



# TrueChange™ under the hood

---

How we check the consistency of large models (almost) instantly

Hugo Lourenço, Rui Eugénio



# The OutSystems Platform

# The OutSystems Platform



# The IDE - Service Studio

The screenshot shows the OutSystems Service Studio IDE interface. The main canvas displays a mobile app design for a screen titled "My Details". The design includes a header, a text input field, a "Job Title" label, and a "SAVE CHANGES" button. A yellow tooltip is overlaid on the input field, displaying a warning message: "Variable GetContactsByld...Contact.Name 'Max. Length' of Input\_Name (50) and Contact.Name 'Length' (250) values should match. Consider changing 'Max. Length' property." The right-hand side of the IDE features a "Widget Tree" showing a hierarchy of components: Contacts, Entity Diagrams, ContactsDataModel, Entities, Database, Contact, Id, Name, JobTitle, and Phone. Below the widget tree, the "Name" entity attribute is detailed with a table:

Name	Name
Description	...
Label	Name
Data Type	Text
Length	250
Is Mandatory	Yes
Default Value	...

The bottom status bar shows a warning icon and the text: "1 Warning | Debugger | 1-Click Publish". A detailed warning message is displayed: "Coherence Suggestion | Warning found | 'Max. Length' of Input\_Name (50) and Contact.Name 'Length' (250) values should match. Consider changing 'Max. Leng...? | The module is valid, but 1 warning was found in this module. Double-click the above message to highlight the issu...". The bottom right corner of the IDE shows "Contacts uploaded at 20:40 | admin | localhost".

# The IDE - Service Studio

The screenshot displays the OutSystems Service Studio IDE interface. The main workspace shows a workflow diagram with the following steps: Start, SendEmail, and End. The SendEmail action is highlighted, and a tooltip displays its configuration:

**Action**  
SendEmail

**Arguments**  
EmailAddress (Email): Value not set

A valid expression must be set for parameter 'EmailAddress'.

The right-hand pane shows the project structure for 'Contacts', including folders for Client Actions, Server Actions, and Integrations. The 'EmailAddress' input parameter is selected, and its properties are shown in the bottom right:

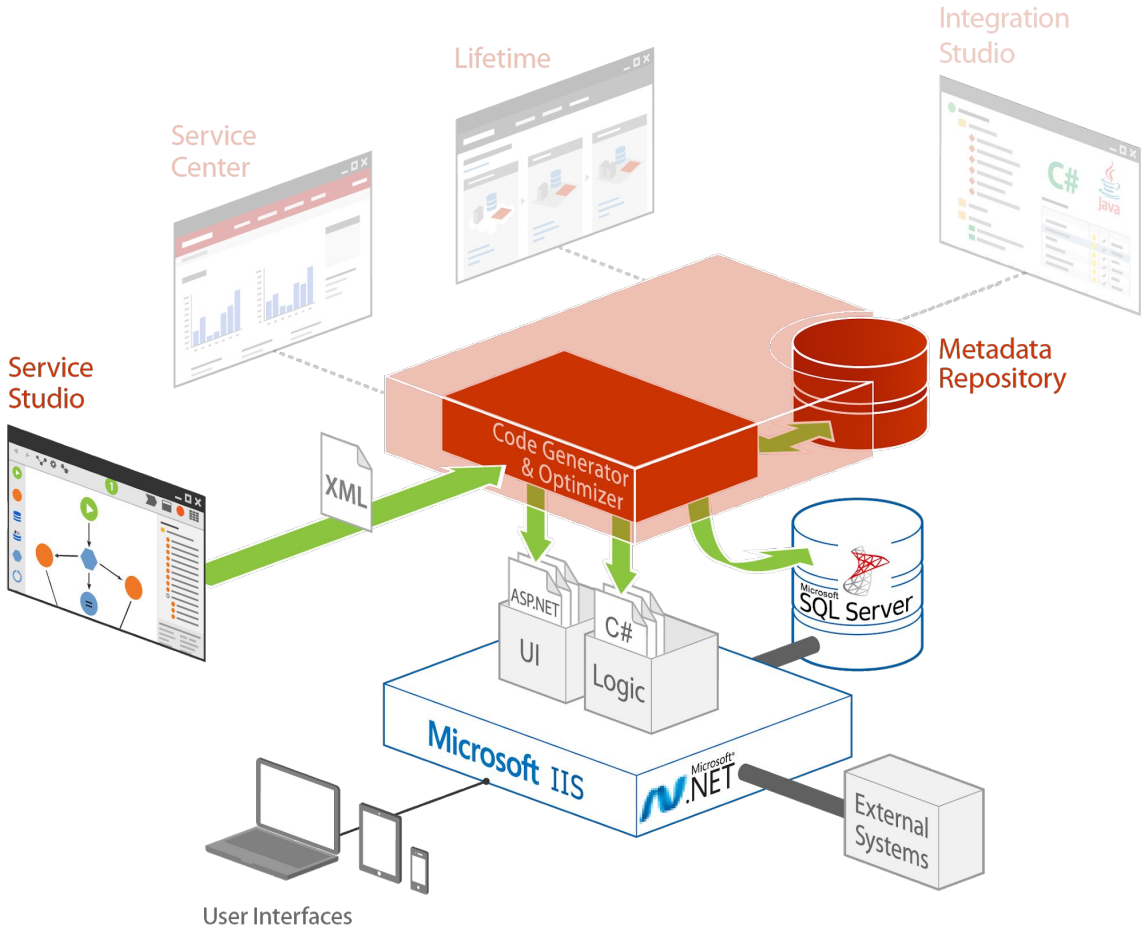
Name	EmailAddress
Description	...
Data Type	Email
Is Mandatory	Yes
Default Value	

