Systematica™

Do more with less

# Abbreviated S*Metamodel Glossary—Ordered by Concept

| Term | Definition |
|---|---|
| ***Terms for Systems*** | |
| System | A collection of interacting Components. |
| Component | A part of a system, capable of interacting with other components. |
| Interact | Two components are said to interact if one impacts the state of the other, through exchange of energy, forces, mass flows, or information. |
| Sub-system | A Component of a system, which is itself a system. |
| Subject System | A System that is being specified or is the focus of attention. |
| Environment | The total system which is external to a Subject System and which has interactions with it of interest. |
| Actor | A component of the Environment of a Subject System that directly interacts with the Subject System. |
| Logical System | A system identified solely by its externally viewable behavior or responsibility. |
| Physical System | A system identified solely by its physical identity or make-up. Physical Systems are Design Components that fulfill Functional Roles allocated to them. |
| ***Terms for System Behavior*** | |
| Functional Interaction | An interaction of Components (which may themselves be Systems), expressed as a relationship in which at least one affects the state of another. Sometimes also called Interaction or Collaboration. (See Interact.) |
| Functional Role | The behavioral description (and therefore a Logical System) of a part played by a System in a Functional Interaction's relationship. |
| Sub-Interaction | A Functional Interaction that is a part of (decomposes) a larger Functional Interaction. |
| Feature | A collection of Functional Interactions providing stakeholder value, valued services, fitness for some purpose, or the basis for selection. Expressed in language and concepts of Stakeholders in related value or fitness, and modeling stakeholder requirements. See also Service. |
| Service | Same as Feature, but with further implications of consumed value and levels of service performance. |
| Input-Output | That which is externally exchanged between interacting systems. Abbreviated as I-O. Typically flow Energy, Force, Mass, or Information. |
| Architectural Relationship | A relationship that summarizes the architectural significance of a relationship between system components (which may themselves be systems). Typically arises from and summarizes Interactions between the related entities. |
| System of Access | A System providing the means of access for interactions between components (which may themselves be systems). |
| Interface | The association of a System (which "has" the Interface) with a set of its Functional Interaction(s) (describing behavior at the Interface), Input-Output(s) (which pass through the interface), Architectural Relationship(s) (which summarize the Interactions), and System(s) of Access (which provide a means of interaction). See Figure 3 below. |

# Abbreviated S*Metamodel Glossary—Ordered by Concept

| Term | Definition |
|---|---|
| ***Terms for Modeling System States or Modes*** | |
| State | The condition of a system component that determines its interactive behavior, viewed externally from the component. That state may in turn be changed by interactions. A particular state may in some cases be thought of as a situation. |
| State Transition | A change of state from one condition to another. |
| Event | An occurrence that triggers a transition from one state to another. |
| Sub-state | A state (situation) that occurs during, but not necessarily throughout, another state (situation). |
| Concurrent state | A state (situation) that occurs simultaneously with another state (situation). |
| Use Case | A State (situation) of an Environment of a System during which certain Functional Interactions are performed. |
| ***Embedded Intelligence (EI) Pattern: Terms for Modeling Management of Systems (By Human or Automated Managers)*** | |
| MDS | Managed System—A System that provides services to a System of Users, and that is managed by a Management System. |
| SOU | System of Users—A System that consumes services from a Managed System or a Management System. |
| MTS | Management System—A System that manages the Performance, Faults, Configuration, Security, or Accounting of a Managed System. |
| SOA | A System providing the means of access for interactions between other Systems. |
| SMFA | System Management Functional Area—All system management functions fall into these five SMFAs: Fault Management, Performance Management, Security Management, Configuration Management, and Accounting Management. |
| ***Terms for Modeling Hierarchies, Relationships, and Attributes of Classes*** | |
| Class | A set of things considered similar to each other by virtue of their membership in the class. |
| Superclass, Subclass | A class is a superclass of another class (called a subclass) if it generalizes the subclass. As sets, a subclass is a subset of a superclass. |
| Relationship | A statement about several classes that may be true or false. If true, the classes are said to be in that relationship with each other. |
| Hierarchy | A sequence of classes, related to each other sequentially by the same type of relationship. |
| Class Hierarchy | A General-Special hierarchy, in which each progressive layer is a more specialized case of the layer above it. (“Is a type of”) |
| Containment Hierarchy | A Whole-Part hierarchy, in which each progressive layer is a part of the layer above it. (“Is a part of”) |
| Attribute | A property or characteristic of members of a class, capable of taking on values to describe members (instances) of the class. |
| Attribute Coupling | A description of how attribute values of one attribute impact attribute values of another attribute, whether due to physical laws, commercial capabilities, human preference, or other value dependencies. |

