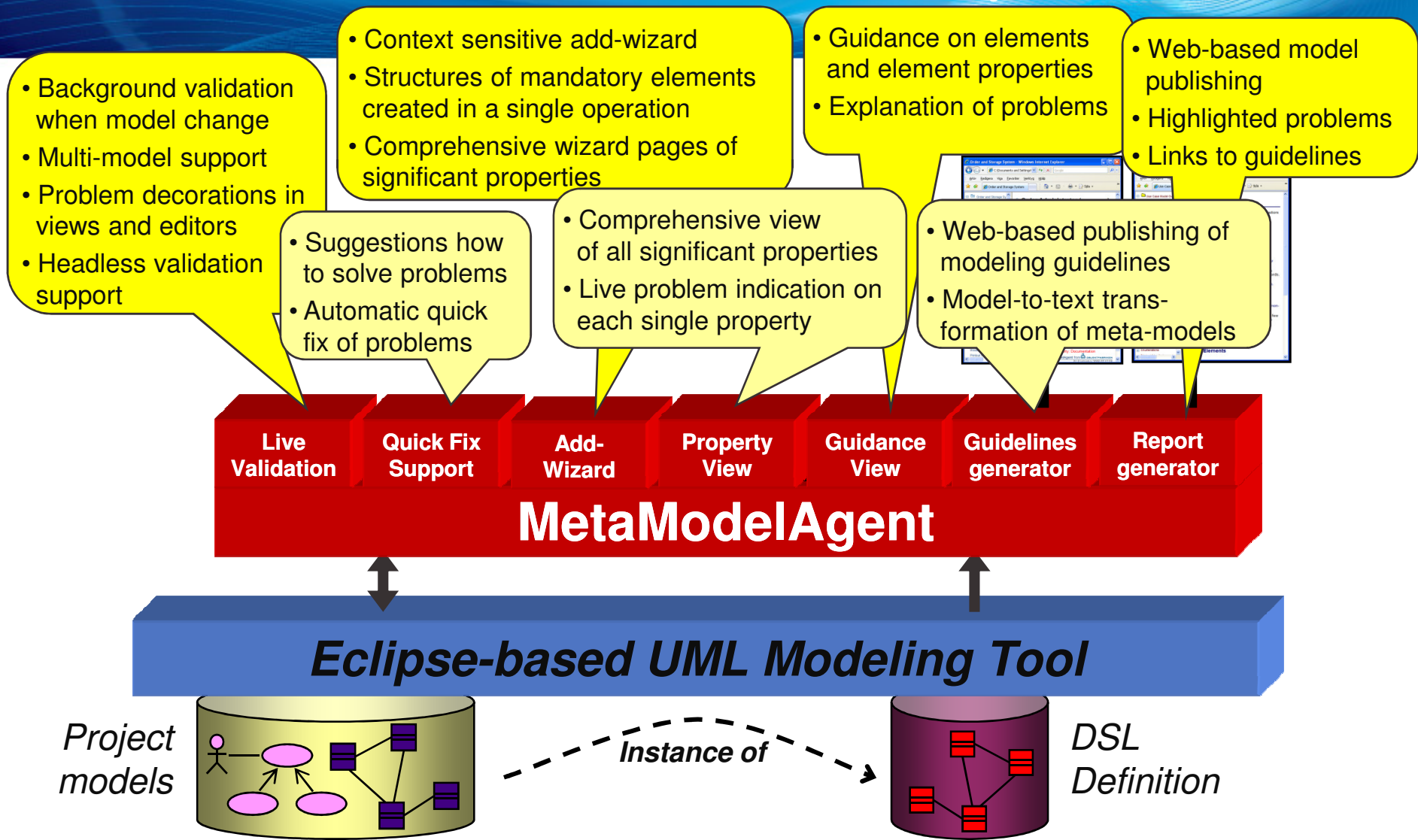
The concept of MetaModelAgent



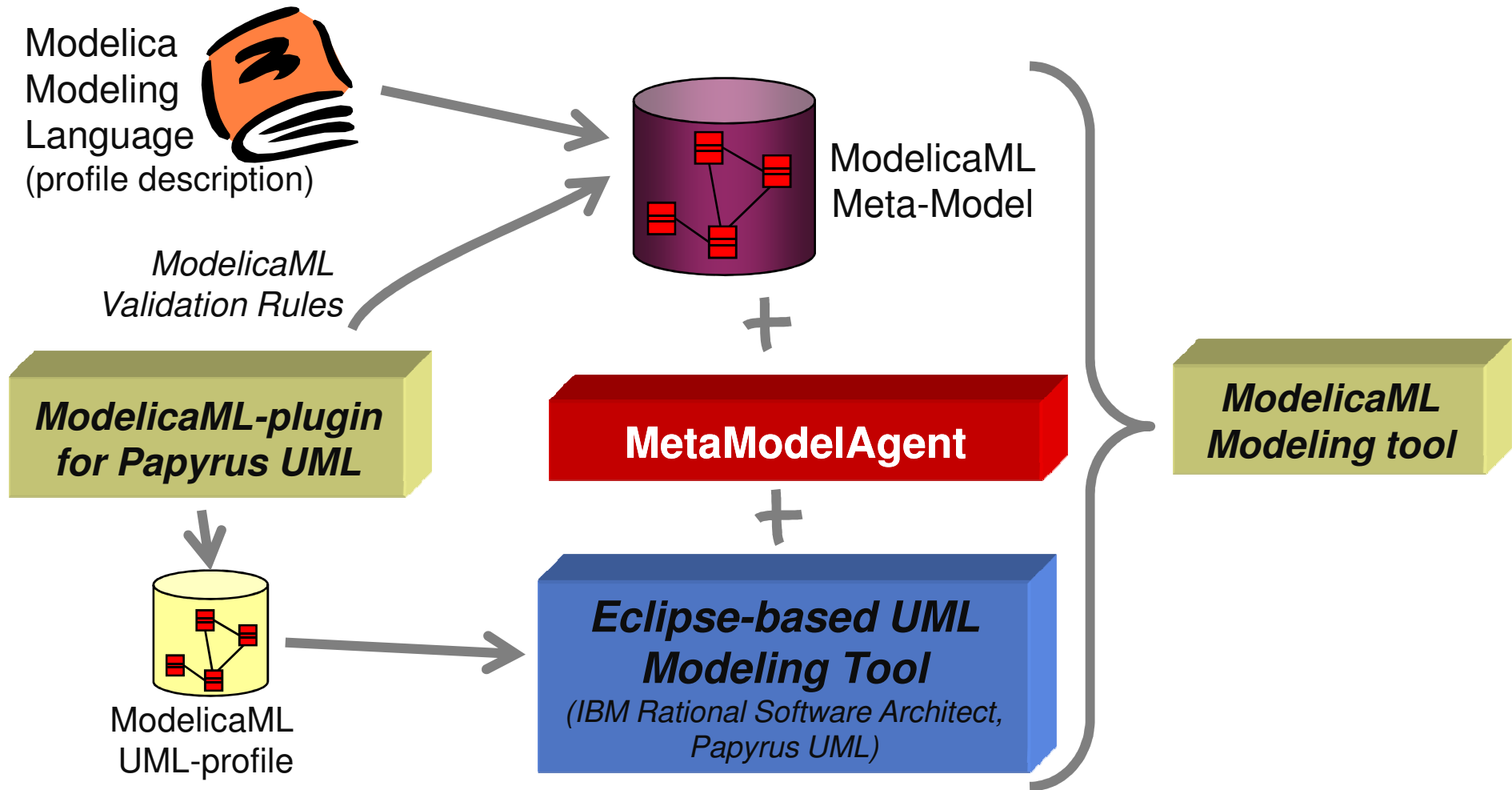
Adapted tool for each purpose



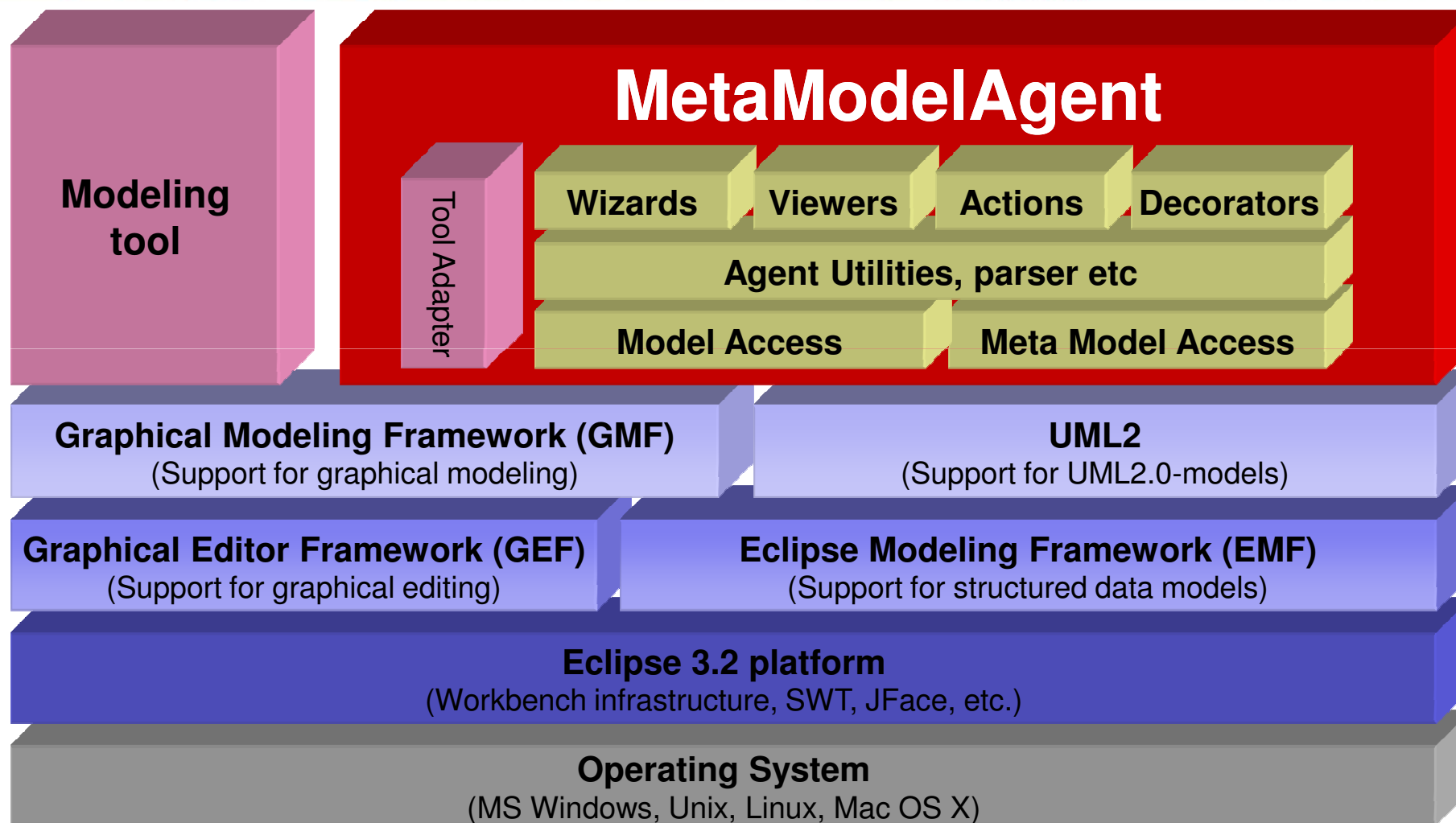
Main functionality



Implementing ModelicaML



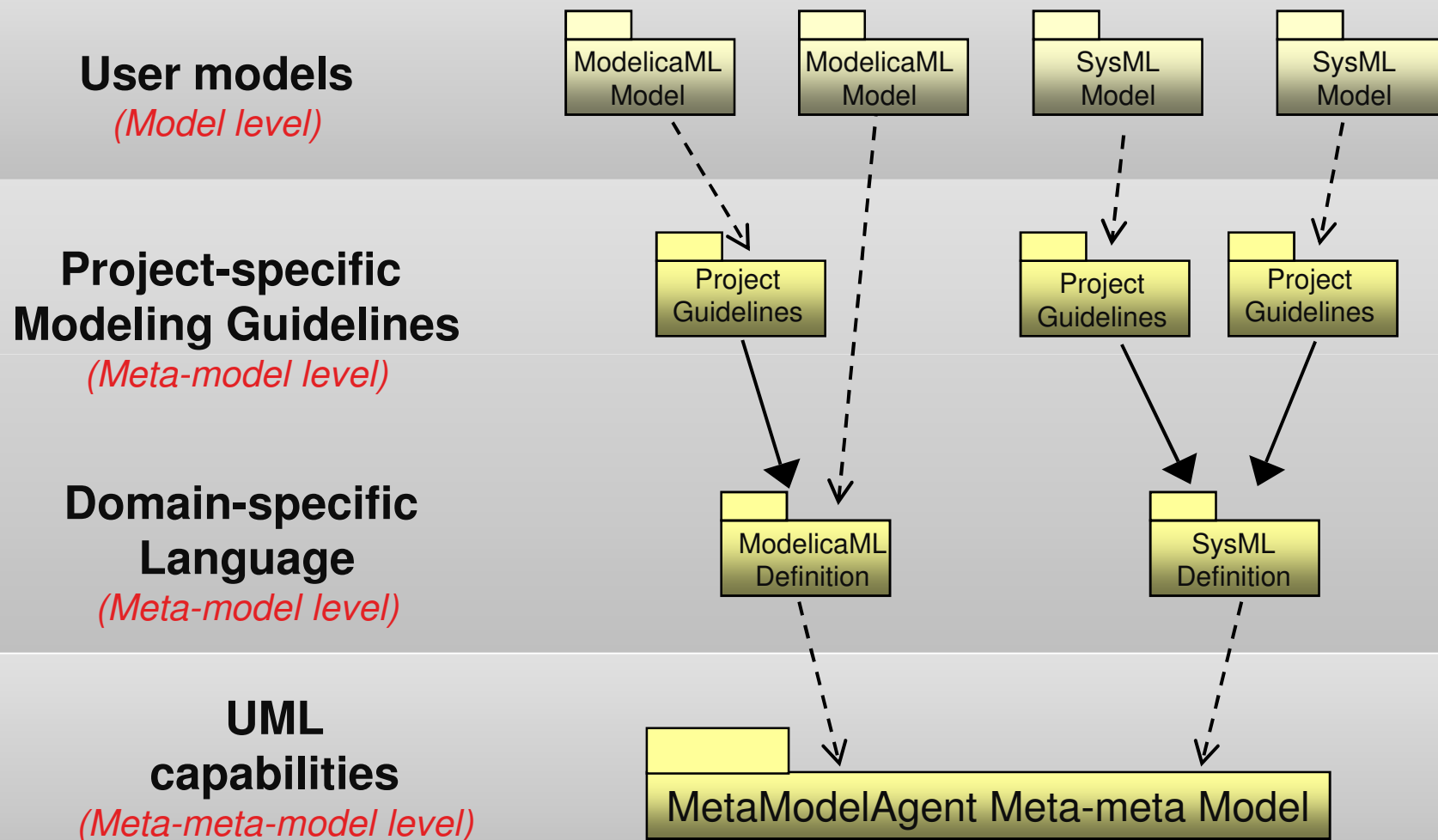
Architecture





Demo

Model-driven DSL-definition approach




Contribution to Papyrus/ModelicaML

- Live validation
 - gives immediate feedback on ModelicaML compliance
- ModelicaML-sensitive UI
 - Wizards and Views with embedded guidance
- Extended ModelicaML validation
 - Makes it possible to define and validate additional ModelicaML-rules not yet supported in the ModelicaML plug-in
- Organization specific modeling guidelines
 - Makes it possible for local organizations to further constraint ModelicaML based on their own need
- Enables ModelicaML-modeling in other Eclipse-tools
 - E.g. IBM Rational Software Architect

MetaModelAgent Summary

- Model-driven tool customization
- Ensures compliance to any domain-specific language
- No impact on your development environment
- Easy to learn and manage
- Scalable to large organizations
- General available for IBM Rational Software Architect
 - *Early Beta-release available for Papyrus UML*



For more info about
MetaModelAgent
please visit

www.adocus.com