The bottom status bar indicates '1 Error, 1 Warning' and provides details for the error:

- Required Property Value**: A valid expression must be set for parameter 'EmailAddress'.
- Coherence Suggestion**: 'Max. Length' of Input\_Name (50) and Contact.Name 'Length' (250) values should match. Consider changing 'Max. Len...
- Error found**: The module has 1 error and 1 warning. Double-click the above messages to highlight the issue or right-click for...

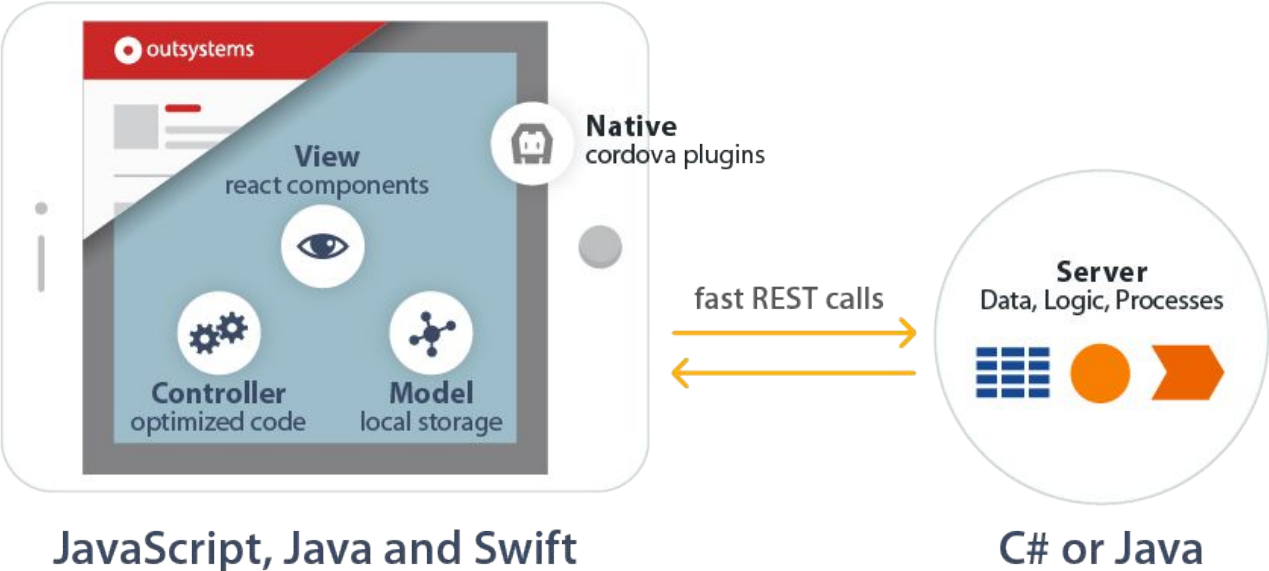
The interface includes a top menu bar (Module, Edit, View, Debugger, Help), a toolbar with navigation and execution icons, and a search bar (Search... (Ctrl+E)). The bottom status bar shows 'TrueChange™' and 'Contacts.omi saved at 21:02 | admin | localhost'.

# The OutSystems Platform



# The OutSystems Platform

Cross-Platform, Standards-Based



A long-exposure photograph of a city street at night. The image shows several tall skyscrapers with lit windows, creating a grid-like pattern of light. In the foreground, there are bright, horizontal light trails from traffic, including a prominent yellow and white trail from a vehicle. The overall scene is dynamic and urban, with a mix of warm and cool tones from the city lights.

# The OutSystems Language



# Actions

The screenshot displays the OutSystems IDE interface for a project named 'Contacts'. The main workspace shows a workflow diagram with the following steps: Start, SendEmail, and End. The SendEmail action is highlighted, and a tooltip provides the following details:

**Action**  
SendEmail

**Arguments**  
EmailAddress (Email): Value not set

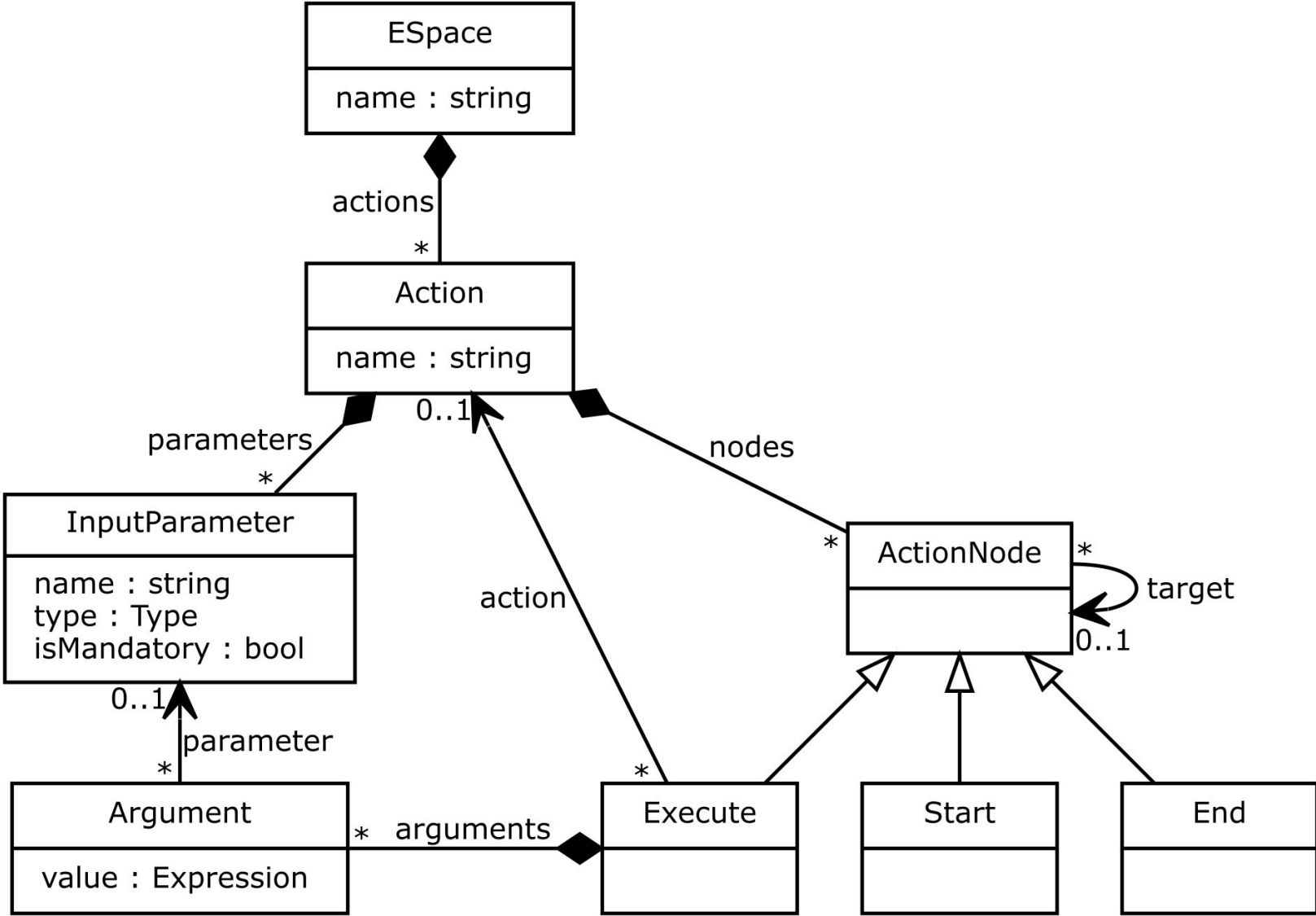
A valid expression must be set for parameter 'EmailAddress'.

The right sidebar shows the project structure, with 'EmailAddress' selected as an input parameter. The details for 'EmailAddress' are as follows:

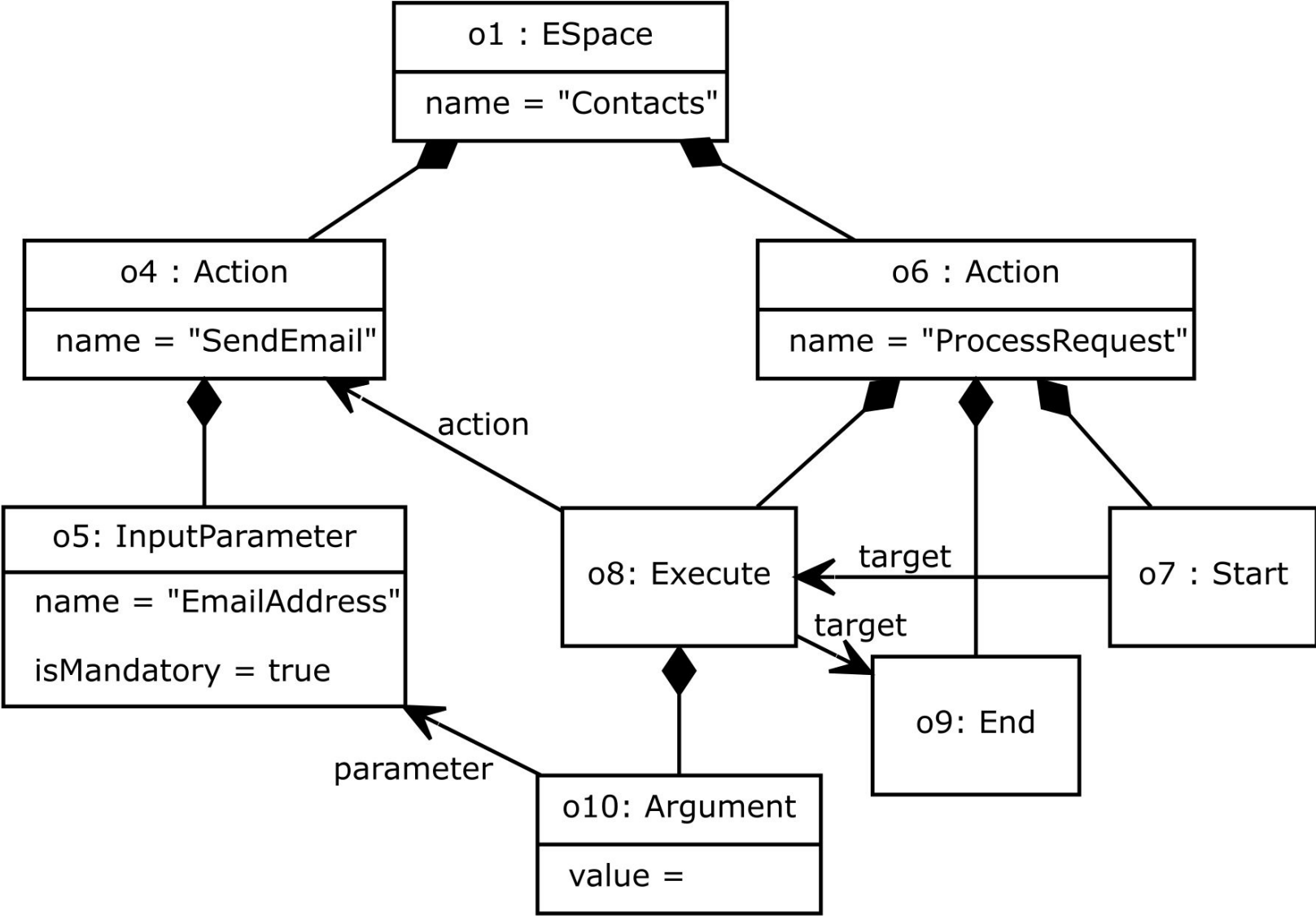
Name	EmailAddress
Description	...
Data Type	Email
Is Mandatory	Yes
Default Value	
Created by admin	
Last modified by admin at 21:00	

