ParaMDO-PIM DSML Definition

1. Grammar definition

PIM model, represents the model used to describe the computational logic of the CAE software. The PIM model can be represented by a 7-tupl:

$$\mathrm{PIM} = < \mathrm{Phase}, \mathrm{Node}, \mathrm{Flow}, \mathrm{Connector}, \mathrm{Variable}, \mathrm{Start}, \mathrm{End} >$$

in which:

(1) $Phase = \{phase_1, phase_2, \dots, phase_n\}$ represents basic blocks of computational activities in the PIM process.

(2) $Node = \{node_1, node_2, \dots, node_n\}$ represents the sequential executing activities contained in the basic block.

(3) $FLow = PimFlow \cup CloneStartFlow \cup CloneEndFlow \cup \mathrm{MutexFlow}$ represents the set of connections.

(4) $Connector = Loop \cup Mutex \cup Clone$ represents the set of logic nodes.

(5) $Variable = \{variable_1, variable_2, \dots, variable_n\}$ represents the set of variables used by the phases and nodes.

(6) Start = $\{start\}$ represents the start point in the model.

(7) End = $\{end\}$ represents the end point in the model.

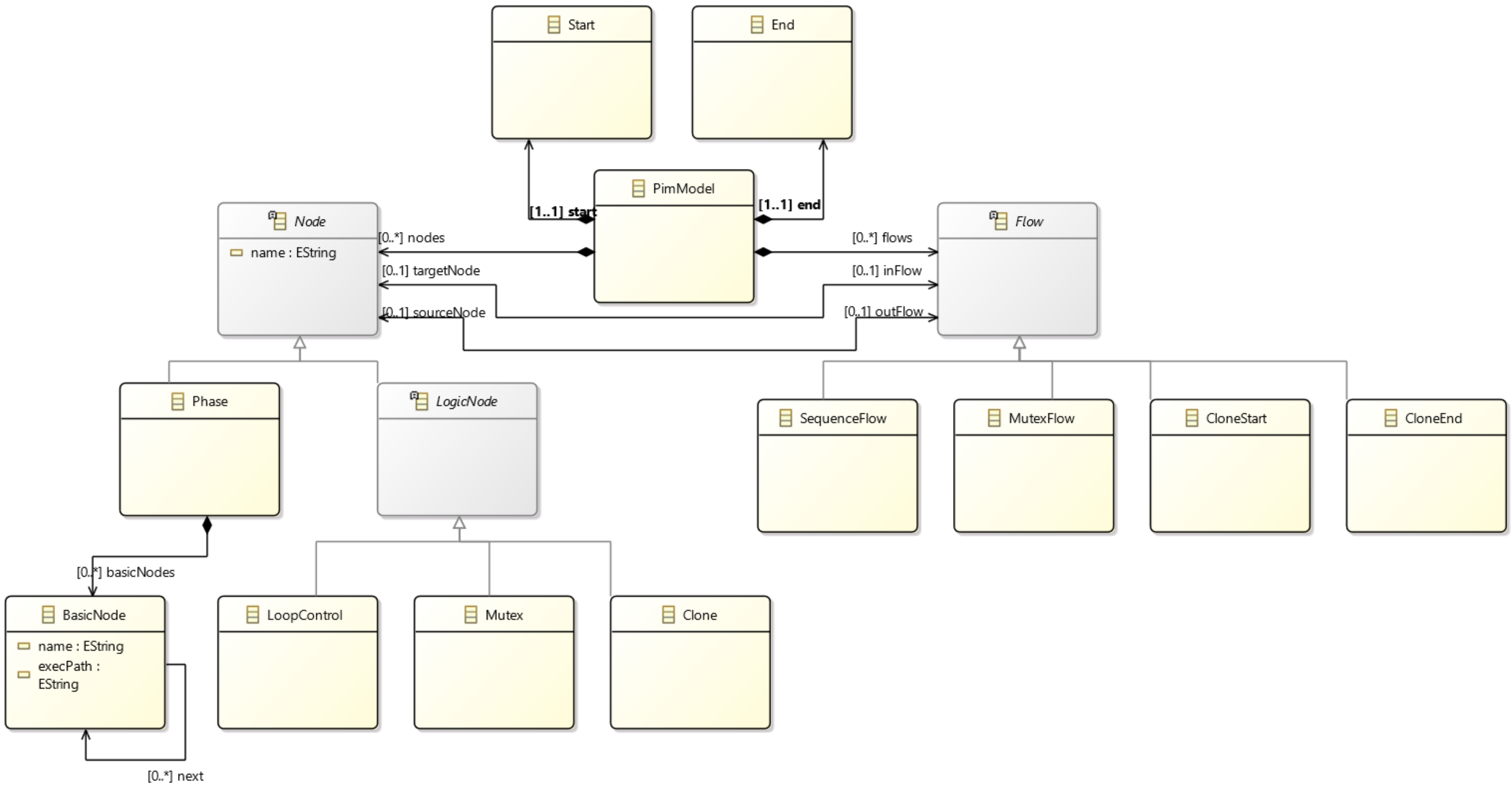


Figure 1 PIM Metamodel

2. Semantics definition

The PIM language elements include Phase nodes, logical nodes, start as well as end nodes.

Table 1 shows the detailed semantic description of Phase node in PIM. Table 2 shows the semantic description of Start and End in PIM model.

Table 1 Phase Semantic Description

Element	Description
Phase	A basic block representing the composition of computational activities, containing a Phase subgraph with several sequentially executed leaf activities of the same type

Table 2 Start / End Semantic Description

Element	Description
Start	Represents the start point of PIM model
End	Represents the end point of PIM model

Table 3 shows the semantic description of logical nodes in PIM model.

Table 3 Logic Node Semantic Description

Element	Description
Loop	Decide which output flow is activated. When the loop count is greater than the end-of-loop variable, exit the loop body and go to the true stream, otherwise continue the loop and go to the false stream
Mutex	Represents the connected activities are not allowed to be executed at the same time

Clone	Indicates that the Phase sequence between the parallel start point marker and the parallel end point marker will be executed in parallel. Creating multiple task instances

