Cost-effective development of flexible self-* applications

Radu Calinescu

Computing Laboratory
University of Oxford

Outline

- Motivation
- Generic self-* framework
- Self-* application development

Motivation

word processing spreadsheets email

✗ cost-effective

✗ flexible

no general-purpose architecture & generic development approaches for self-* applications

✓ cost-effective

✓ flexible

Framework objectives

Significant reduction in development effort/expertise/cost

- component reuse
- off-the-shelf technologies & tools

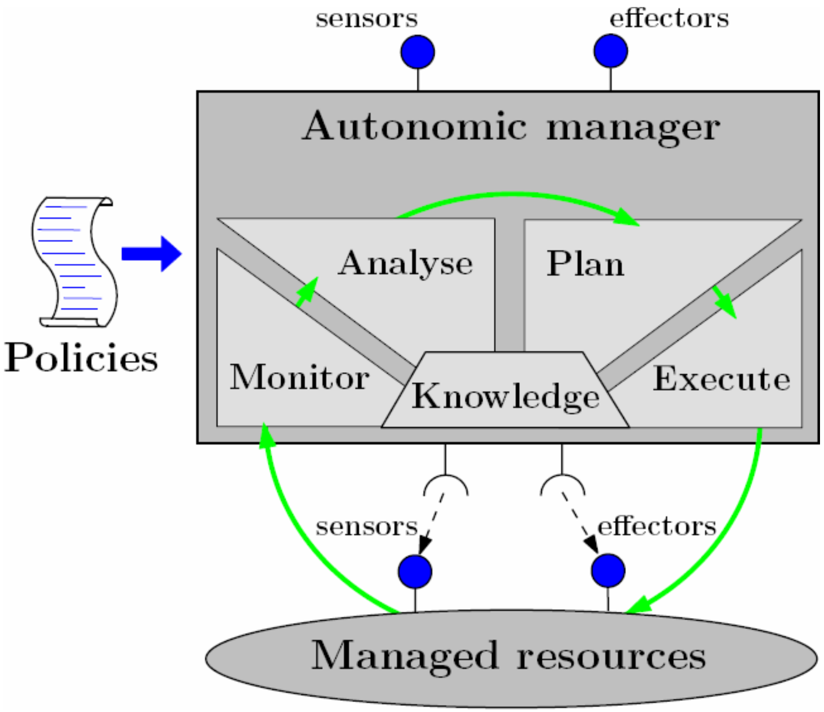
Major improvement in application flexibility/robustness

- model-based development
- automated code generation

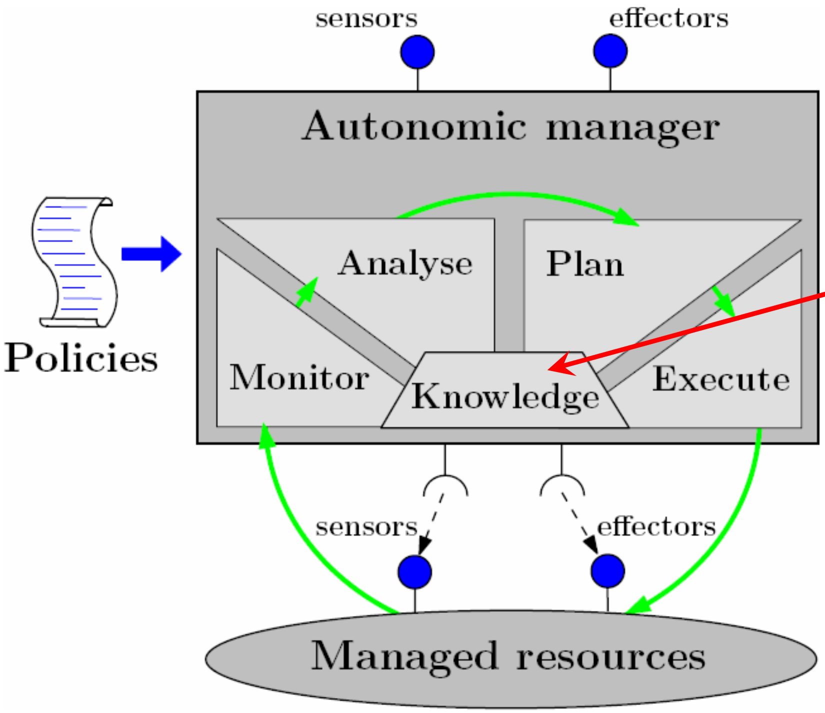
New powerful capabilities

- new classes of policies
- support for system-of-systems development

Typical drawbacks of self-* architectures

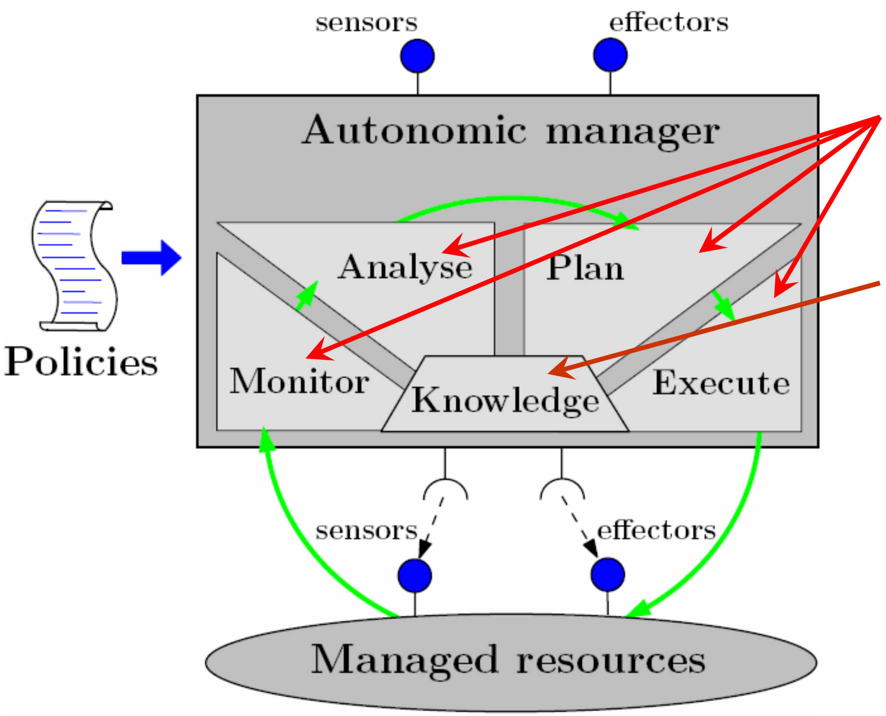


Typical drawbacks of self-* architectures



application-specific metadata hard-coded within the autonomic manager

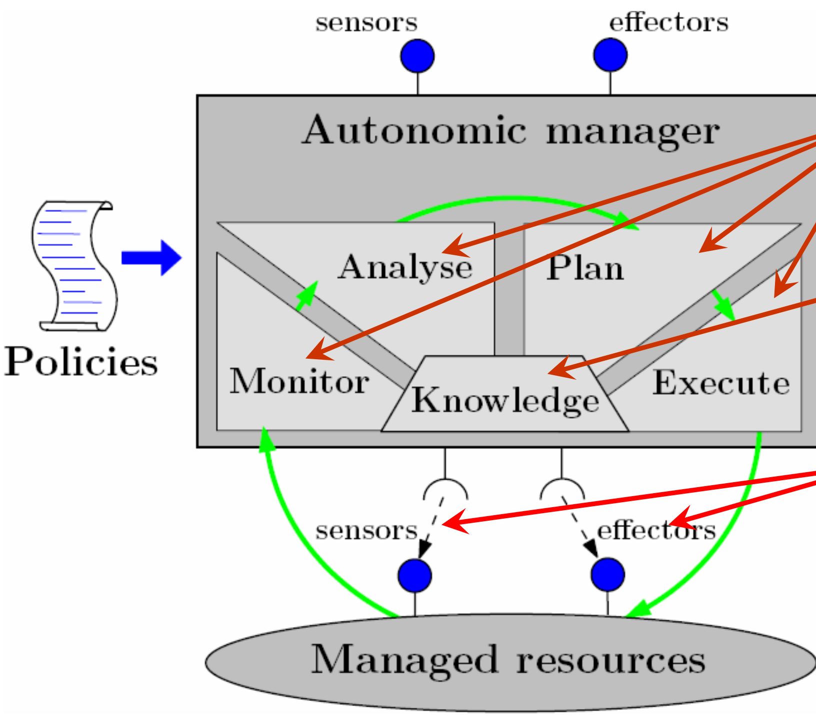
Typical drawbacks of self-* architectures



resource-specific code

application-specific metadata hard-coded within the autonomic manager

Typical drawbacks of self-* architectures

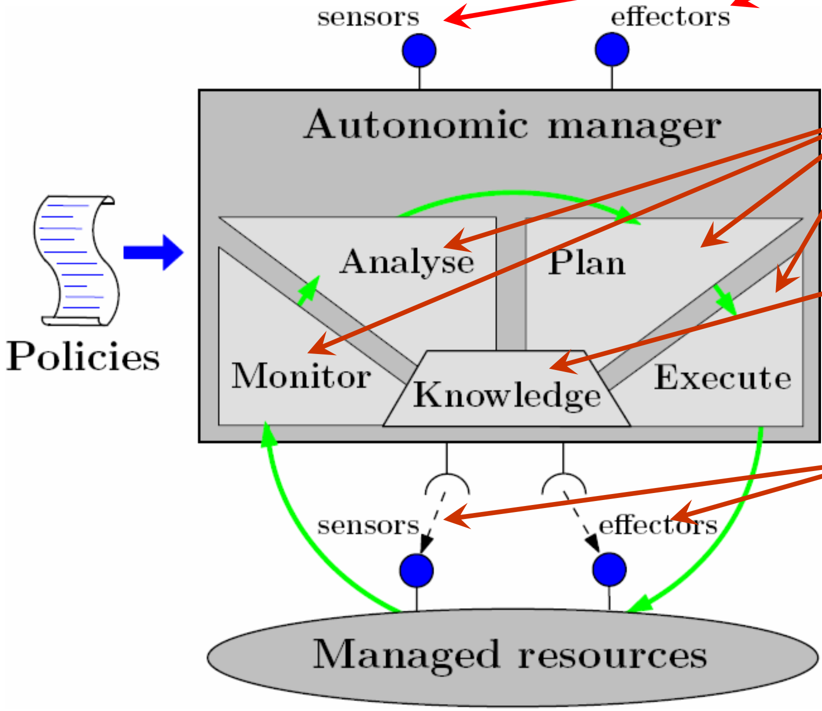


resource-specific code

application-specific metadata hard-coded within the autonomic manager

manually-implemented interfaces

Typical drawbacks of self-* architectures



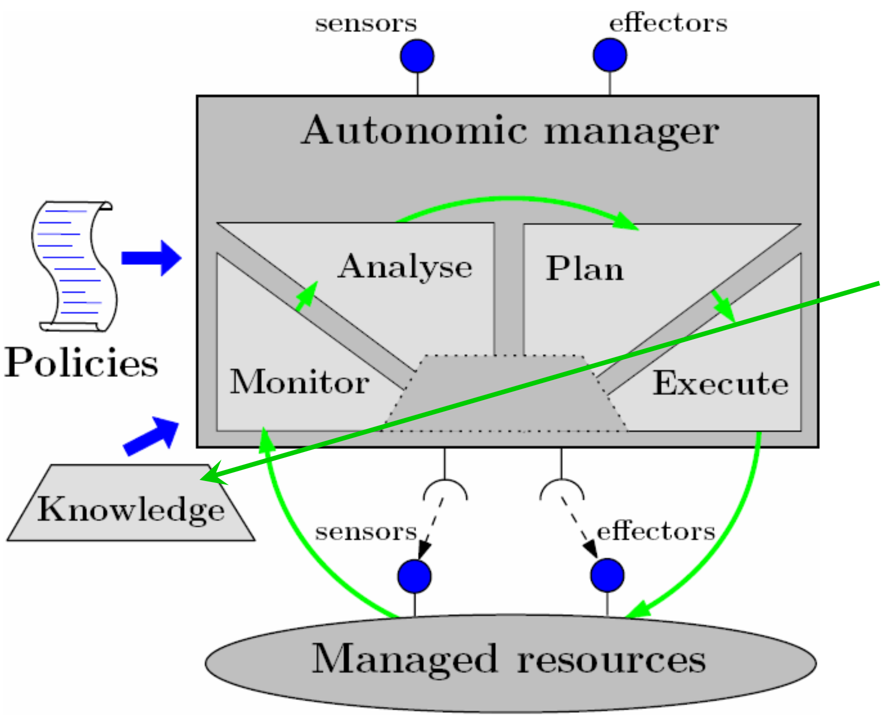
statically-defined high-level interfaces