The bottom status bar indicates '1 Error, 1 Warning' and the TrueChange logo. The error message is: 'Required Property Value: A valid expression must be set for parameter 'EmailAddress''. A coherence suggestion is also visible: 'Coherence Suggestion: 'Max. Length' of Input\_Name (50) and Contact.Name 'Length' (250) values should match. Consider changing 'Max. Len...'. An error found message states: 'Error found: The module has 1 error and 1 warning. Double-click the above messages to highlight the issue or right-click for...'.

# Actions metamodel



# A sample model



# Metamodel definition

<MetaModel>

```
<Class name="ESpace">  
  <Property name="Name" type="Text" />  
  <Child name="Actions" type="Action" />  
</Class>
```

```
<Class name="Action">  
  <Property name="Name" type="Text" />  
  <Child name="InputParameters" type="InputParameter" />  
  <Child name="Nodes" type="ActionNode" />  
</Class>
```

```
<Class name="InputParameter">  
  <Property name="Name" type="Text" />  
  <Property name="Type" type="Type" />  
  <Property name="IsMandatory" type="Bool" />  
</Class>
```

...

# Metamodel definition

...

```
<Class name="ActionNode">  
  <Property name="Target" type="ActionNode" />  
</Class>
```

```
<Class name="Start" base="ActionNode" />  
<Class name="End" base="ActionNode" />
```

```
<Class name="Execute" base="ActionNode">  
  <Property name="Action" type="Action" />  
  <Child name="Arguments" type="Argument" />  
</Class>
```

```
<Class name="Argument">  
  <Property name="Parameter" type="InputParameter" />  
  <Property name="Value" type="Expression" />  
</Class>  
</MetaModel>
```

# Generated classes

```
partial class ESpace : ModelObject {  
    public string Name { get; set; }  
    public List<Action> Actions { get; };  
}
```

```
partial class Action : ModelObject {  
    public string Name { get; set; }  
    public List<InputParameter> InputParameters { get; };  
    public List<ActionNode> Nodes { get; };  
}
```

```
partial class InputParameter : ModelObject {  
    public string Name { get; set; }  
    public Type Type { get; set; }  
    public bool IsMandatory { get; set; }  
}
```

```
partial class ActionNode : ModelObject {  
    public ActionNode Target { get; set; }  
}
```

```
...
```



# Consistency Check

# Consistency check

The screenshot displays the OutSystems Studio interface for a module named 'Contacts'. The central workspace shows a flowchart with a 'Start' node, a 'SendEmail' action, and an 'End' node. A tooltip for the 'SendEmail' action indicates an error: 'A valid expression must be set for parameter 'EmailAddress''. The right-hand pane shows the project structure, with 'EmailAddress' selected under the 'SendEmail' action. Below the workspace, a 'Debugger' window is open, displaying a list of messages:

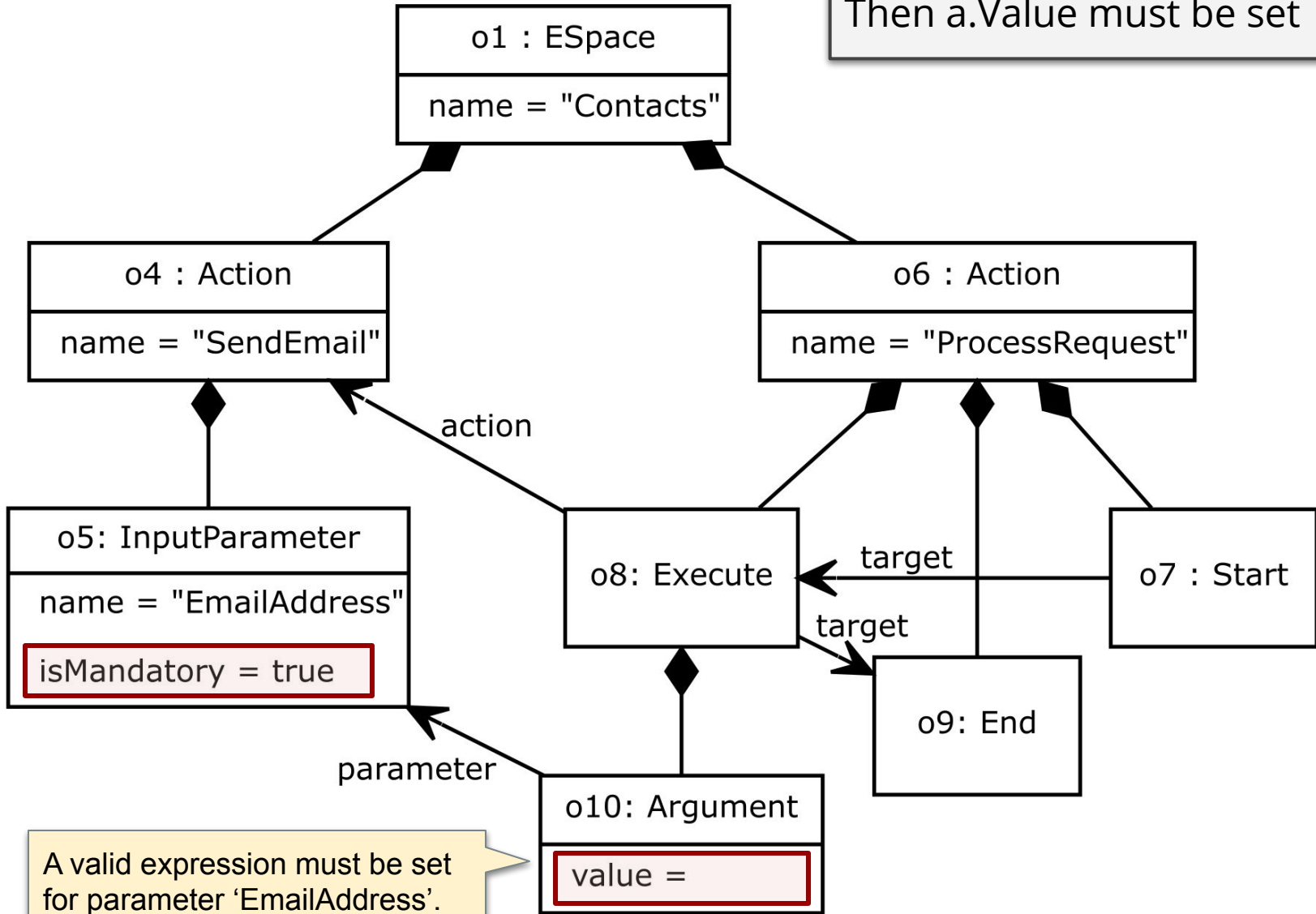
- 1 Error, 1 Warning
- Required Property Value: A valid expression must be set for parameter 'EmailAddress'.
- Coherence Suggestion: 'Max. Length' of Input\_Name (50) and Contact.Name 'Length' (250) values should match. Consider changing 'Max. Len...
- Error found: The module has 1 error and 1 warning. Double-click the above messages to highlight the issue or right-click for...

The 'Debugger' window also shows a 'TrueChange™' watermark and a status bar at the bottom indicating 'Contacts oml saved at 21:02 | admin | localhost'.



