Flow Subsystem

Here we are concerned with the flow of data within an Activity.

Relationship numbering range: R800-R850

Class Descriptions

Activity

An Activity is a set of Actions and Flows, which define some sort of computation. An Activity may be associated with a State, Method, Domain Operation or External Entity Operation.

Attributes

ID

A unique value

Type: Activity ID, based on the Nominal type

Identifiers

1. **ID** + **Domain**

Control Flow

An Action executes only when all of its Flows are available. In the case of a Data Flow, this means that data in the flow is available. A Control Flow makes it possible to enable the activation of an Action without supplying any data. This is helpful, for example, when you want to sequence two Actions with no data dependency.

The control dependency between a decision Action and a downstream Action triggered based on the result of the decision is also represented by a Control Flow. No data is communicated along the Flow other than whether or not the Control Flow activates.

In classic data flow diagrams outgoing signals are often depicted with dashed arrows suggesting that they might be control flows. In our metamodel, however, they are not since those dashed arrows don't activate any Action internal to the Activity.

Attributes

ID

A unique value

Type: Selection ID, based on the Nominal type

Identifiers

1. **ID** + **Activity** + **Domain**

Data Flow

This type of Flow indicates that some kind of data is communicated to or between Actions within a single Activity. Data flowing from an action that writes a Class's attribute value can certainly be thought of as flowing data, but is not, in fact represented by a Data Flow. Instead, the write action will simply specify its target class.

Attributes

ID

A unique value

Type: Selection ID, based on the Nominal type

Identifiers

1. **ID** + **Activity** + **Domain**

Flow

A Flow is a data or control dependency on or between Actions in a single Activity. An Action may execute only when each of its input Flows is active. In the case of a Data Flow, this means that data is available. In the case of a Control Flow, some condition has been met, but no data is transferred.

Any data entering an Activity or exchanged between Actions is represented as a Flow. It's easiest to just imagine arrows entering or connecting processes within an Activity's data flow diagram.

Keep in mind, though, that this action metamodel does not model any particular visualization syntax, written or drawn. There are several cases, for example, where a drawn arrow on a data flow diagram does not correspond to an instance of Flow in the metamodel.

Attributes

ID

A unique value

Type: Selection ID, based on the Nominal type

Identifiers

1. **ID** + **Activity** + **Domain**

Instance Flow

An Instance Flow communicates zero or more instance references. An instance reference is simply a relation whose header corresponds to the primary identifier of a Class. The body of this relation will then supply zero or more tuples where each tuple defines an instance of that Class.

The choice of the primary Identifier is an arbitrary convention since any identifier would do the job. There is nothing special about being 'primary' other than it provides us with an obvious choice. The only purpose of the data in any given Instance Flow tuple is to select a unique instance of the Class.

Attributes

No non-referential attributes

Identifiers

1. **ID** + **Activity** + **Domain**

Multiple Instance Flow

This is an Instance Flow whose cardinality may be any positive integer. Any Action taking such a flow as input then expects zero, one or many instances.

Attributes

No non-referential attributes

Identifiers

1. **ID** + **Activity** + **Domain**

Scalar Flow

From a relational perspective, the internal structure of a scalar value is opaque. It can only be accessed by user defined type specific operations. An example Scalar Flow might communicate an integer value or a GPS location or a video image or any other arbitrarily complex (or simple) unit of data.

Attributes

No non-referential attributes

Identifiers

1. **ID** + **Activity** + **Domain**

Single Instance Flow

This is an Instance Flow whose cardinality may be one or zero. Any Action taking such a flow as input then expects zero or one instance.

Attributes

No non-referential attributes

Identifiers

1. **ID** + **Activity** + **Domain**

Table Attribute

Each column of a Table Flow is defined by an attribute:type pair. A Table Attribute is essentially the name of the column.

Attributes

Name

A name that describes the meaning of a table column.

Type: Attribute Name based on String

Identifiers

1. **Name** + **Table** + **Activity** + **Domain**

Table Flow

A Table Flow is defined as a relational variable or, 'relvar'. This is a variable that takes a relation as its value.