resource-specific code

application-specific metadata hard-coded within the autonomic manager

manually-implemented interfaces

Generic self-* architecture



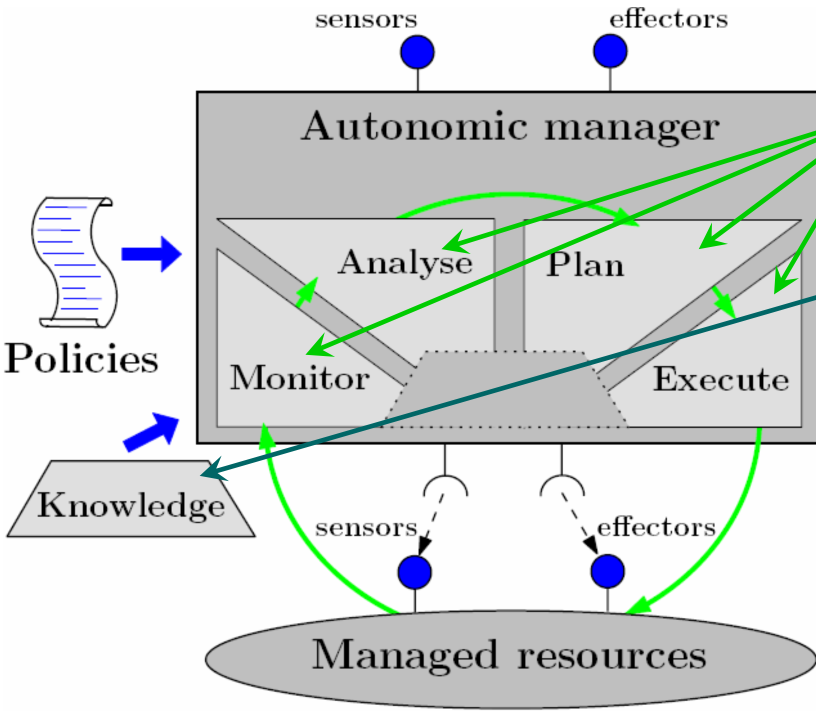
statically-defined high-level interfaces

resource-specific code

supply the “knowledge” at runtime to application-oblivious autonomic manager

manually-implemented interfaces

Generic self-* architecture



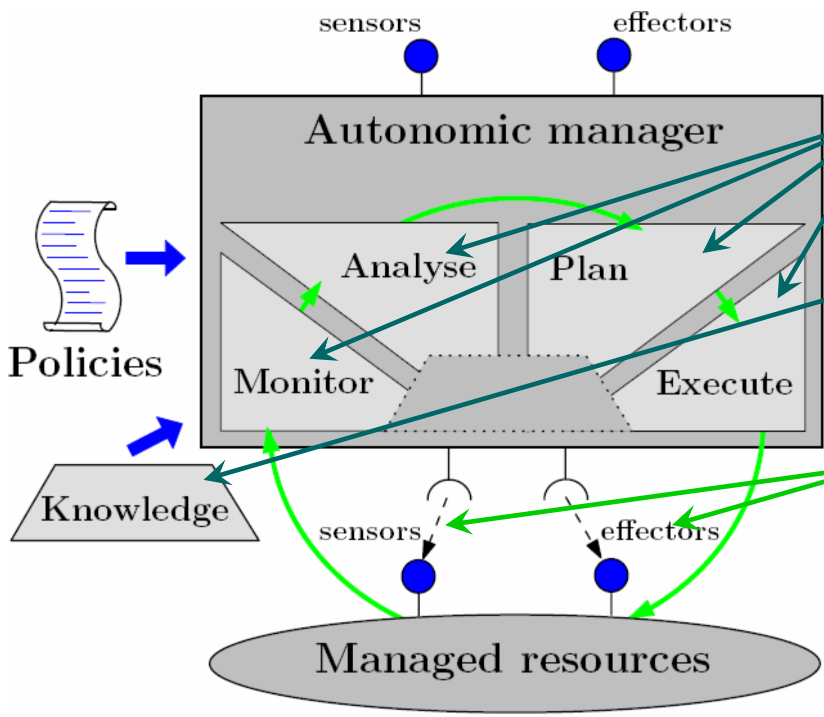
statically-defined high-level interfaces

resource-oblivious code

supply the “knowledge” at runtime to application-oblivious autonomic manager

manually-implemented interfaces

Generic self-* architecture



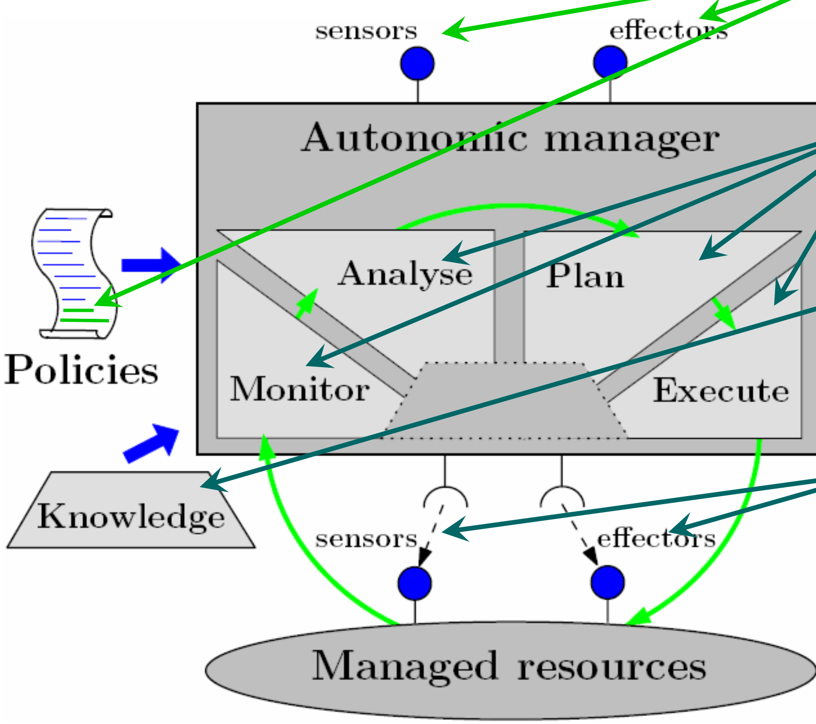
statically-defined high-level interfaces

resource-oblivious code

supply the "knowledge" at runtime to application-oblivious autonomic manager

automatic generation of autonomic manager interfaces & computer-aided development of resource interfaces

Generic self-* architecture



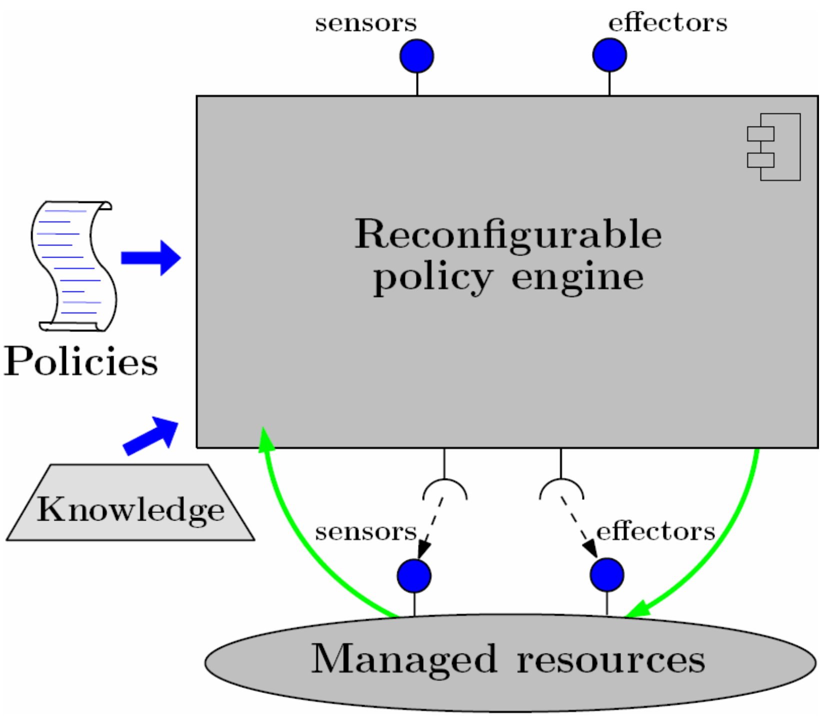
policy-driven, runtime generation of high-level interfaces

resource-oblivious code

supply the “knowledge” at runtime to application-oblivious autonomic manager

automatic generation of autonomic manager interfaces & computer-aided development of resource interfaces

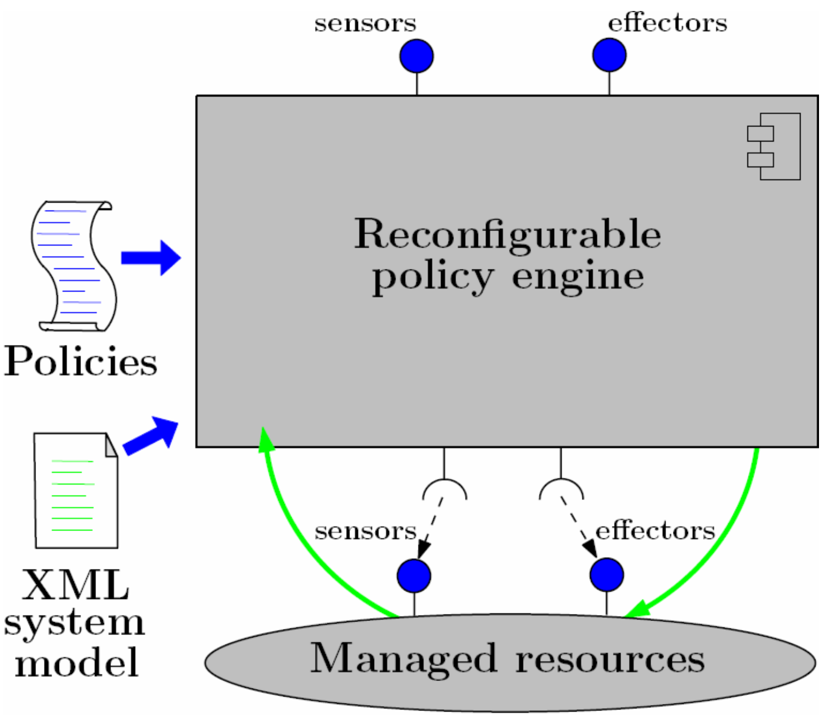
Implementation



Policy engine = (.NET/C#) web service

- platform independence
- standards-based support for security
- loose coupling

Implementation

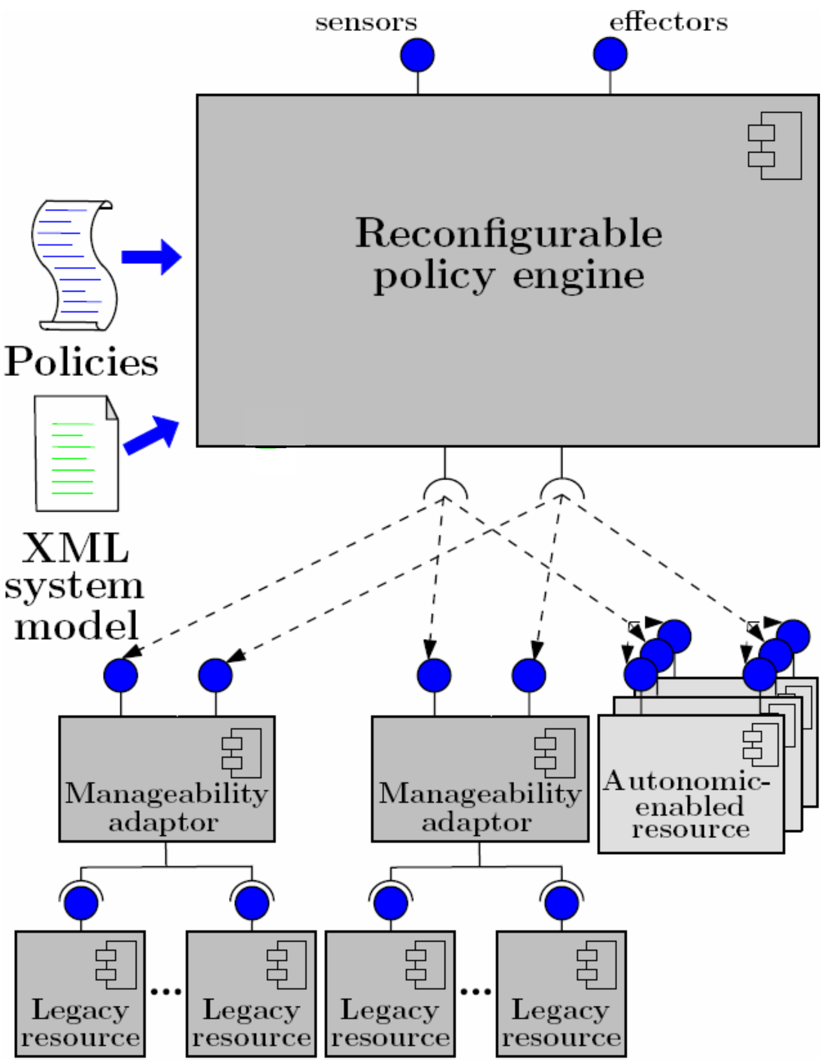


Policy engine = (.NET/C#) web service

Knowledge = XML-encoded model of resource parameters & behaviour

- off-the-shelf tools to process model (XSLT engines, XSD code generators)