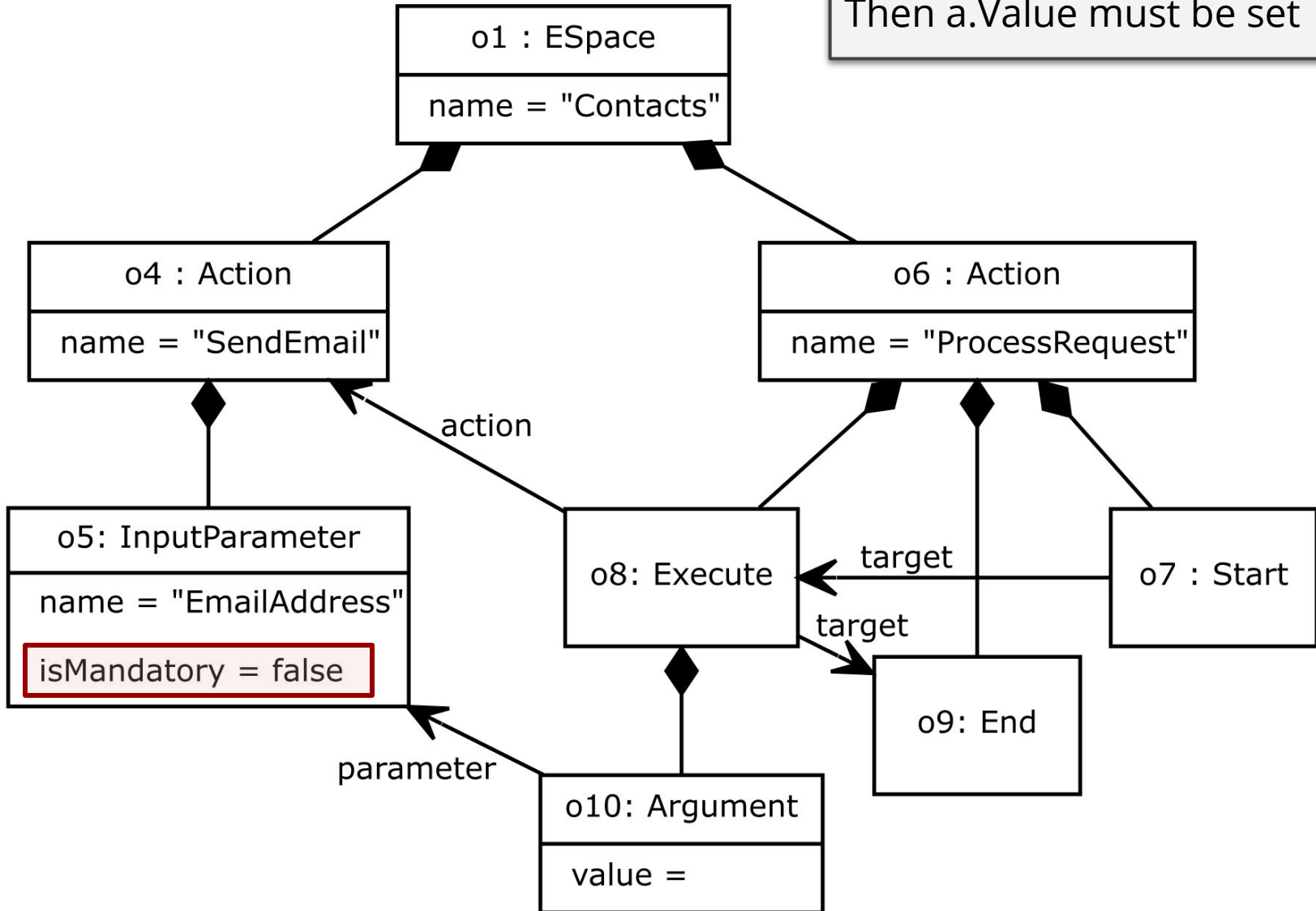
# Consistency check

For a : Argument  
If a.Parameter.IsMandatory  
Then a.Value must be set

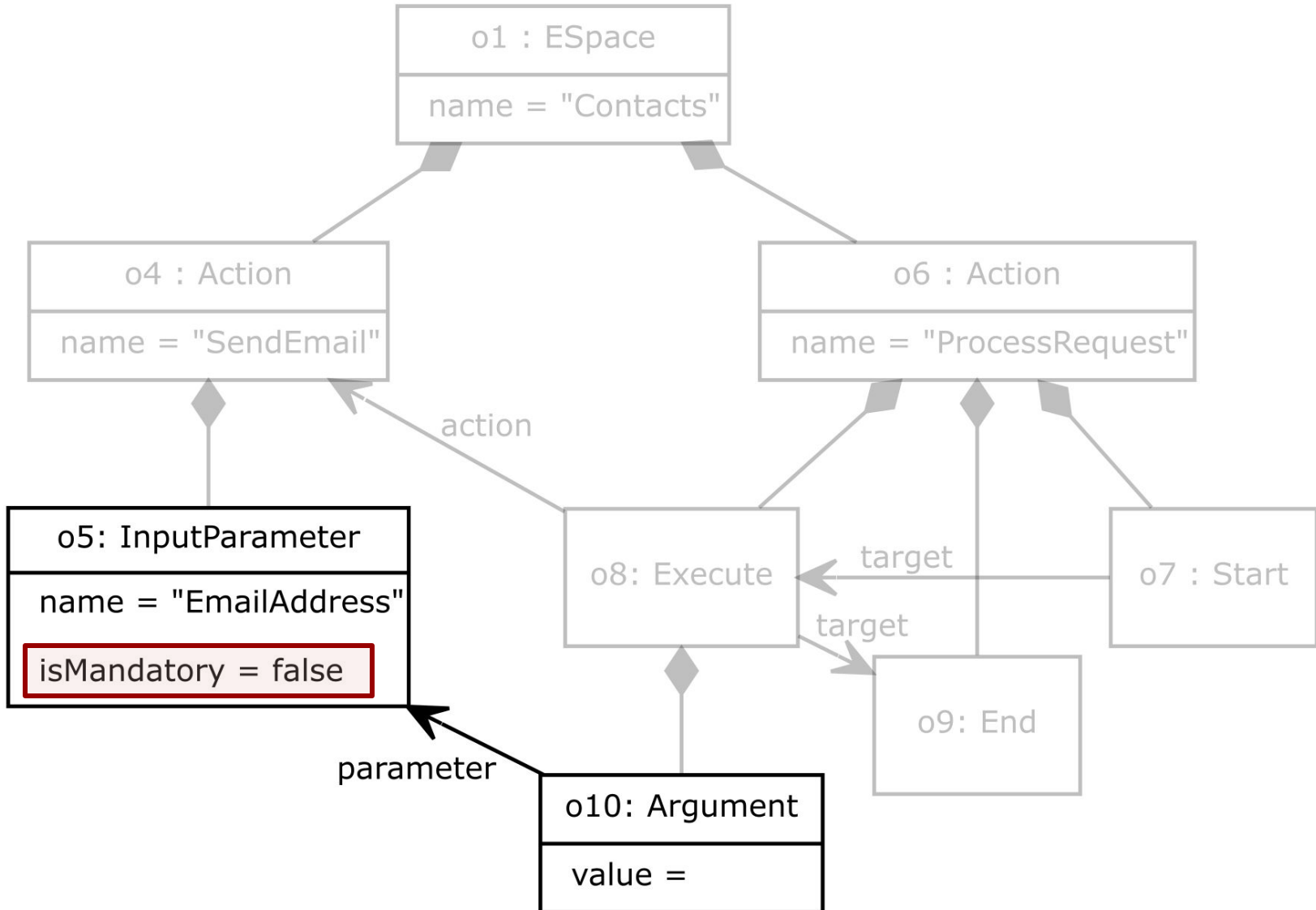


# Consistency check

For a : Argument  
If a.Parameter.IsMandatory  
Then a.Value must be set



# Incremental consistency check



# Incremental consistency check

**If** we know the objects whose consistency check depends on `InputParameter.IsMandatory`

**then** when `InputParameter.IsMandatory` changes we only need to re-check those objects

Intuitively, those are objects that *refer* to the input parameter: its “referers” (or back-references)

TrueChange relies on:

- “Verify” caches
- Automatically kept back-references
- Information about consistency check dependencies

# Verify Caches

# Verify cache base code

```
enum VerifyMessageKind { Warning, Error }
```

```
class VerifyMessage { public VerifyMessageKind Kind; public string Text; }
```

```
partial class ModelObject {
```

```
    VerifyMessage[] verifyMessages;  
    protected bool isVerified = false;
```

```
    public IEnumerable<VerifyMessage> Verify() {  
        if (!isVerified) {  
            verifyMessages = CalculateVerifyMessages().ToArray();  
            isVerified = true;  
        }  
        return verifyMessages;  
    }
```

```
    protected virtual IEnumerable<VerifyMessage> CalculateVerifyMessages() {  
        yield break;  
    }
```

```
}
```

# Verify cache invalidation code (generated)

```
