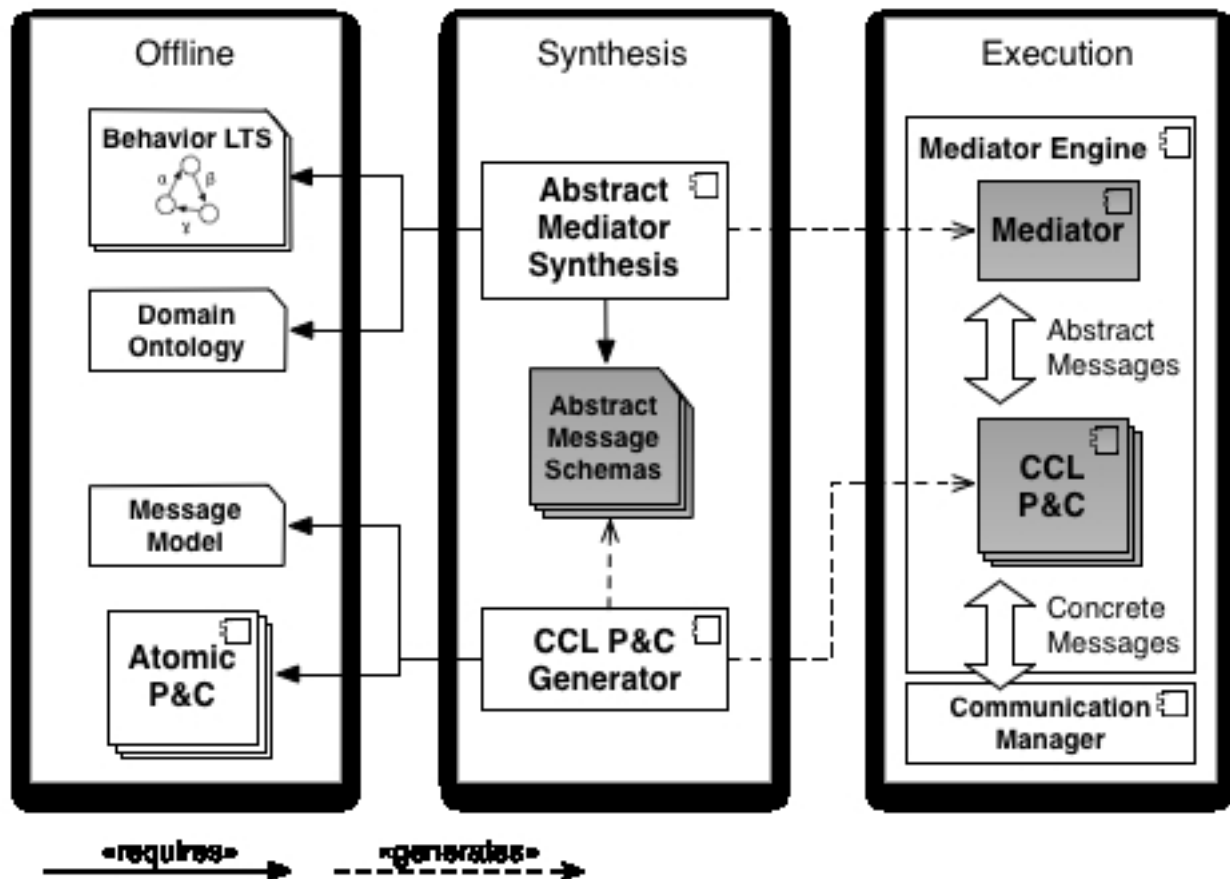


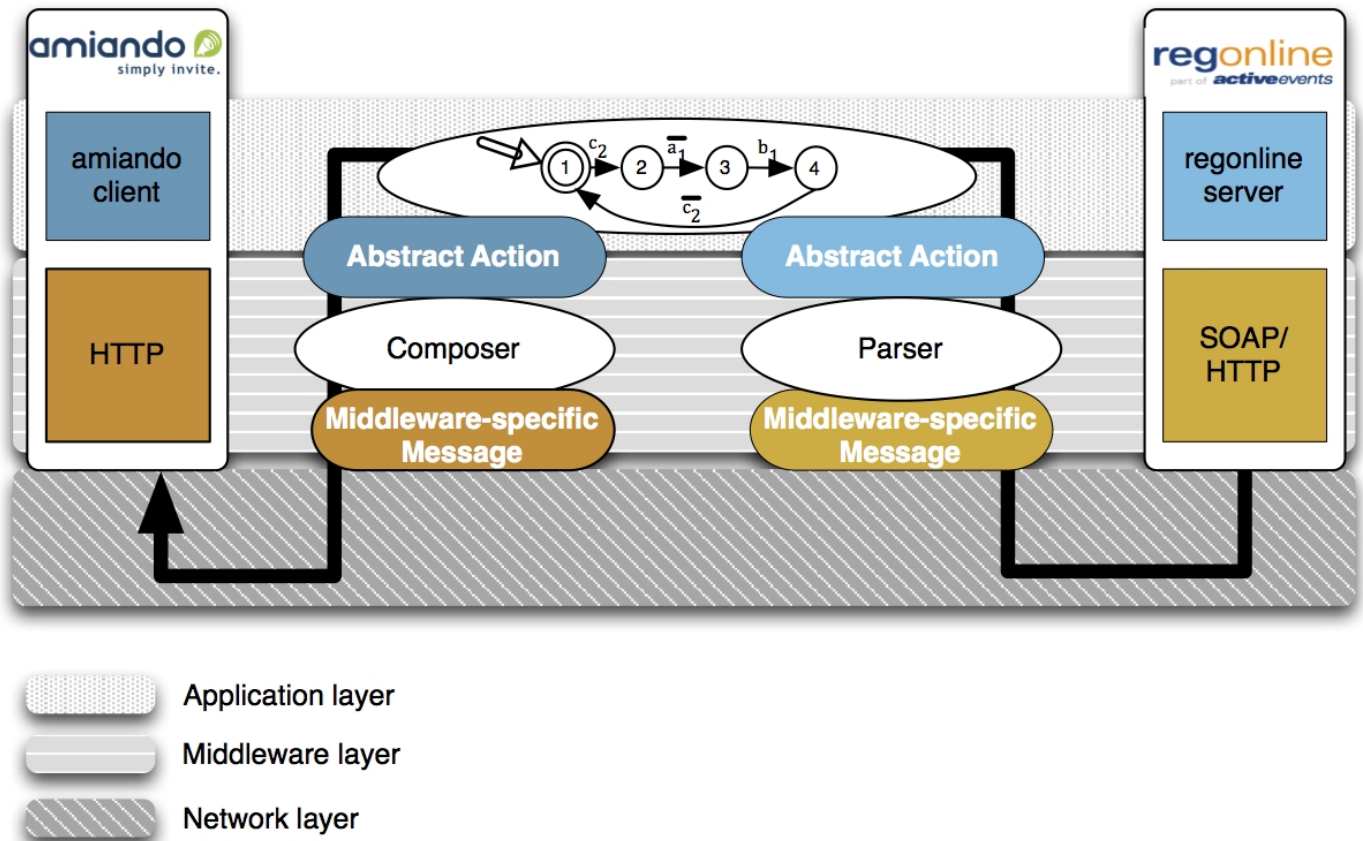
Overview

Today's software systems, and systems of systems, are increasingly developed by composing existing and independently-developed components. Such components often interact using different application protocols and are implemented on top of heterogeneous middleware, which hamper their interoperation. While existing approaches to interoperability consider either application or middleware heterogeneity separately, we argue that in real world scenarios this does not suffice: application and middleware boundaries are ill-defined and solutions to interoperability must consider them in conjunction.