Implementation



Policy engine = (.NET/C#) web service

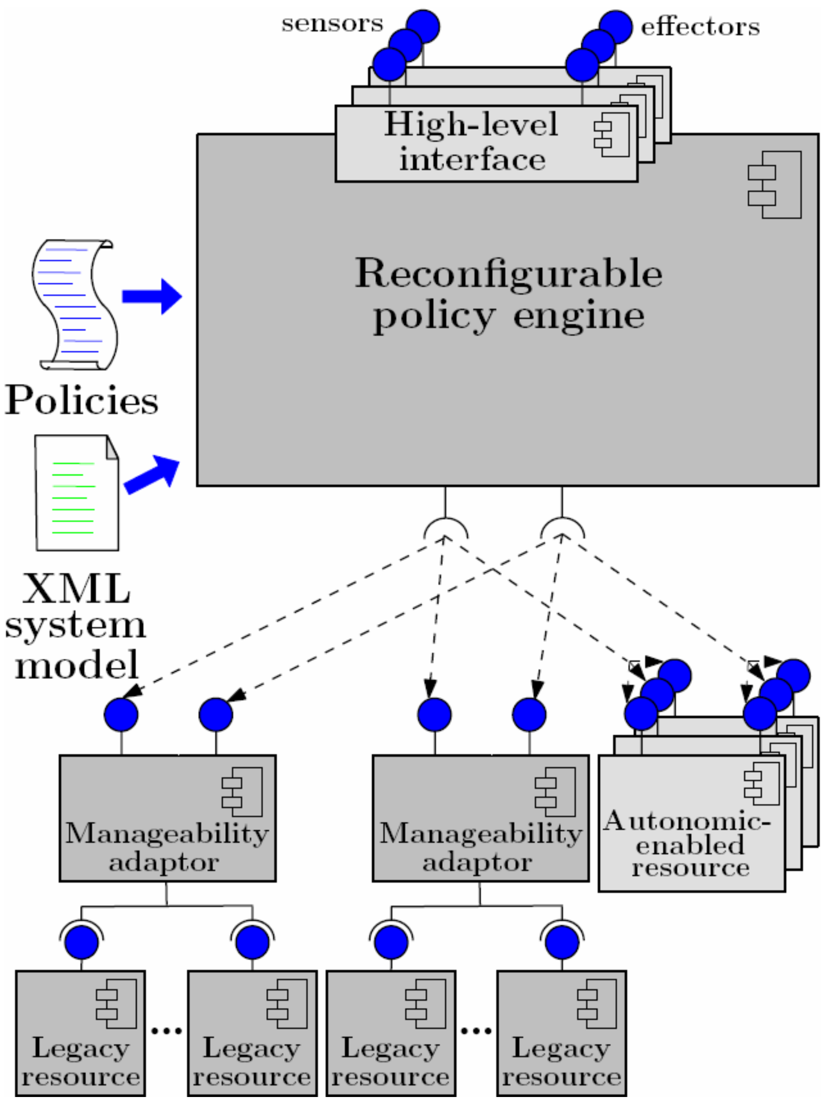
Knowledge = XML-encoded model of resource parameters & behaviour

Manageability adaptors = web services that subclass abstract *Adaptor* class

- *Adaptor* class implements the bulk of the functionality
- several other components generated automatically from the system model

Policy engine interface = generated online from model using OO *reflection*

Implementation



Policy engine = (.NET/C#) web service

Knowledge = XML-encoded model of resource parameters & behaviour

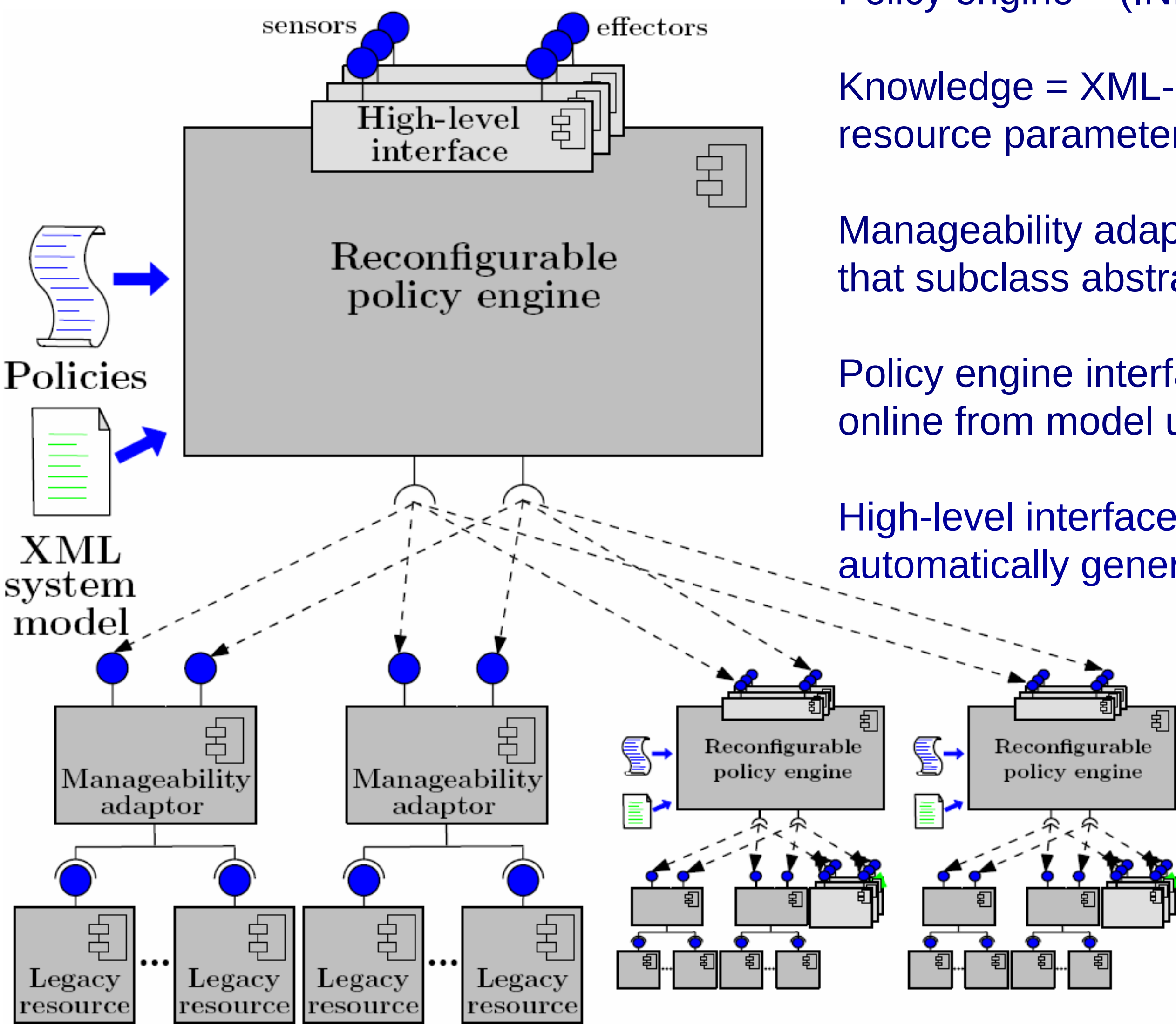
Manageability adaptors = web services that subclass abstract *Adaptor* class

Policy engine interface = generated online from model using OO *reflection*

High-level interfaces = policy-driven, automatically generated web services

- expose system to its environment
- specified by “resource definition” policies
- enable system integration into self-* systems of systems

Implementation – hierarchical system of systems



Policy engine = (.NET/C#) web service

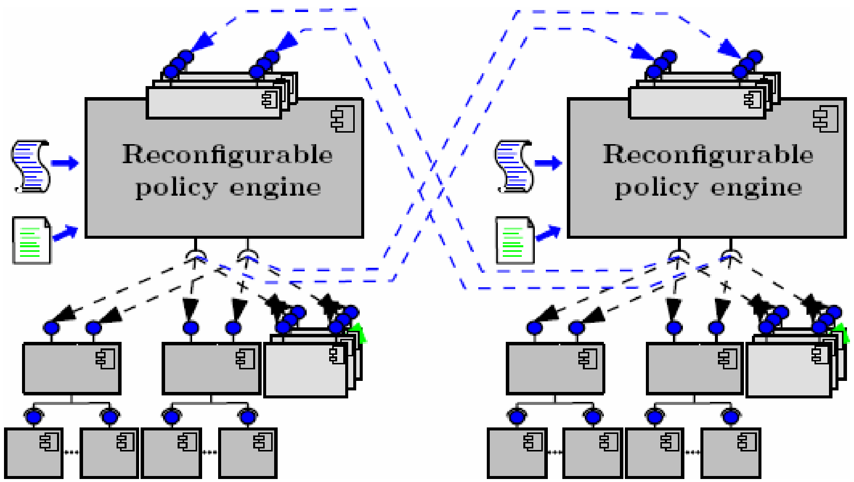
Knowledge = XML-encoded model of resource parameters & behaviour

Manageability adaptors = web services that subclass abstract *Adaptor* class

Policy engine interface = generated online from model using OO *reflection*

High-level interfaces = policy-driven, automatically generated web services

Implementation – system-of-systems federation



Policy engine = (.NET/C#) web service

Knowledge = XML-encoded model of resource parameters & behaviour

Manageability adaptors = web services that subclass abstract *Adaptor* class

Policy engine interface = generated online from model using OO *reflection*

High-level interfaces = policy-driven, automatically generated web services

Application development

generate/implement application-specific components

(system developer)

Generation

configure policy engine & deploy new components

(system administrator)

Deployment

specify/select self-* policies

(system user)

Exploitation

Application development

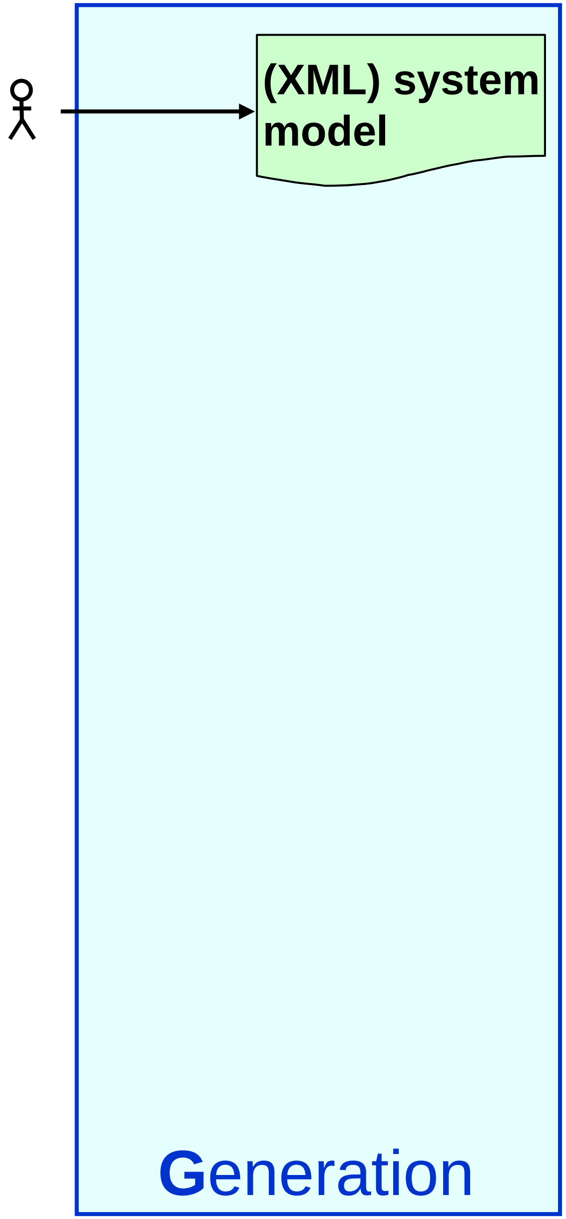
Sample self-* application

Allocate data-centre servers to clusters of different priorities & variable workloads such that they achieve user-defined levels of availability in the presence of cluster component failures.