partial class InputParameter : ModelObject {  
    private string _name;  
  
    public string Name {  
        get => _name;  
        set {  
            if (_name != value) {  
                _name = value;  
                isVerified = false;  
            }  
        }  
    }  
}
```

# Verify rules (hand-coded)

```
partial class Argument {  
  
    protected override IEnumerable<VerifyMessage> CalculateVerifyMessages() {  
        foreach (var msg in base.CalculateVerifyMessages()) {  
            yield return msg;  
        }  
  
        if (Parameter.IsMandatory && Value == null){  
            yield return new VerifyMessage() {  
                Kind = VerifyMessageKind.Error,  
                Text = $"A valid expression must be set for parameter '{Parameter.Name}'."  
            };  
        }  
    }  
}
```



# **Back References**

# Back-references base code

```
partial class ModelObject {
```

```
List<ModelObject> referers = new List<ModelObject>();
```

```
// called when a property is being changed from "oldValue" to "newValue"
```

```
protected void UpdateReferers(ModelObject oldValue, ModelObject newValue) {
```

```
    if (oldValue != null) {  
        oldValue.referers.Remove(this);
```

```
    }
```

```
    if (newValue != null) {  
        newValue.referers.Add(this);
```

```
    }
```

```
}
```

```
protected void InvalidateReferers() {
```

```
    foreach (var obj in referers) {  
        obj.isVerified = false;
```

```
    }
```