We propose such a solution, which solves cross-layer interoperability by automatically generating parsers and composers that abstract physical message encapsulation layers into logical protocol layers, thus supporting application layer mediation. Further, whenever possible, the framework automatically synthesizes the appropriate protocol mediators between interacting components based on the reasoning about the components' functional and behavioral semantics. % To demonstrate the validity of our approach, we show how the framework solves cross-layer interoperability between existing conference management systems.



Example of Conference Management Systems



Generation of CCL Parsers & Composers

Generate CAM

```
<?xml version="1.0" encoding="utf-8"?>
<application name="RegOnline">
  <operation>Login</operation>
  <operation>GetEvents</operation>
  <extension>
    <ext path="/" oper="*:Request">
      <identifier>org.ambientic.cam.staticpc.http.HttpRequest</identifier>
    </ext>
    <ext path="/" oper="*:Response">
      <identifier>org.ambientic.cam.staticpc.http.HttpResponse</identifier>
    </ext>
    <ext path="/body" oper="*:Request">
      <identifier>org.ambientic.cam.staticpc.soap.SoapEnvelope</identifier>
    </ext>
    <ext path="/body" oper="*:Response">
      <identifier>org.ambientic.cam.staticpc.soap.SoapEnvelope</identifier>
    </ext>
    <ext path="/body/body" oper="Login:Request">
      <identifier>org.ambientic.cam.dynamicpc.wsdl.WsdlMessageFactory</id
      <description>regonline.wsdl</description>
      <content>com.regonline.api.Login</content>
    </ext>
    <ext path="/body/body" oper="Login:Response">
      <identifier>org.ambientic.cam.dynamicpc.wsdl.WsdlMessageFactory</id
      <description>regonline.wsdl</description>
      <content>com.regonline.api.LoginResponse</content>
    </ext>
    <ext path="/body/body" oper="GetEvents:Request">
      <identifier>org.ambientic.cam.dynamicpc.wsdl.WsdlMessageFactory</id
      <description>regonline.wsdl</description>
```

Atomic Parsers & Composers

- org.ambientic.cam.staticpc.soap.SoapEnvelope
- org.ambientic.cam.dynamicpc.json.JsonLearnFactory
- org.ambientic.cam.staticpc.http.HttpResponse
- org.ambientic.cam.dynamicpc.wsdl.WsdlMessageFactory
- org.ambientic.cam.staticpc.http.HttpRequest

Combined App. & Middleware P. & C.

- Login:Response
- GetEvents:Request
- Login:Request
- GetEvents:Response

Message samples:

Add Remove

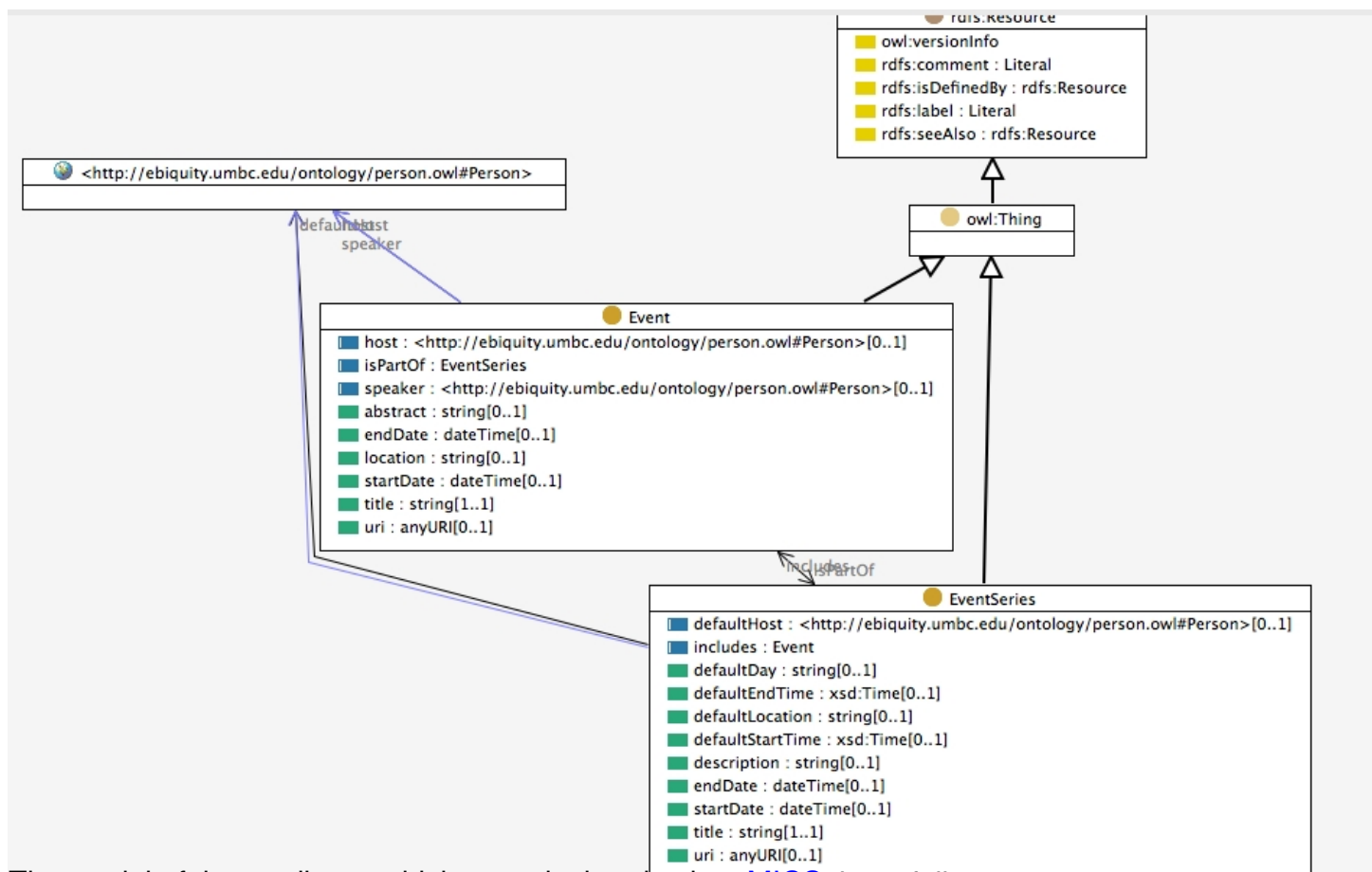
RegOnline Message Model

- HttpResponseGetEventsResponse2 : org.ambientic.cam.staticpc.http.HttpResponseGetEventsResponse2
 - head : org.ambientic.cam.staticpc.http.ResponseHead
 - status : java.lang.Integer
 - version : java.lang.String
 - optionalHeaders : java.util.List
 - name : java.lang.String
 - value : java.lang.String
 - body : org.ambientic.cam.staticpc.soap.SoapEnvelopeGetEventsResponse2
 - header : byte[]
 - body : org.ambientic.cam.dynamicpc.wsdl.WsdlMessageGetEventsResponse2GetEventsResponse
 - description : java.lang.String
 - hidden_content : java.lang.Object
 - content : com.regonline.api.GetEventsResponse
 - getEventsResult : com.regonline.api.ResultsOfListOfEvent
 - success : boolean
 - message : java.lang.String
 - data : com.regonline.api.ArrayOfAPIEvent
 - apiEvent : java.util.List
 - id : int
 - customerID : int