# Abbreviated S*Metamodel Glossary—Ordered by Concept

| Term | Definition |
|---|---|
| ***Terms for Managing and Applying System Models and Patterns*** | |
| S*Metamodel | A modeling framework used to describe systems, with the goal of providing the smallest model for purposes of engineering or science. Figure 1 is an informal summary of some key aspects. |
| S*Metaclass | One of the S*Metamodel foundation classes used to formally describe systems and system related information, for purposes of engineering or science. S*Metaclasses include System, Functional Interaction, State, Feature, Interface, Input-Output, etc. |
| S*Model | A system model which is consistent with (formed using) the S*Metamodel |
| S*Pattern | An S*Model of a family of systems, capable of being configured and re-used as S*Models of specific cases. |
| Gestalt Rules™ | Rules that tell us whether a pattern is a special case conforming to a more general pattern. |
| Configuration (of a Pattern) | Configuration is a highly constrained operation on an S*Pattern that produces a configured S*Model result. A configuration of an S*Pattern is the result of only populating (or depopulating) the items described by the pattern, and setting values of attributes described by the pattern—no other operations allowed. (No changes to names, definitions, or attributes, except for their values.) The items populated may be any S*Metaclasses (Systems, Features, States, etc.) found in the pattern. A configured S*Pattern is an S*Model that conforms to the S*Pattern that describes a family of systems. |
| Specialization (of a Pattern) | Specialization is similar to configuration, but less restrictive. A specialization of a pattern is the result of making some of the items described by the pattern more specific, including the option to specialize names and definitions of items in the pattern, and adding attributes to those items in addition to those in the pattern. The items populated may be any metaclasses (Systems, Features, States, Services, etc.) found in the pattern. A specialized system conforms to (satisfies the Gestalt rules for) the pattern that describes its family. |
| Configuration Rule | A rule that restricts a configuring person's population/de-population options and choice of attribute values, during pattern configuration. |
| Use (as a noun) | A configured application of a class or pattern of classes. (More than one of these of the same type are re-uses of the same assets.) Note that a Use is a class, not an instance (member) of the class. Strictly speaking, a use is a subclass of its parent pattern class. |
| ***Terms for Modeling System Requirements and Designs*** | |
| Requirements | What a Subject System must be or do as seen by the environment with which it interacts, without regard to how this is accomplished internal to the System. The sum of all the Functional Interactions required of the System, including the attributes of these interactions (e.g., capacity, speed, cost, reliability, etc.). |
| High Level Requirements Model | The combination of a Domain Model, Feature Framework, State Model, Functional Interaction Model, and Logical Architectural Model that determines the overall framework with which detailed requirements analysis can be performed and managed. |
| Domain Model | The combination of a Domain Diagram, showing the interacting systems of a particular domain, and the definitions of the Systems, Interfaces, Input-Outputs, and Architectural Relationships that appear on it. |
| Feature Framework | Linked set of system Stakeholders, Advocates, Stakeholder Requirements, Features, Attributes, describing a system fitness (trade) space. |

# Abbreviated S*Metamodel Glossary—Ordered by Concept

| Term | Definition |
|---|---|
| Stakeholder | A person or other entity having a stake (interest) in some aspects of the fitness or value of a system. |
| Stakeholder Requirement | A statement (either in formal or informal language) in the language and concepts of stakeholders, that implies formally modeled requirements or design constraints upon a system. |
| Stakeholder Advocate | An Advocate represents a Stakeholder during the elicitation of Stakeholder Requirements and Validation of the Requirements. |
| State Model | The combination of a system's State Transition Diagram, the definitions of each State and Event, and a list of defined functional interactions that are expected during each State. |
| Interactions Overview Model | A set of defined Functional Interactions associated with the Features each supports and States during which each is required. |
| Logical Architecture Model | A Logical Architecture Diagram, decomposing a system's external (black box) behavior into interacting logical (white box) subsystems, their Architectural Relationships (and/or Input-Outputs), Attributes, and related definitions. |
| Interaction Diagram | A diagrammatic model representation of functional roles of an interaction and the input-outputs they exchange during the interaction. |
| Detail Level Requirements Model | The set of Interaction Diagrams; Input-Output Definitions and Relationships; Functional Role Definitions, Requirement Statements, and Relationships; Role Attribute Definitions, Values, and Functional Role Attribute Couplings to Feature Attributes for each identified Functional Interaction. |
| Requirement Statement | A behavioral description relating a Functional Role's Inputs, Outputs, and Attributes, describing behavior during an Interaction. |
| Design (as noun) | The plan of how a set of Requirements will be met or accomplished by the internal physical components and relationships of a Subject System. Consists of physical architecture, plus decomposition of required black box behavior to logical architecture, further decomposed to functional roles, then allocated to components of the physical architecture. |
| High Level Design Model | The combination of a Physical Architecture Diagram, an Allocation Table, Attribute Couplings between Physical System Attributes and Functional Role Attributes, and definitions of each Physical System. |
| Requirements Validation | The process of ascertaining that stated candidate (specified) Requirements for a System correctly state the needs of its stakeholders. Also used to refer to System Validation--the process of ascertaining that a candidate System conforms to the *actual* needs of such a system authority (as opposed to meeting the *specified* Requirements). |
| Design Verification | The process of ascertaining that the specified Requirements have been (or could be) successfully met by a candidate System design. |
| Model Validation | The process of confirming that a conceptual model adequately describes, for a given purpose and model user, a system of interest. |
| Model Verification | The process of confirming that an implemented computational model adequately implements, for a given purpose and model user, a conceptual model. |