In addition to nodes, there are six connection relations included in PIM. Table 4 provides a detailed semantic description of the connection relations.

Table 4 Connection Semantic Description

Connection	Description	Usage
Sequence Flow	Sequential and causal relationships between Phases, where the starting Phase completes execution and the pointing Phase begins execution	Connections between various nodes
Clone Start	Used to mark the beginning of the computational basic block of a single program with multiple data streams for parallel computation	Starts at a single program node and points to the Phase computation basic block
Clone End	End-of-computation basic block or logical connective used to mark the end of a single program with multiple data streams for parallel computation	Starts at a single program node and points to Phase computing basic blocks or logical connectors
Mutex Flow	One end points to a Phase and the other to a mutex linker, indicating that the associated Phases need to compete for the mutex lock	One end is connected to the mutually exclusive logical node and the other end is connected to the Phase computational basic block

Phase represents a basic block consisting of one or more serial activities, so each Phase node corresponds to a Phase view, which contains only a series of leaf nodes of the same type of activity, where the activity corresponds to the computational program of the leaf activity in the CIM as well as its input and output data. Table 5 shows a detailed description of the basic node nodes representing the activities in the Phase view.

表 1　PIM 建模语言 Phase 视图节点语义表

Element	Description
Basic Node	The computational activity contained under the Phase node corresponds to the leaf activity in the CIM and its input and output data

2.1 Basic node semantics

All basic node nodes represent the basic unit of activity execution in the PIM model that cannot be subdivided, and are responsible for describing the information related to the external activity invocation. They need to contain the execution program address, execution parameters, input data, optional input data, and output data information. The optional input data describes the scenario in which the data node points to the activity node through a special reference flow in the corresponding CIM model, indicating that it is used as the input data when the data file exists, and the optional input data is ignored if the file for the input data does not exist yet.