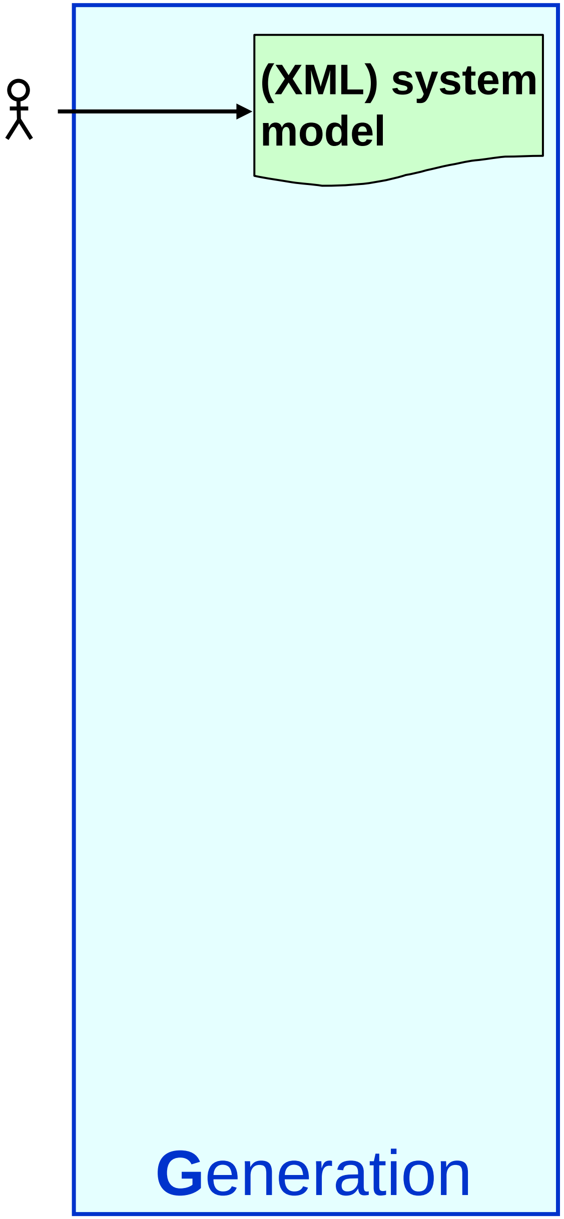
Generation

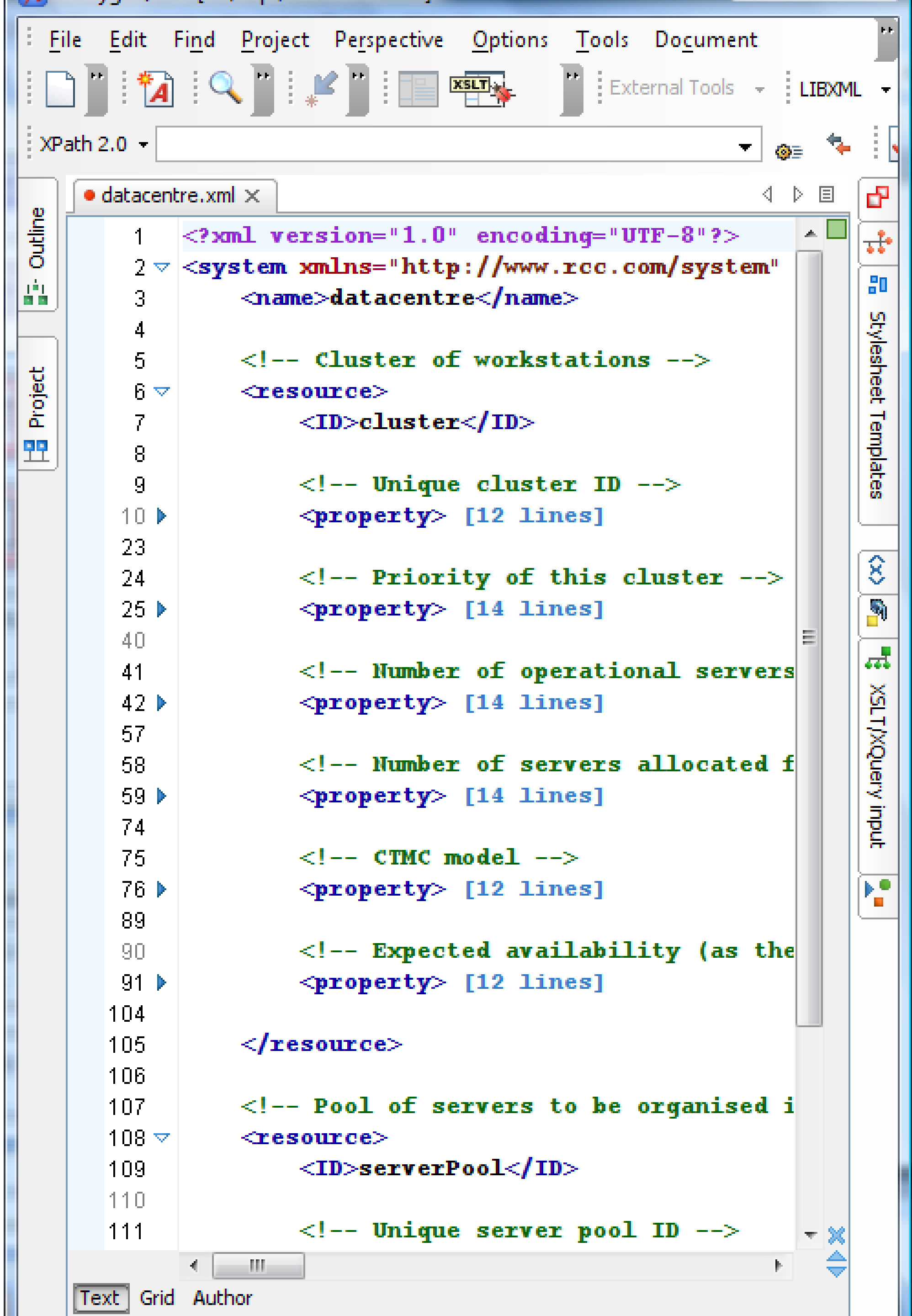
Deployment

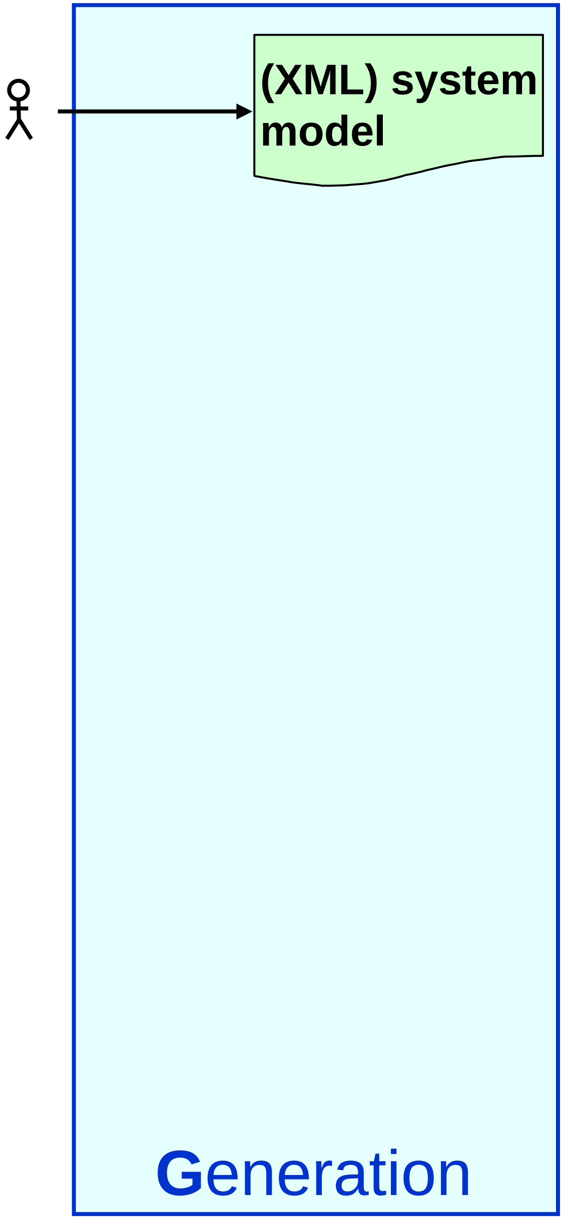
Exploitation



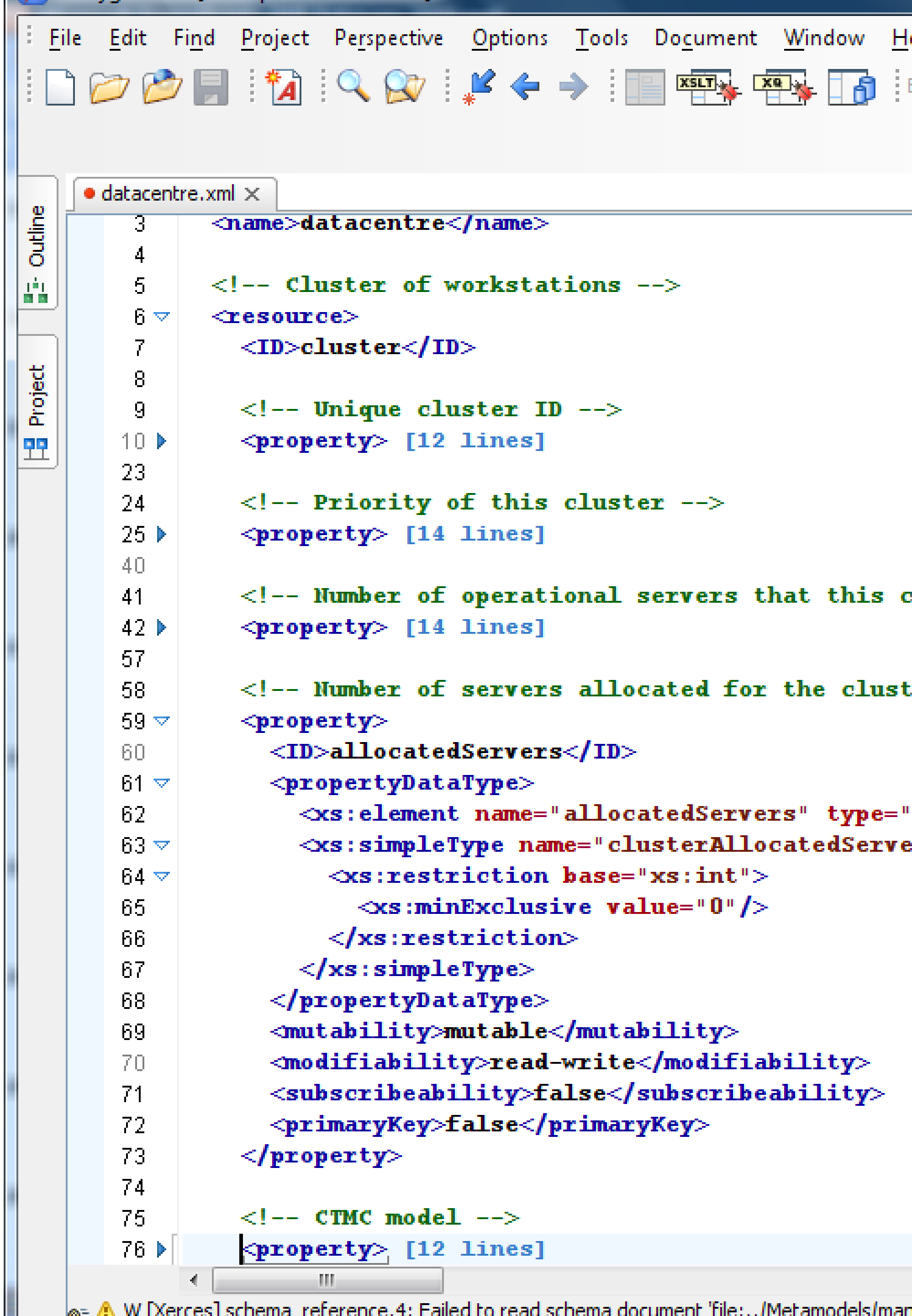
- Build XML model describing the system resources, their parameters and behaviour
- Instance of pre-defined XML schema (*meta-model*)