# Abbreviated S*Metamodel Glossary—Ordered by Concept

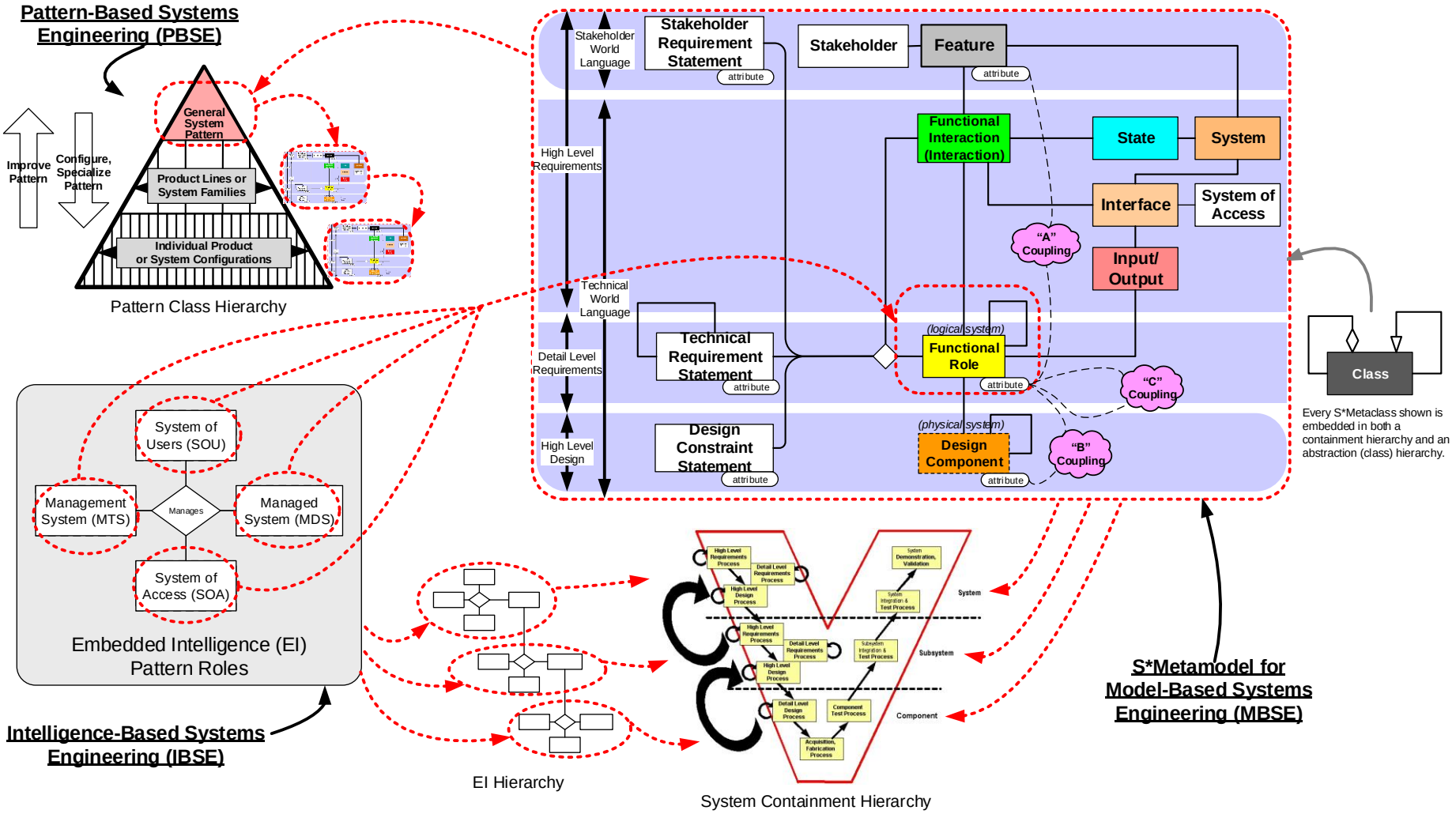



**Figure 1: Informal summary of S*Metamodel, embedded in PBSE and IBSE