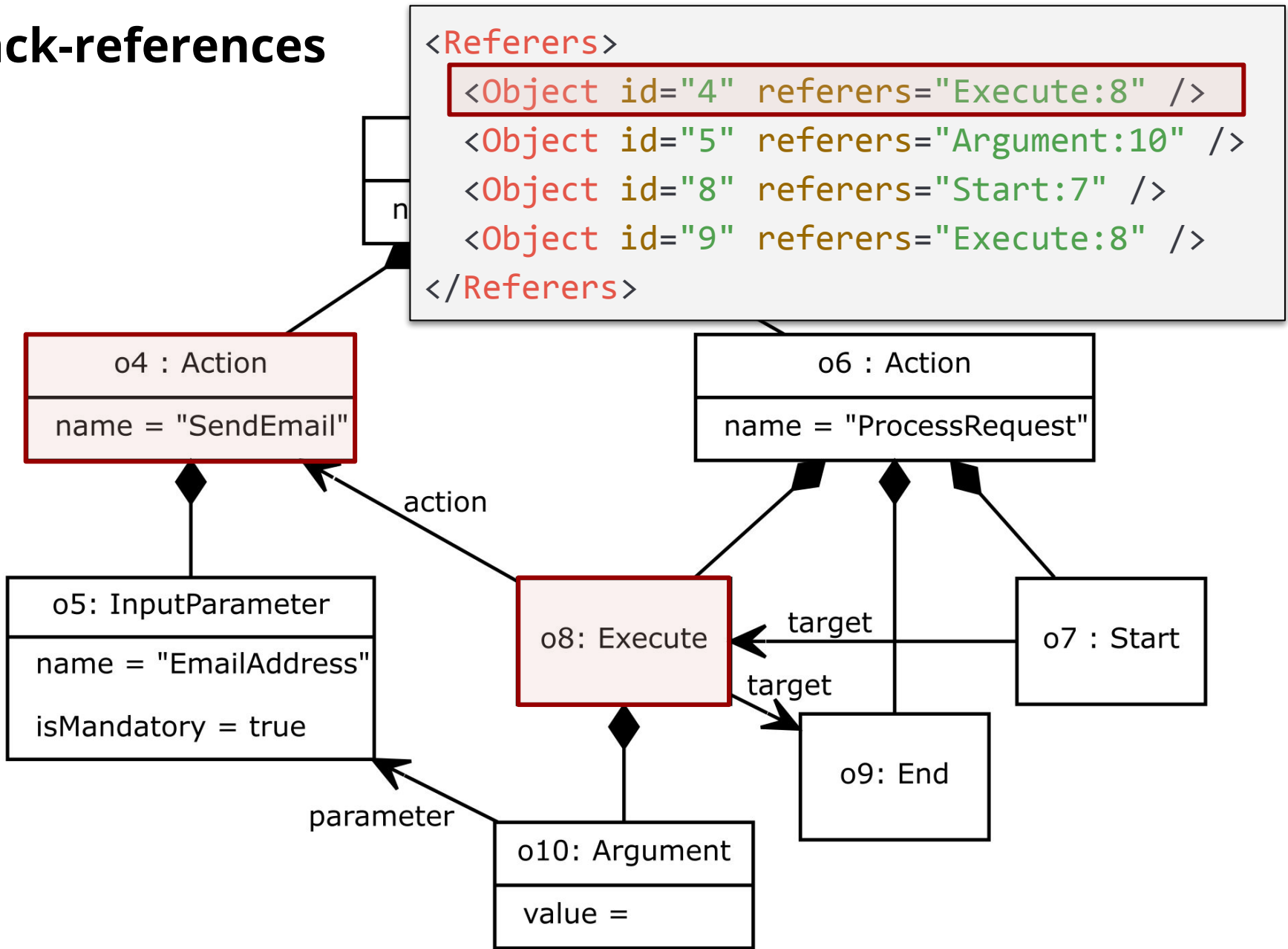
```
}
```

```
}
```

# Back-references tracking code (generated)

```
partial class Argument : ModelObject {  
    private InputParameter _parameter;  
  
    public InputParameter Parameter {  
        get => _parameter;  
        set {  
            if (_parameter != value) {  
                UpdateReferers(_parameter, value);  
                _parameter = value;  
                isVerified = false;  
            }  
        }  
    }  
}
```

# Back-references



# **Verify Dependencies**

# Verify dependencies - metamodel

```
...  
<Class name="ActionNode">  
  <Property name="Target" type="ActionNode" />  
</Class>  
  
<Class name="Start" base="ActionNode" />  
<Class name="End" base="ActionNode" />  
  
<Class name="Execute" base="ActionNode">  
  <Property name="Action" type="Action" />  
  <Child name="Arguments" type="Argument" />  
</Class>  
  
<Class name="Argument" verifyDependencies="Parameter.IsMandatory">  
  <Property name="Parameter" type="InputParameter" />  
  <Property name="Value" type="Expression" isOptional="true" />  
</Class>  
</MetaModel>
```

# Verify dependencies - generated concrete classes

```
partial class InputParameter : ModelObject {  
  private bool _isMandatory;  
  public bool IsMandatory {  
    get => _isMandatory;  
    set {  
      if (_isMandatory != value) {  
        InvalidateReferers();  
        _isMandatory = value;  
        isVerified = false;  
      }  
    }  
  }  
}
```



# Evaluation



# Time

Model	Number of elements	Full check time (ms)	Incremental check		
			Change kind	Time (ms)	% of full check time
Model #1	259759	11468	Add input parameter to screen	70	0.6%
			Change input parameter type	148	1.3%
			Change input parameter to mandatory	54	0.5%
...					
ServiceCenter	278187	26388	Add attribute to entity	215	0.8%
			Change attribute type	388	1.5%
			Add input parameter to action	91	0.3%
			Change input parameter type	77	0.3%
			Change input parameter to mandatory	62	0.2%
<b>Average</b>				<b>94</b>	<b>0.8%</b>

# Space

Model	Compressed model size (MB)	Compressed referers size (MB)	Referers size / model size
Model #1	12.0	4.2	35%
Model #2	7.4	2.8	37%
Model #3	8.3	3.0	36%
Model #4	7.3	2.7	37%
Service Center	16.4	4.5	28%
<b>Average</b>			<b>35%</b>



# Conclusion

# Conclusion

## Back-references

- Simple technique
- Easy to implement via a code generator
- Not limited to accelerating consistency checking!

## Verify dependencies

- The Achilles heel of our approach...
- Not very expressive
- Easy for developers to forget to declare them
- We'll take a look into alternatives (OCL, Alloy, ...)



**Thank You!**

[hugo.lourenco@outsystems.com](mailto:hugo.lourenco@outsystems.com)