(XML) system model
Generation

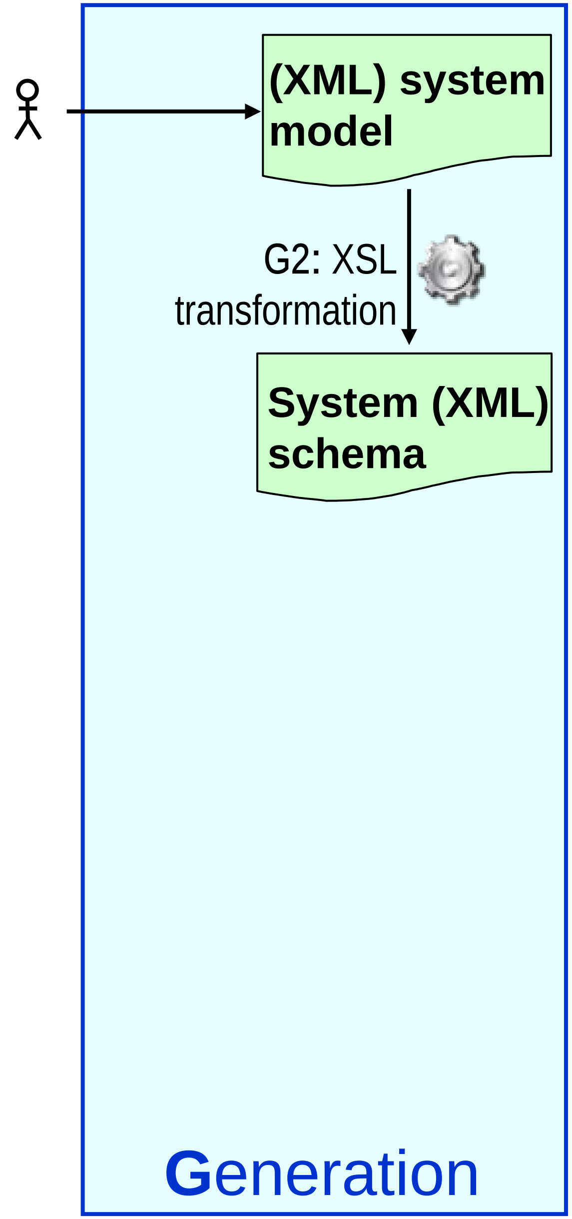

File Edit Find Project Perspective Options Tools Document
External Tools
LIBXML
XPath 2.0
datacentre.xml
Outline
Project
Stylesheet Templates
XSLT/XQuery input
<?xml version="1.0" encoding="UTF-8"?>
<system xmlns="http://www.rcc.com/system"
<name>datacentre</name>
<!-- Cluster of workstations -->
<resource>
<ID>cluster</ID>
<!-- Unique cluster ID -->
<property> [12 lines]
<!-- Priority of this cluster -->
<property> [14 lines]
<!-- Number of operational servers
<property> [14 lines]
<!-- Number of servers allocated f
<property> [14 lines]
<!-- CTMC model -->
<property> [12 lines]
<!-- Expected availability (as the
<property> [12 lines]
</resource>
<!-- Pool of servers to be organised i
<resource>
<ID>serverPool</ID>
<!-- Unique server pool ID -->
Text Grid Author