frameworks**

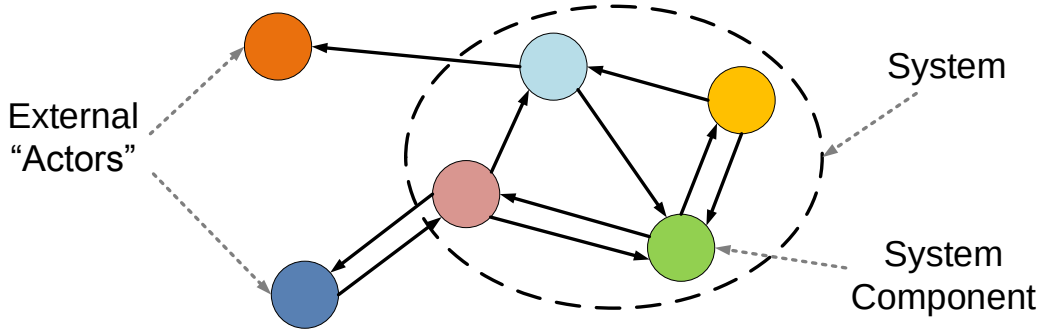



**Figure 2: The System Perspective**

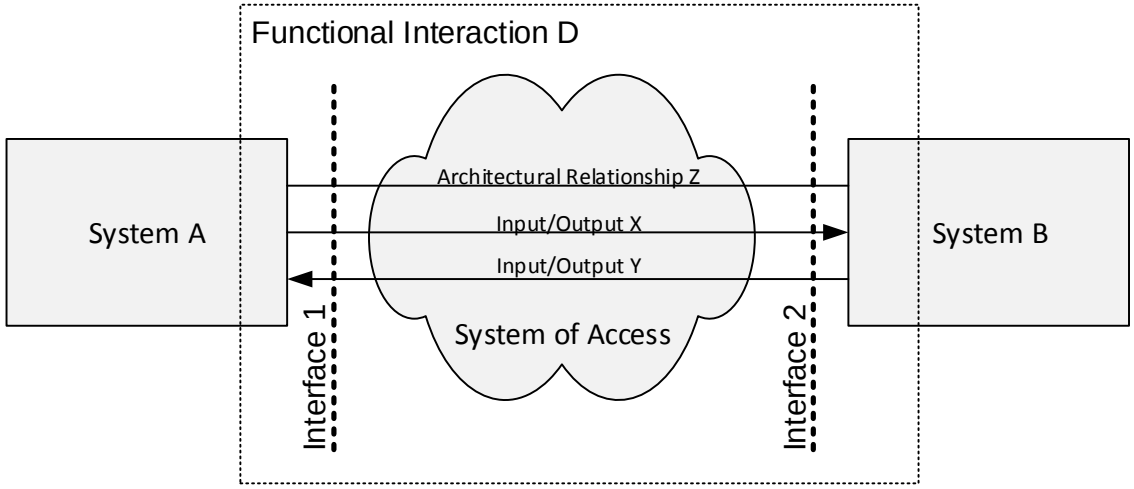
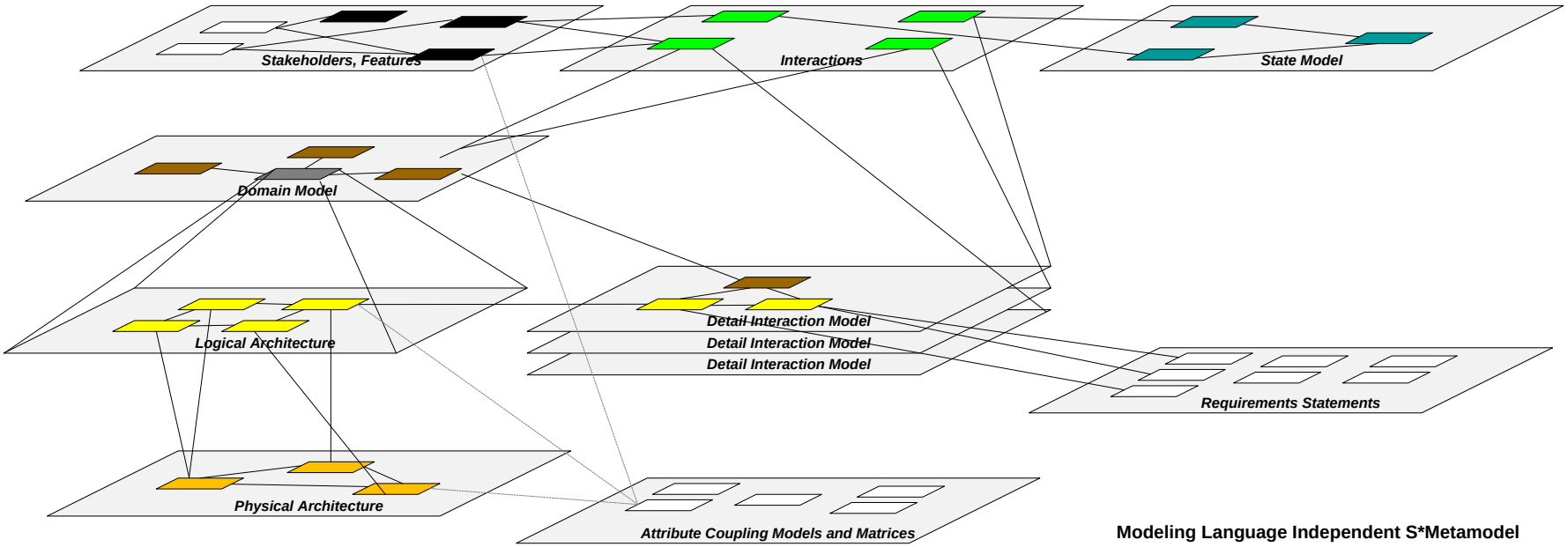