The state transfer graph of the basic node is shown in Figure 2, which contains three states: initial state, active state, and completed state.

The basic node, as the basic descriptive unit of the computational activity, is not represented in the PIM process view, but is visualized in the Phase view inside Phase.

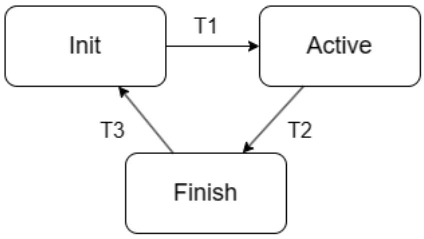


Figure 2 State Transfer Graph

The default state is the initial state.

- **T1** (*state transfer condition for initial state to active state*): the following two conditions need to be satisfied at the same time
  (1) The external Phase node to which the node belongs is in active state.
  (2) The current node is the first node in the Phase, or the state of the predecessor node is complete.
- **T2** (*state transfer condition for initial state to completion state*): the execution of the calling program of the current node is completed.
- **T3** (*state transfer condition for completion state to initial state*): the Phase node state external to the current node is reset to the initial state.

2.2 Phase semantics

Phase node has the same meaning in both tiled and nested modes, denoting the basic block divided in the PIM process model, which internally contains a series of serial computational activities of the same type. And inside the Phase node, except for the last computational activity, each computational activity has a unique successor activity of the same type as the next activity when the activity execution is completed. That is, for Phase node, there is a set of basic node $Node = \{node_1, node_2, \dots, node_n\}$, then the basic node nodes inside the activity will be executed sequentially according to the order of $node_1, node_2, \dots, node_n$.

Phase node as the basic unit of PIM process view of PIM model, all node nodes describing computational activities should be encapsulated by Phase node. Each Phase node corresponds to a Phase view of the PIM model and is used to visualize the basic node situation contained within the Phase.

Phase nodes should contain the activity type attribute. It supports both reductive and automatic activity options for describing the type of computational activity corresponding to the internal base node. When the activity type is a subsumption activity, there is one and only one basic node inside the Phase node, and the Phase node shall be connected to the end-of-parallelization marker, indicating the activity of subsuming the data generated by multiple instances of a parallel process two by two. When the Phase activity type is Automatic, the Phase itself only serves as a basic block describing the serial execution of the internal computational activity, with no other special meaning.

The state type of Phase is the same as that of node, with three states: initial, active, and completed, and the state transfer diagram is shown in Figure 2.

The default state is the initial state.

- **T1** (*state transfer condition for initial state to active state*): the following two conditions need to be satisfied simultaneously
  (1) For any input PIM stream of the Phase node, the start node of the PIM stream is in the completion state.

(2) For any input Phase node, the starting node of the PIM stream is in the completion state and the value of the conditional expression is true. If the starting node of the PIM stream is a loop control node, the result of the loop control node being in the completion state and index $\geqslant$ loopMax is true, where index is the loop count variable value of the loop control node and loopMax is the value of the loop count variable of the loop control node. loop count variable of the loop control node, and loopMax is the value of the maximum loop end variable of the loop control node.

(3) For any no stream input to the Phase, if the no stream start node is a conditional branch node, the conditional branch node must be in the completion state and the value of the conditional expression is false; if the no stream start node is a loop control node, the result of the loop control node being in the completion state and index $\geqslant$ loopMax is false, where index is the value of the loop count variable of the loop control node and loopMax is the value of the maximum loop end variable of the loop control node. where index is the value of the loop count variable of the loop control node and loopMax is the value of the maximum loop end variable of the loop control node.

- **T2** (*State transfer condition for initial state to completion state*): the following conditions need to be satisfied

(1) If the current Phase node activity type is automatic activity, all basic node nodes of the Phase node are in completion state.

(2) If the current Phase node is a reduce activity, the basic node nodes inside the Phase node are in the completion state, and the parallel instance described by the clone node connected to this phase via the end-of-parallel marker has been subsumed to a single instance.