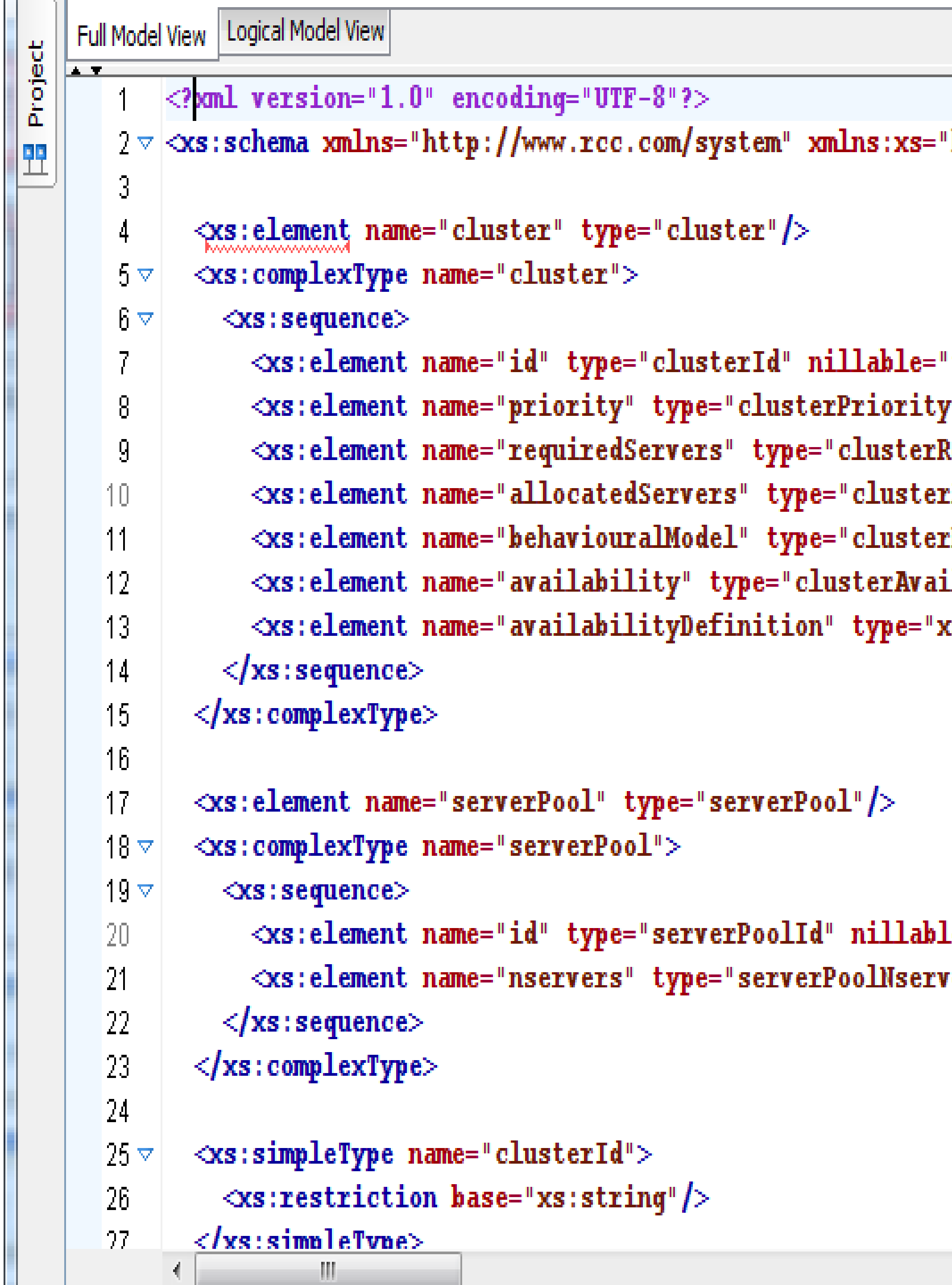


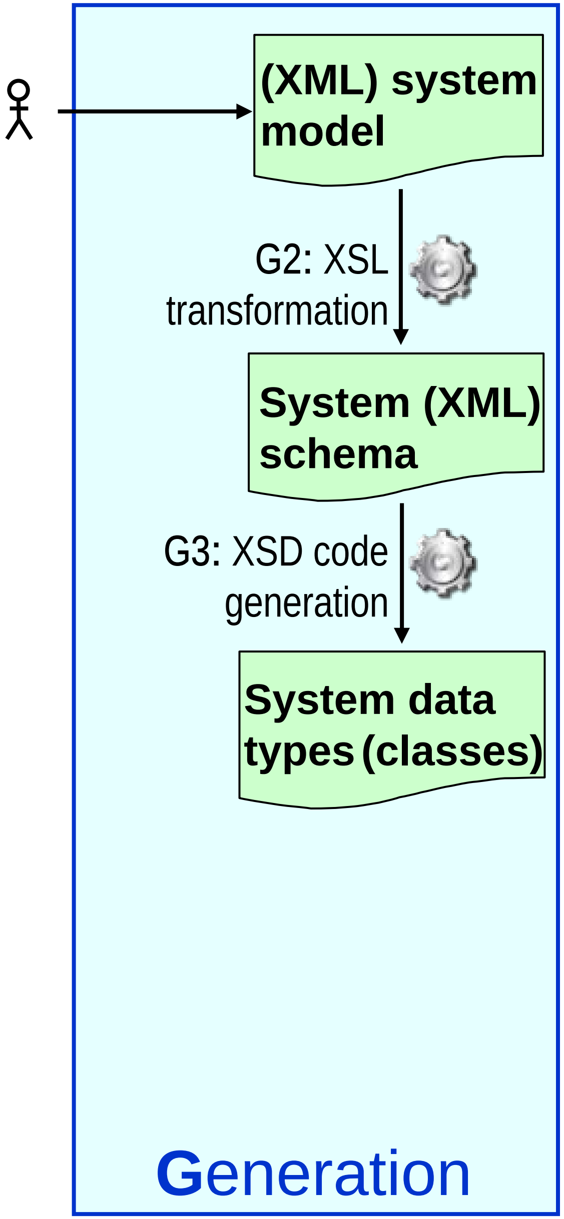
(XML) system model
Generation

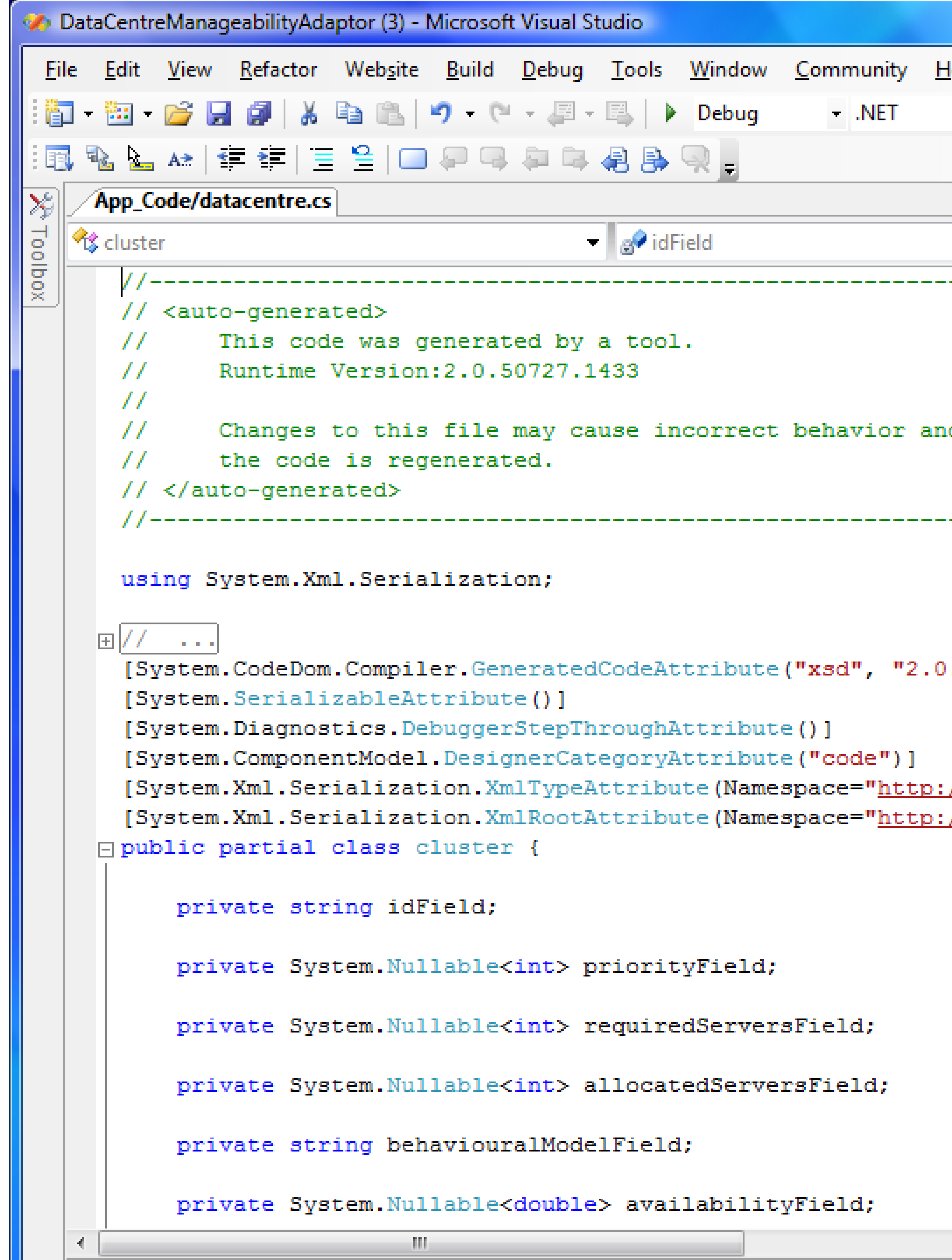


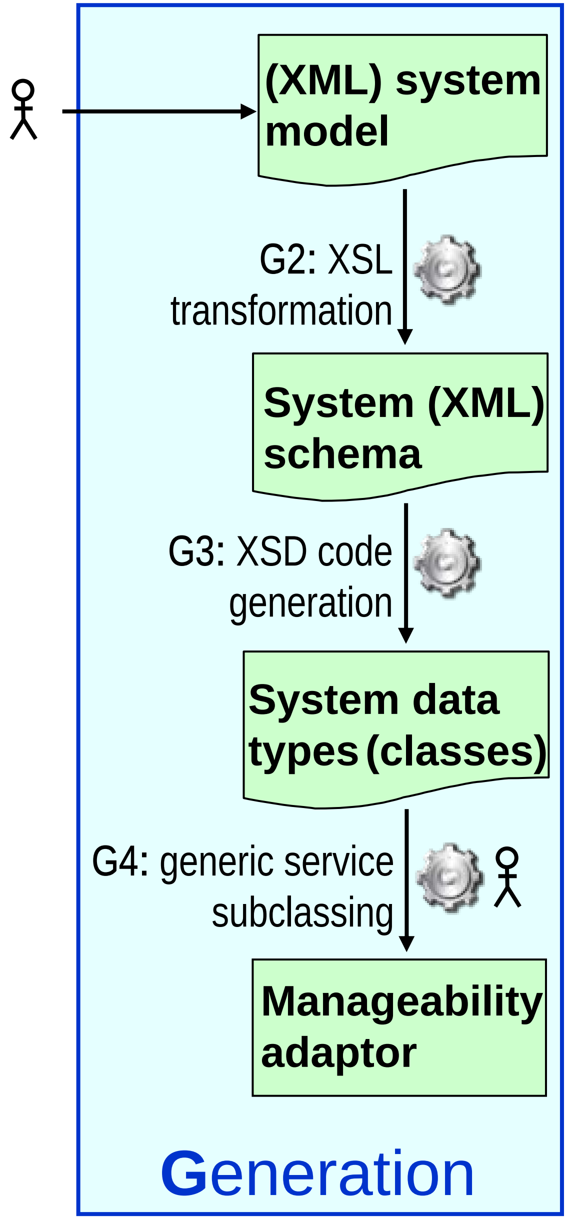
File Edit Find Project Perspective Options Tools Document Window
datacentre.xml
Outline
Project
<name>datacentre</name>
<!-- Cluster of workstations -->
<resource>
<ID>cluster</ID>
<!-- Unique cluster ID -->
<property> [12 lines]
<!-- Priority of this cluster -->
<property> [14 lines]
<!-- Number of operational servers that this c
<property> [14 lines]
<!-- Number of servers allocated for the clust
<property>
<ID>allocatedServers</ID>
<propertyDataType>
<xs:element name="allocatedServers" type="
<xs:simpleType name="clusterAllocatedServe
<xs:restriction base="xs:int">
<xs:minExclusive value="0"/>
</xs:restriction>
</xs:simpleType>
</propertyDataType>
<mutability>mutable</mutability>
<modifiability>read-write</modifiability>
<subscribeability>false</subscribeability>
<primaryKey>false</primaryKey>
</property>
<!-- CTMC model -->
<property> [12 lines]
W [Xerces] schema_reference.4: Failed to read schema document 'file:../Metamodels/mar


(XML) system model
G2: XSL transformation
System (XML) schema
Generation

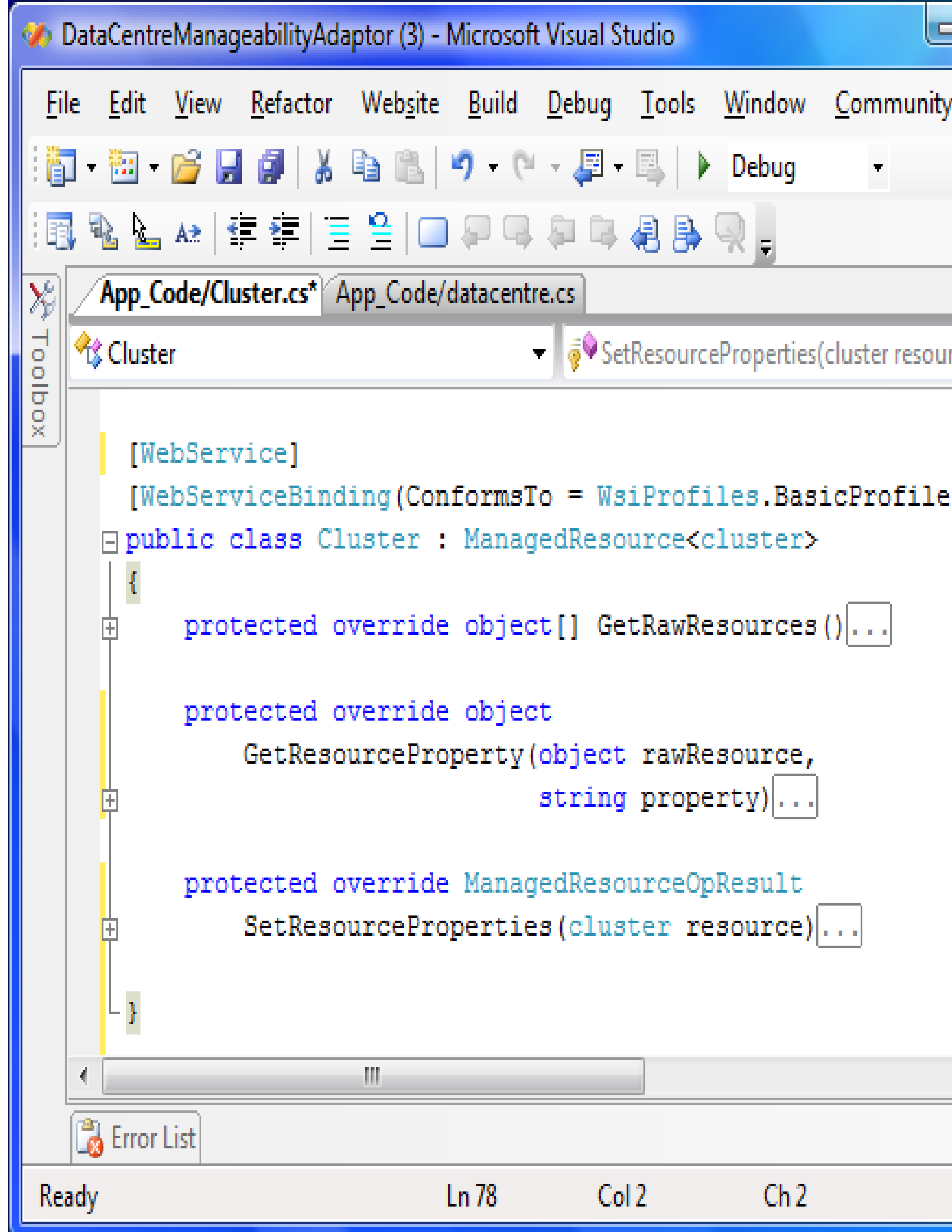

Full Model View
Logical Model View
Project
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.rcc.com/system" xmlns:xs="
<xs:element name="cluster" type="cluster"/>
<xs:complexType name="cluster">
<xs:sequence>
<xs:element name="id" type="clusterId" nillable="
<xs:element name="priority" type="clusterPriority
<xs:element name="requiredServers" type="clusterR
<xs:element name="allocatedServers" type="cluster
<xs:element name="behaviouralModel" type="cluster
<xs:element name="availability" type="clusterAvai
<xs:element name="availabilityDefinition" type="x
</xs:sequence>
</xs:complexType>
<xs:element name="serverPool" type="serverPool"/>
<xs:complexType name="serverPool">
<xs:sequence>
<xs:element name="id" type="serverPoolId" nillabl
<xs:element name="nservers" type="serverPoolNserv
</xs:sequence>
</xs:complexType>
<xs:simpleType name="clusterId">
<xs:restriction base="xs:string"/>


(XML) system model
G2: XSL transformation
System (XML) schema
G3: XSD code generation
System data types (classes)
Generation

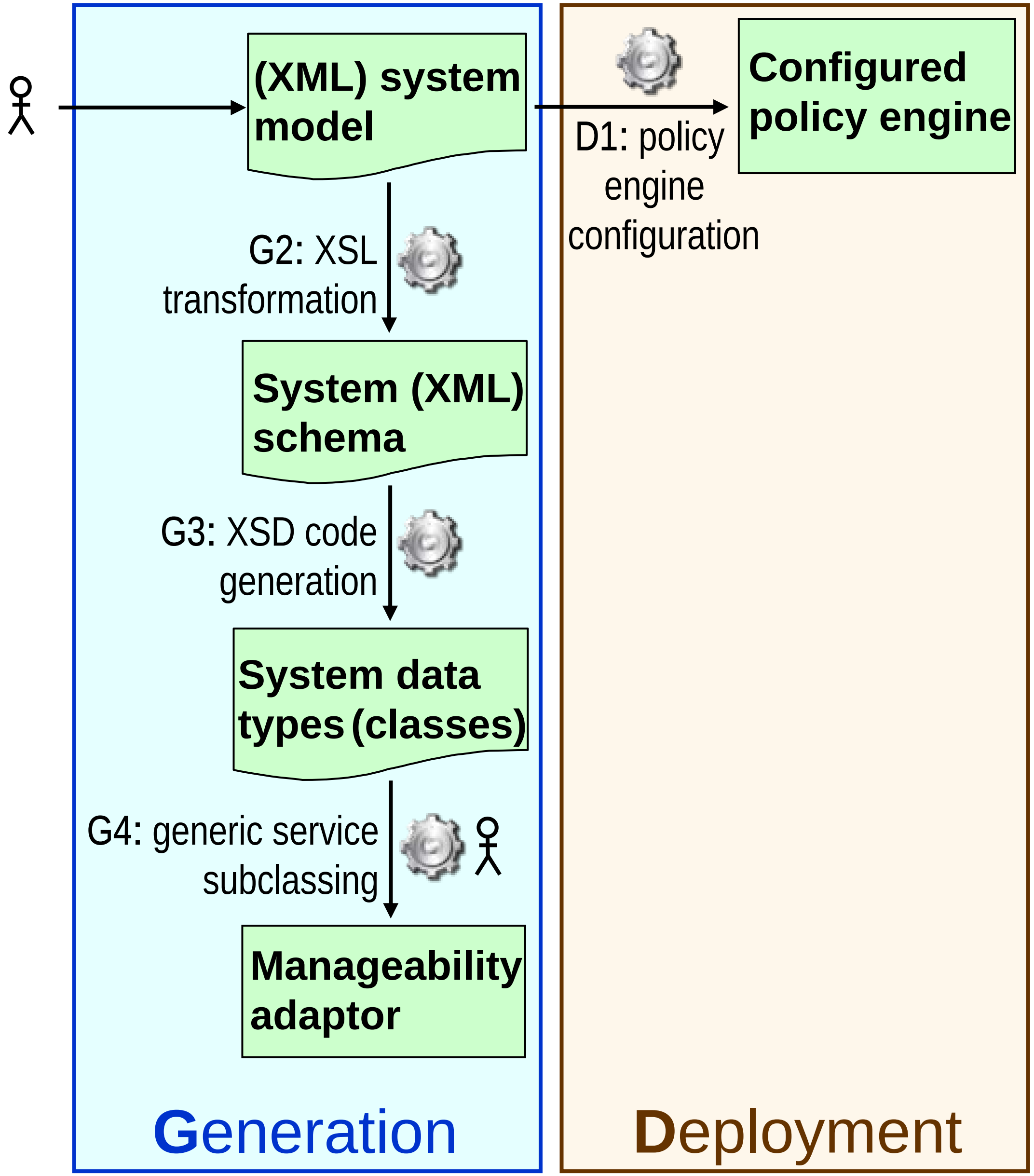

DataCentreManageabilityAdaptor (3) - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Window Community
Debug .NET
Toolbox
App_Code/datacentre.cs
cluster
idField
//--
// <auto-generated>
// This code was generated by a tool.
// Runtime Version:2.0.50727.1433
//
// Changes to this file may cause incorrect behavior and
// the code is regenerated.
// </auto-generated>
//--
using System.Xml.Serialization;
// ...
[System.CodeDom.Compiler.GeneratedCodeAttribute("xsd", "2.0
[System.SerializableAttribute()]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http:
[System.Xml.Serialization.XmlRootAttribute(Namespace="http:
public partial class cluster {
private string idField;
private System.Nullable<int> priorityField;
private System.Nullable<int> requiredServersField;
private System.Nullable<int> allocatedServersField;
private string behaviouralModelField;
private System.Nullable<double> availabilityField;