**Figure 3: System Interactions, Systems of Access, Interfaces**

# Abbreviated S*Metamodel Glossary—Ordered by Concept



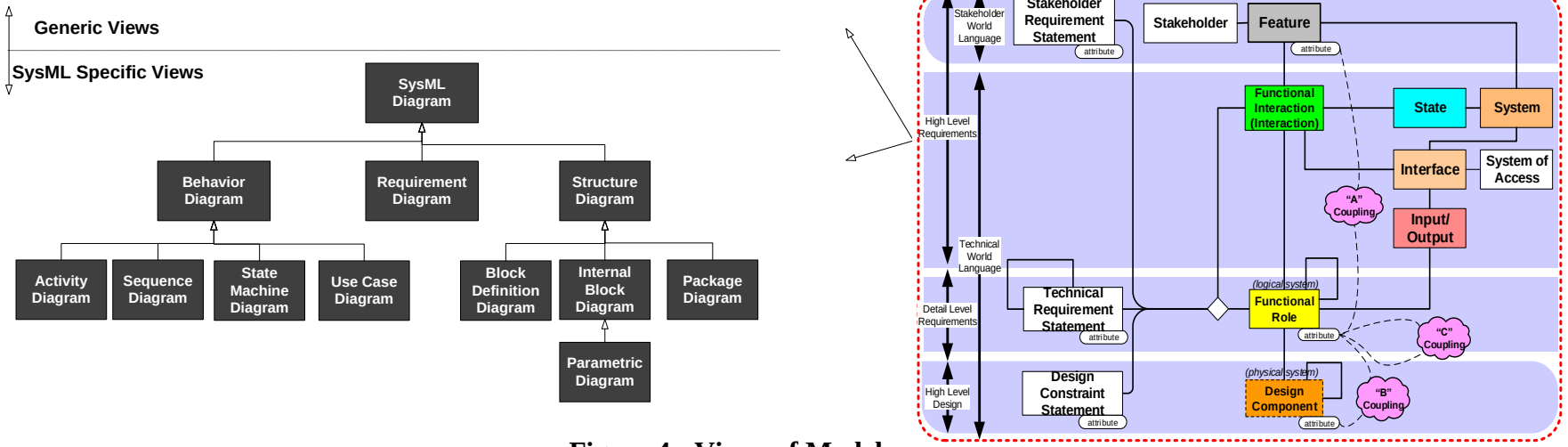




**Figure 4: Views of Models**