- **T3** (*State Transfer Condition for Completion State to Initial State*): the loop in which the current Phase is located starts a new iteration. That is, the following scenario

(1) For the loop control node loop, there exists a PIM process fragment subSequence for the loop PIM fragment controlled by loop, the meaning of which is described in 4.2.2.7. When the loop node changes from the initial state to the completion state and the result of index ⩾ loopMax is false (where index is the value of the loop count variable of the loop control node loop, and loopMax is the value of the end-of-loop variable of the loop control node loop), the loop continues, and all the nodes within the subSequence are reset to the initial state. After that the noflow node is set to the active state and the loop is set to the initial state, starting a new loop iteration. If the current Phase node belongs to subSequence, the state is transferred to the initial state.。

2.3 Start / End semantics

A start node is a node that identifies the location of the start of a process. The start node has one and only one PIM stream output, and the end of the data stream is of type Phase node, which indicates the first basic block of the process that begins execution.

An end node is a node that identifies the end position of a PIM computation process. The end node has one and only one input PIM stream, and the current PIM computation process ends when the input PIM stream is available, i.e., when the state of the start node of the input PIM stream is the completion state.

The start and end nodes are non-redivisible basic nodes with two states: initial and end. Before the PIM process starts, the start and end node states are both initial. When the PIM process starts, the start node turns to the completion state and the process starts executing along the PIM stream output from the start node. When the start node state of the PIM stream input to the end node turns to completion state, the PIM process ends and the end node turns to completion state.

2.4 Loop control semantics

The loop control node is used to control the loop iteration of the tiling mode PIM process. The input direction of the loop control node supports the input of PIM streams, yes and no streams, and the output direction supports the output of only one yes stream and one no stream.

There exists a PIM process fragment $subSequence$, the start node of the fragment is of type Phase or logical node, denoted as $startPimNode$, and the end node is denoted as $endPimNode$. there exists a loop control node loop, and there exists a noflow, the start node of the noflow is loop, and the end node is denoted as $startPimNode$.

If $endPimNode$ is a Phase node, or a node of any node type, then there must be a PIM flow $flow_1$. The start of $flow_1$ is $endPimNode$, and the end of $endPimNode$ is loop; if $endPimNode$ is a loop control node, then there must be a flow $flow_1$, the start of $flow_1$ is loop, and the end of $endPimNode$ is loop; there must be a no flow $flow_1$, the start of noflow is loop, and the end of $startPimNode$. If the end node $endPimNode$ is a loop control node, then there should be a yes or no flow $flow_1$, and the start point of $flow_1$ is $endPimNode$ and the end point is loop; if it meets the above scenarios, then $subSequence$ is called loop PIM controlled by loop. loop PIM segment controlled by loop.

The loop control node contains three special attributes: loop count macro variable, loop start variable, and loop end variable. Remember that the value of loop count variable is index, the value of loop start variable is $loopBegin$, and the value of loop end variable is $loopMax$, then the output of loop control node is the execution condition of stream as $index \geq loopMax$, i.e., the yes-stream indicates the scenario of exiting the loop; and the output of loop control node is the execution condition of no-stream as $!\ (index \geq loopMax)$, i.e., the no stream indicates a scenario where the loop end condition is not met and iteration continues.

The loop control node contains two states: initial and completion. The default state is the initial state, and the inter-state transfer conditions T1 and T2 are described as follows:

- **T1** (*state transfer condition from initial state to completion state*):
  (1) For all input PIM stream connections of a loop control node, the starting node shall be in the completion state.
  (2) For all inputs to a loop control node that are streams, if the start node of that is stream is a conditional branch node, the conditional branch node must be in the completion state and the value of the conditional expression is true; if the start node of that is stream is a loop control node, the result of that is loop control node being in the completion state and $index \geq loopMax$ is true, where index is the loop control node's loop count variable of the loop control node.
  (3) For all input no streams of the loop control node, if the no stream start node is a conditional branch node, the conditional branch node must be in the completion state and the value of the conditional expression is false; if the yes stream start node is a loop control node, the result of the loop control node being in the completion state and $index \geq loopMax$ is false where index is the value of the loop count variable of the loop control node, loopMax is the value of the maximum loop end variable of the loop control node.
- **T2** (*completion state to initial state transfer condition*): start a new round of iteration
  (1) There exists a loop PIM fragment controlled by the current loop control node loop then subSequence. when the loop node is transferred from the initial state to the completion state and the result of $index \geq loopMax$ is false (where index is the value of the loop count variable of the loop control node loop and $loopMax$ is the value of the loop end variable of the loop control node loop), the loop continues, then all the nodes within the subSequence are reset to the initial state. end variable value of loop control node loop), the loop continues, then all nodes within the subSequence are reset to the initial state. After that, the noflow node is set to the active state, and the loop is set to the initial state.

2.5 Mutex semantics

A mutex node is used to control the mutex execution of a Phase and has a mutex id attribute. The mutex id attribute is consistent with the mutex node attribute in the CIM model and is a string expression type. Mutex node can be connected with several mutex streams, and the other end of the mutex streams is connected with Phase node, the Phase set connected by mutex node mutex through the mutex streams is $Phase = \{phase_1, phase_2, \dots, phase_n\}$, then all the basic blocks in the Phase set are connected to the Phase node. All the basic node computational activities in the basic block of Phase need to compete for the mutex lock represented by the mutex before execution. Mutually exclusive nodes are stateless nodes, no need to consider the state transfer condition.

2.6 Clone semantics

The clone node represents a description of the single-procedure multiple data stream SPMD parallel mode. The clone node contains a parallel start marker and a parallel end marker to mark the parallel PIM process segments. The clone node has several attributes such as input batch data, data mapping method, number of parallel instances variable, and task order number variable. The batch input data refers to the data nodes that need to be used in the parallel PIM fragment, where the data node type is the data node in the CIM process, and its path is a folder, which represents the set of all data files inside the folder. The data mapping method is of enumeration type, with two options of equalized score and full data; the number of task instances variable is of integer variable type; and the task serial number is of macro variable type.

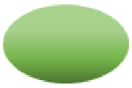
If the value of the number of instances variable is n, the engine will create n instances of PIM parallel segments for parallel execution, which are independent of each other and have independent node states. The task serial number variable is used for each instance to obtain the serial number of the current instance among all instances, and the values that n instances can obtain through this variable are $1,2,\dots,n$. The data mapping method is used to realize the same single program with multiple data streams in parallel, and when the data mapping method is equally divided, assume that there are a total of m files under the path of the input batch of data of the current parallel fragment, and as $F = \{f_1, f_2, \dots, f_m\}$. Then the engine will divide the m data files equally into n parts $\{f_1, f_2, \dots, f_{\lceil m/n \rceil}\}, \{f_{\lceil m/n \rceil+1}, f_{\lceil m/n \rceil+2}, \dots, f_{2\lceil m/n \rceil}\}, \dots, \{f_{(n-1)*\lceil m/n \rceil+1}, f_{(n-1)*\lceil m/n \rceil+2}, \dots, f_m\}$ of which each fragment is able to obtain an independent copy of the input data for independent computational processing. When the data mapping method is full data, all n parallel PIM fragment instances are able to access all m input data files, and each parallel PIM fragment instance is required to reasonably select the input data for processing according to the current task instance order number, in order to realize a more complex data mapping method.

The cloned nodes are stateless nodes and there is no need to consider the state and transfer conditions of the cloned nodes.

3. Graphical Modeling Definition

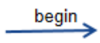
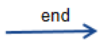
Icons of the nodes in the PIM view is shown in Table 5.

Table 5 PIM modeling elements

Element	Icon	Element	Icon
Phase		Loop Control	
Basic Node		Mutex	
Start		Clone	
End			

Icons of the connections in the PIM view is shown in Table 6.

Table 6 PIM modeling connections

Connection	Icon	Connection	Icon
Sequence Flow		Clone Start	
No Flow		Clone End	
Yes Flow		Mutex Flow	