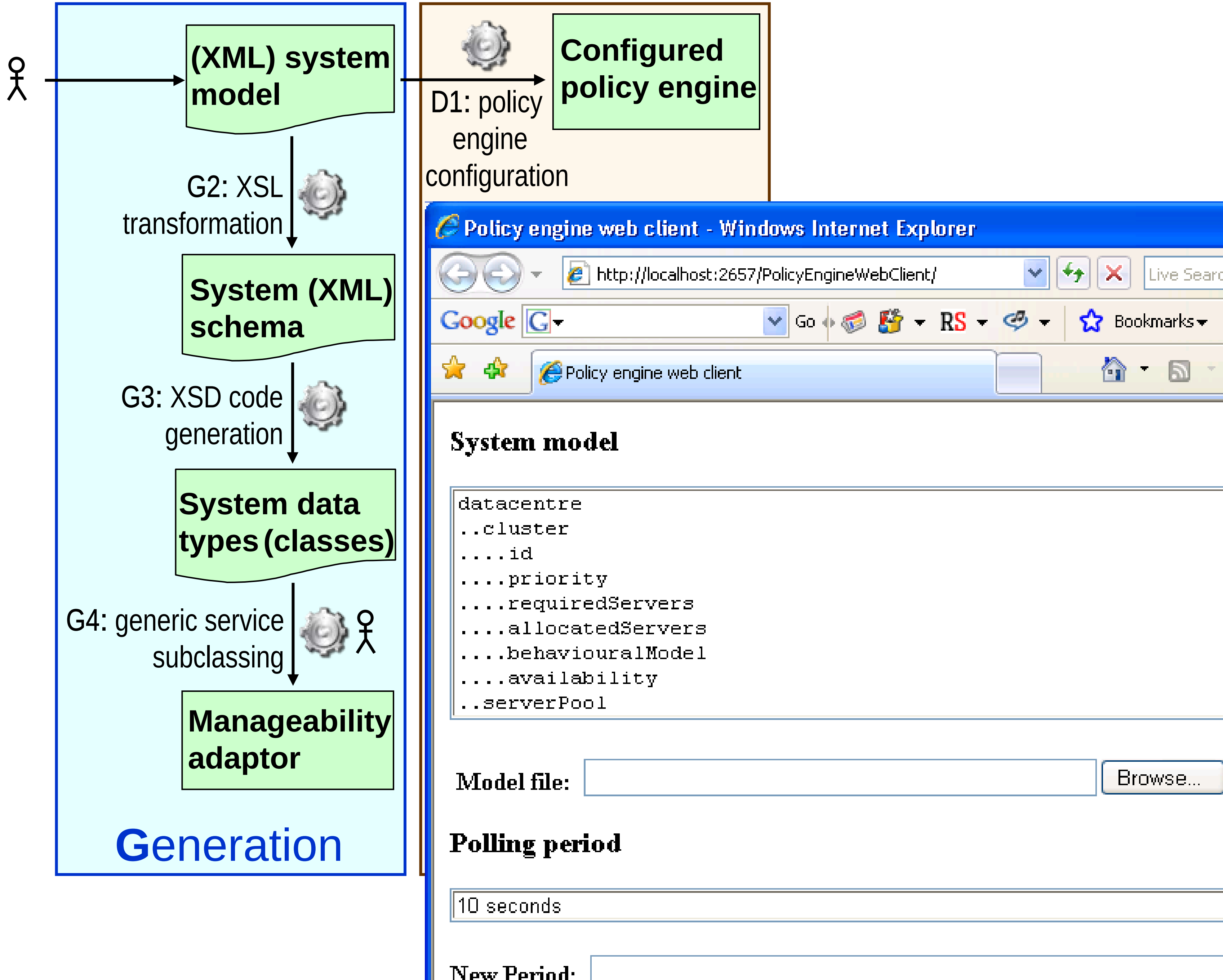
(XML) system model
G2: XSL transformation
System (XML) schema
G3: XSD code generation
System data types (classes)
G4: generic service subclassing
Manageability adaptor
Generation



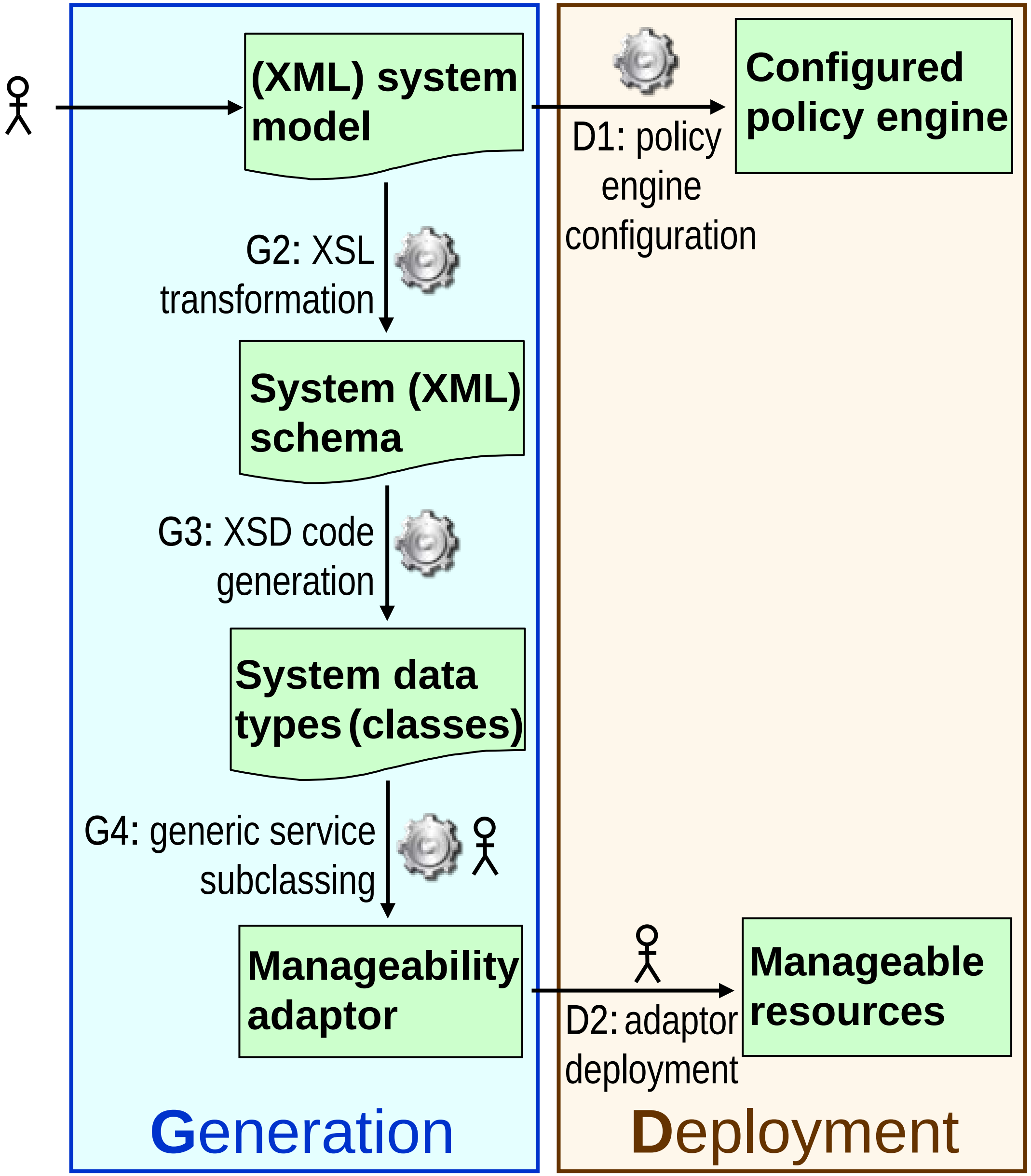
DataCentreManageabilityAdaptor (3) - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Window Community
Debug
Toolbox
App_Code/Cluster.cs* App_Code/datacentre.cs
Cluster
SetResourceProperties(cluster resou
[WebService]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile
public class Cluster : ManagedResource<cluster>
{
protected override object[] GetRawResources()...
protected override object
GetResourceProperty(object rawResource,
string property)...
protected override ManagedResourceOpResult
SetResourceProperties(cluster resource)...
}
Error List
Ready Ln 78 Col 2 Ch 2