A relation is defined as having a header with zero or more attribute:type pairs and a body consisting of zero or more tuples such that each tuple supplies a value corresponding to each attribute:type pair in the header. A relation is often visualized as a table with a relvar visualized as just the table header.

Note that an Instance Flow is really just a special type of Table Flow where the relvar must be defined by a primary Identifier.

Unlike an Instance Flow, a Table Flow relvar is typed by any combination of Table Attributes and Types. A Table Attribute may or may not correspond to any given class Attribute.

Attributes

No non-referential attributes

Identifiers

1. **ID** + **Activity** + **Domain**

Relationship Descriptions

R800 / 1:Mc

Activity communicates data via *zero, one or many* **Flow**

Flow is path of data communicated within *one* **Activity**

An Activity might not specify any Actions at all and be associated with a model element (state, method, etc) with no input parameters. In such a case, neither Actions or Flows are present.

For a non-empty Activity, there are typically one or more Flows providing data to and between various Actions.

A Flow is specified in the context of an Activity. All flowing data terminates in the class model or is packaged up for transmission in a signal or s

Formalization

Flow.(Activity, Domain) -> Activity.(ID, Domain)

R801 / Generalization

Flow is a **Data** or **Control Flow**

The only two kinds of information exchanged among Actions is data and control.

Data can be arbitrarily complex and each Data Flow will be constrained by some sort of data type such as a class or a table definition.

Control carries no actual data, but it does provide input to an Action to indicate that it may or may not execute.

Formalization

<subclass>.(ID, Activity, Domain) -> Flow.(ID, Activity, Domain)

R802 / Generalization

Data Flow is an **Instance**, **Table** or **Scalar Flow**

In relational theory we have two basic types of data, relations and scalars. For our object-oriented analysis purposes, we split the relation type into relations that describe only instances, and relations that do not necessarily describe instances. This then gives us our three flows.

Any data not described as a relation such as an integer or a video image, for example, is considered to be a scalar value which may be transmitted in a Scalar Flow.

Formalization

<subclass>.(ID, Activity, Domain) -> Data Flow.(ID, Activity, Domain)

R803 / 1:Mc

Instance Flow flows instances of *one* **Class**

Class defines type of *zero, one or many* **Instance Flow**

All instance references communicated in an Instance Flow must belong to the same Class. We can say that the Class, and more specifically, the primary Identifier of that Class establishes the type of data that can be transmitted in the Instance Flow.

The same Class may define many Instance Flows, both in the same Activity and across multiple Activities.

Formalization

Instance Flow.(Class, Domain) -> Class.(Name, Domain)

R804 / 1:Mc

Table Flow column is defined by *zero, one or many* **Table Attribute**

Table Attribute defines column of *one* **Table Flow**

A Table Flow defines its content as set of attribute:type pairs which may be the empty set.

Each Table Attribute is defined specifically for one Table Flow.

Formalization

Table Attribute.(Table, Activity, Domain) -> Table Flow.(ID, Activity, Domain)

R806 / 1:Mc

Type constrains the value of *zero, one or many* **Table Attribute**

Table Attribute is constrained by *one* **Type**

Formalization

Table Attribute.Type -> Type.Name

R807 / 1:Mc

Scalar Flow is defined by *one* **Type**

Type defines *zero, one or more* **Scalar FLow**

The data communicated in a Scalar Flow is constrained to the set of values defined by a single Type.

A Type can be referenced in many places such as on a Class's Attribute. So it is certainly reasonable to define a Type which is not currently used on any Scalar Flow.

Any number of Scalar Flows may be constrained by the same Type.

Formalization

Scalar Flow.Type -> Type.Name

R808 / Generalization

Instance Flow is a **Single** or **Multiple Instance Flow**

Certain Actions require that, at most, a single instance reference be supplied. It is therefore helpful to separate out such an Instance Flow.

An Action that performs a selection using an Identifier as input produces an output flow that may not contain more than one instance reference. The definition of a Single Instance Flow associated with such an output reinforces this constraint.

Formalization

<subclass>.(ID, Activity, Domain) -> Instance Flow.(ID, Activity, Domain)