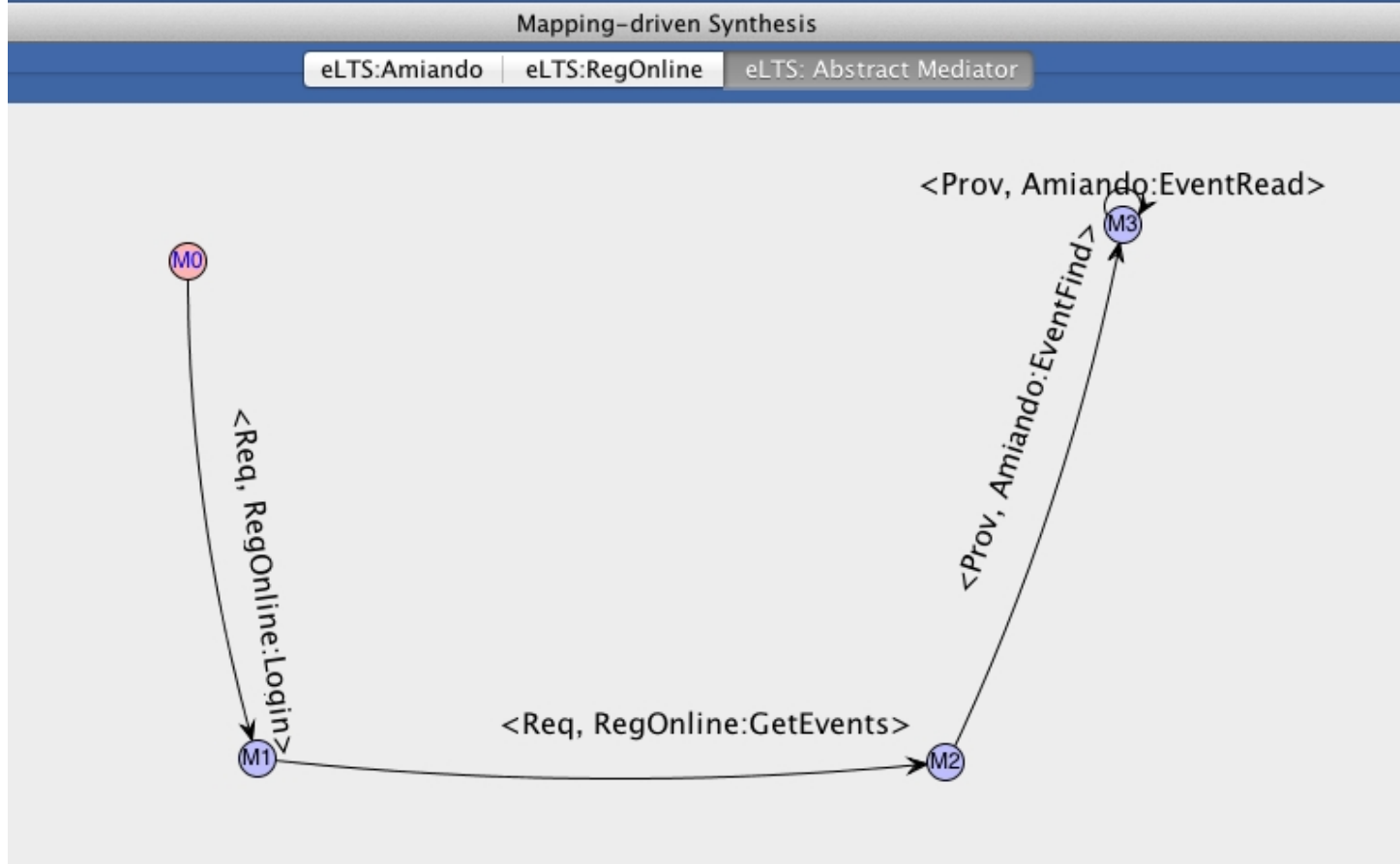
RegOnline Message Model

Generation of Mediators

The [event ontology](#) , which we defined by extending [eBiquity ontology](#) is the following:



The model of the mediator which we calculated using [MCS](#) is as follows:



Contributors

- [Emil Andriescu](#)
- [Amel Bennaceur](#)
- [Roberto Speicys Cardoso](#)
- [Valérie Issarny](#)

Supporting Grant

- [Connect](#) -- IST FP7 FET IP - Emergent Connectors for Eternal Software Intensive Networked Systems

Related Research Projects

- [Dynamic Synthesis of Connectors](#)

Download

-