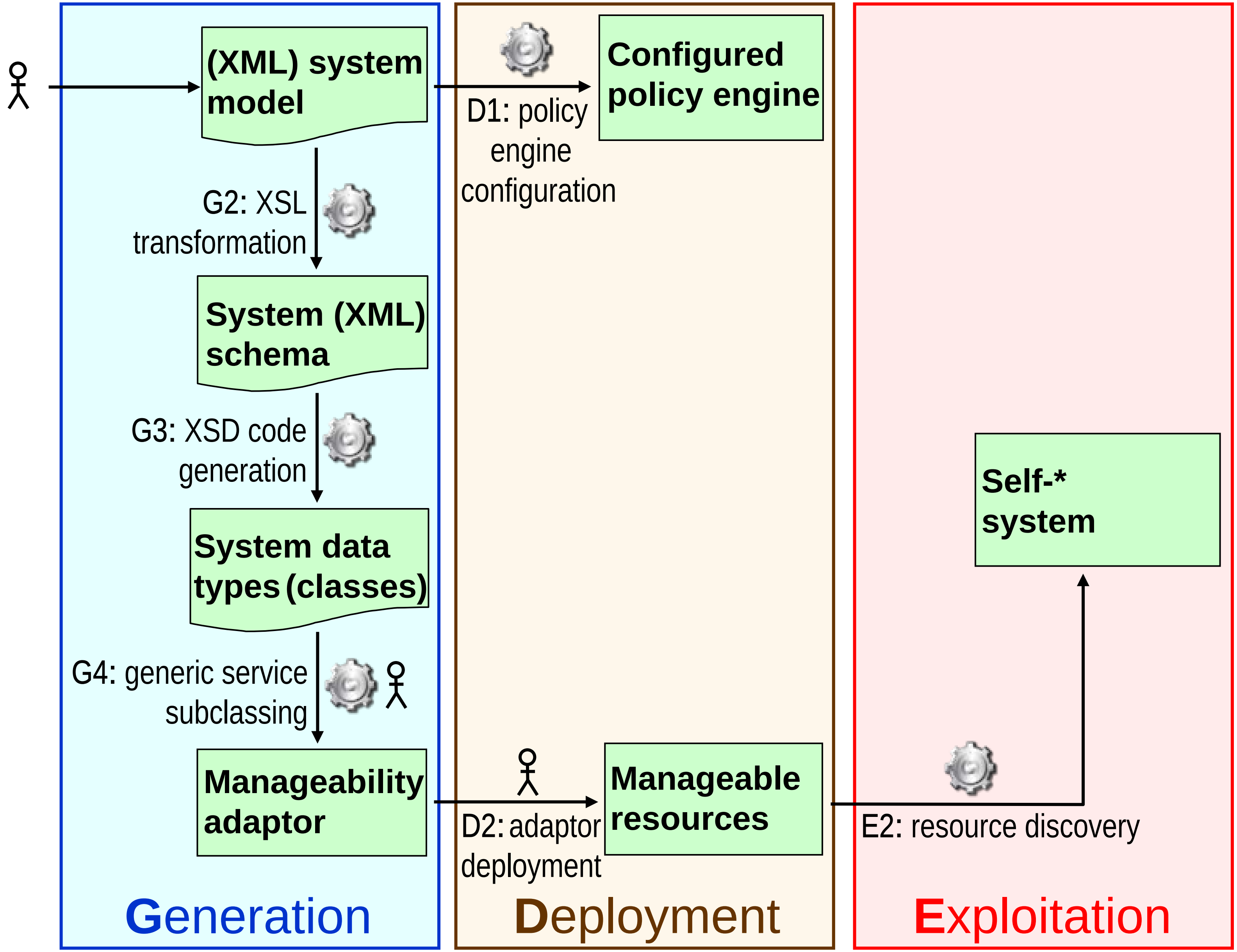


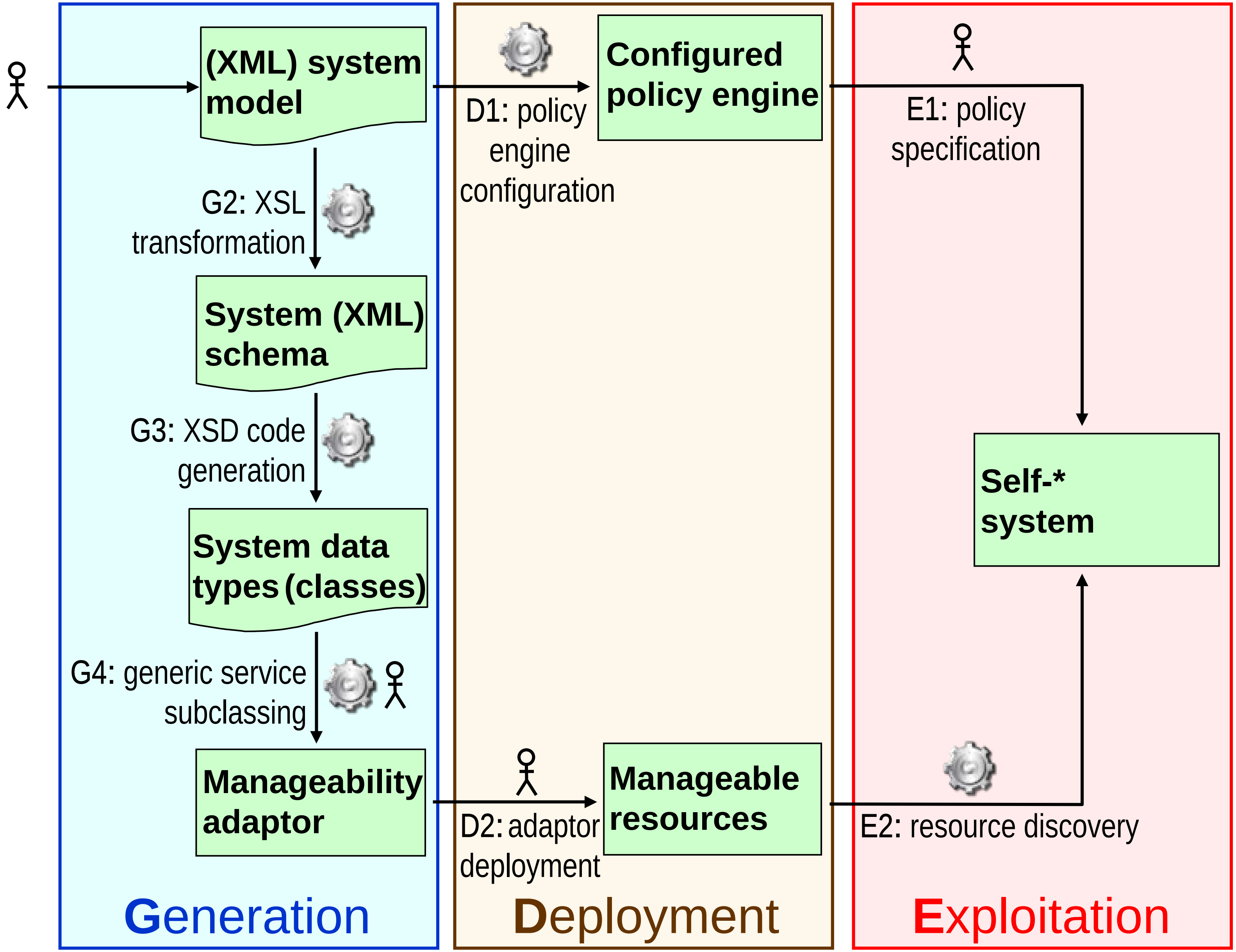
- Use web client to supply model to the policy engine



(XML) system model
G2: XSL transformation
System (XML) schema
G3: XSD code generation
System data types (classes)
G4: generic service subclassing
Manageability adaptor
Generation
D1: policy engine configuration
Configured policy engine
Policy engine web client - Windows Internet Explorer
http://localhost:2657/PolicyEngineWebClient/
Live Searc
Google
Go
RS
Bookmarks
Policy engine web client
System model
datacentre
..cluster
....id
....priority
....requiredServers
....allocatedServers
....behaviouralModel
....availability
..serverPool
Model file:
Browse...
Polling period
10 seconds
New Period:


(XML) system model
G2: XSL transformation
System (XML) schema
G3: XSD code generation
System data types (classes)
G4: generic service subclassing
Manageability adaptor
Generation
D1: policy engine configuration
Configured policy engine
D2: adaptor deployment
Manageable resources
Deployment


(XML) system model
G2: XSL transformation
System (XML) schema
G3: XSD code generation
System data types (classes)
G4: generic service subclassing
Manageability adaptor
Generation
D1: policy engine configuration
Configured policy engine
D2: adaptor deployment
Manageable resources
Deployment
E2: resource discovery
Self-* system
Exploitation


(XML) system model
G2: XSL transformation
System (XML) schema
G3: XSD code generation
System data types (classes)
G4: generic service subclassing
Manageability adaptor
Generation
D1: policy engine configuration
Configured policy engine
D2: adaptor deployment
Manageable resources
Deployment
E1: policy specification
Self-* system
E2: resource discovery
Exploitation

Utility-function policy

$$\mathbf{MAXIMIZE}\left(\sum_{i=1}^{N} priority_i \mathbf{GOAL}(availability_i \geq targetAvailability_i) - \epsilon \sum_{i=1}^{N} allocatedServers_i\right)$$

Utility-function policy

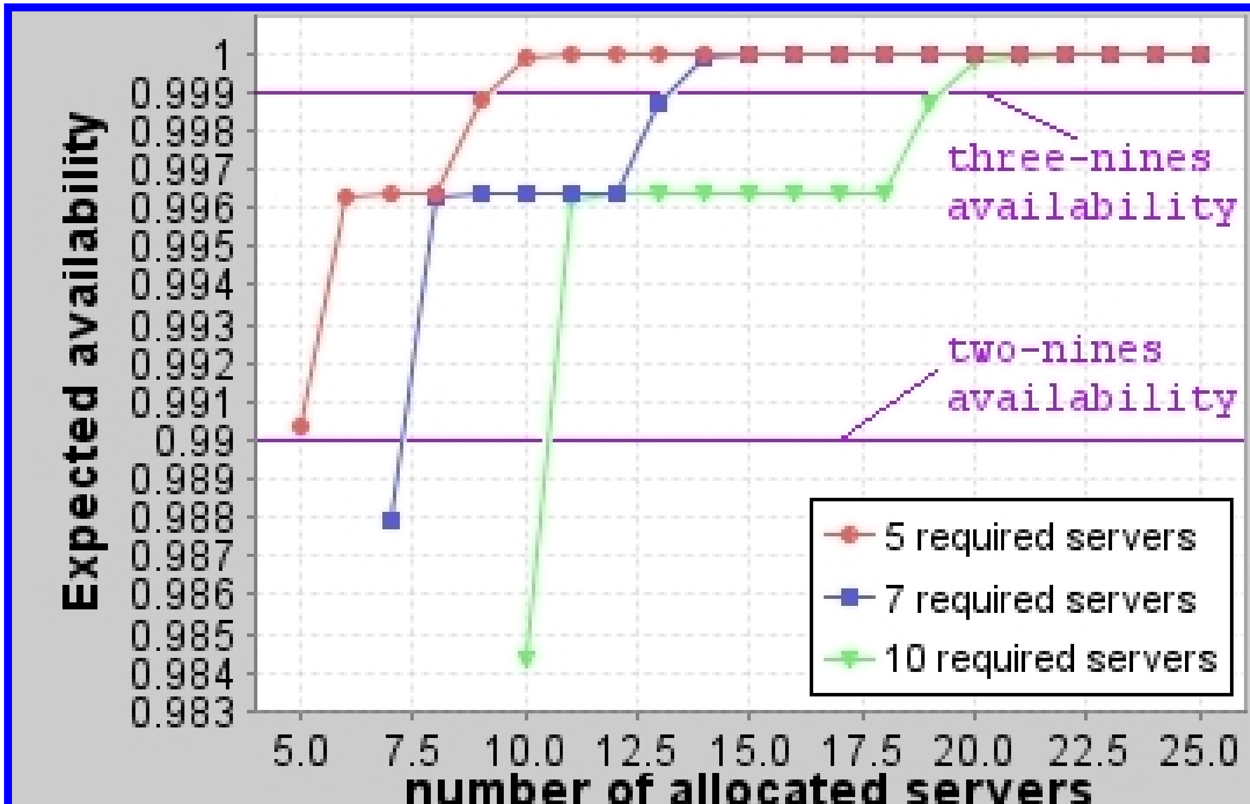
$$\mathbf{MAXIMIZE}\left(\sum_{i=1}^{N} priority_i\,\mathbf{GOAL}(availability_i \geq targetAvailability_i) - \epsilon \sum_{i=1}^{N} allocatedServers_i\right)$$

Utility-function policy

$$\mathbf{MAXIMIZE}\left(\sum_{i=1}^{N} priority_i \mathbf{GOAL}(availability_i \geq targetAvailability_i) - \epsilon \sum_{i=1}^{N} allocatedServers_i\right)$$

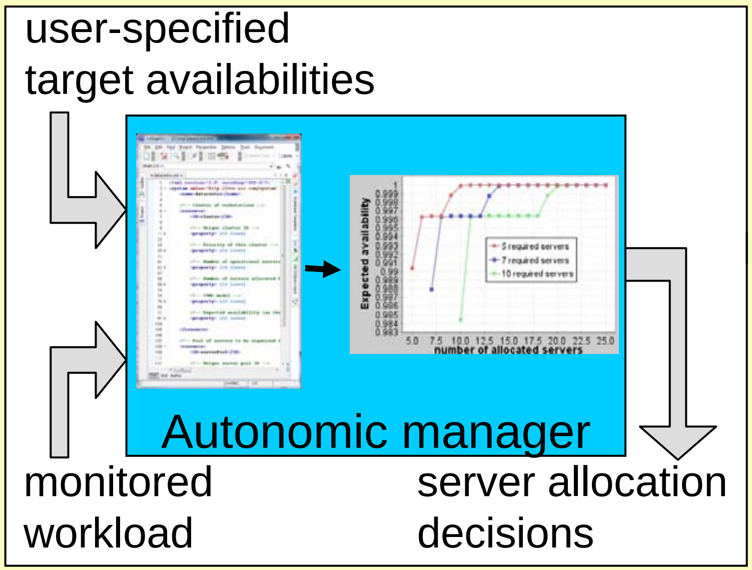
Utility-function policy

$$\mathbf{MAXIMIZE}\left(\sum_{i=1}^{N} priority_i \mathbf{GOAL}(availability_i \geq targetAvailability_i) - \epsilon \sum_{i=1}^{N} allocatedServers_i\right)$$



Online quantitative analysis in self-* systems

Sample self-* application: summary



The allocation of data-centre servers to clusters is managed automatically, based on high-level system objectives specified by data-centre administrators.

Case study summary

	Resource type					self-* areas & policy type					application domain
	software	hardware	data	legacy	autonomic-enabled	main self-* functional areas	action	goal	utility-function	resource-definition	
Allocation of CPU capacity	√			√		self-monitoring self-optimisation			√		CPU capacity allocation
Goal-driven CPU sched.	√			√		self-monitoring self-optimisation		√			CPU capacity allocation
Disk drive DPM		√		√		self-monitoring self-adaptation			√		dynamic power management
Ctrl. of cluster availability		√		√		self-configuration self-protection			√		availability management
Dynamic gen. of web content		√	√	√	√	self-monitoring self-generation	√		√	√	online report generation