Adaptability as a Programming Pattern in SEArch

Carlos G. Lopez Pombo

Pablo Montepagano

Emilio Tuosto

WACA

June 20, 2025 Lille

– Prelude –

What is this talk about?

Adaptation in **S**ervice **E**xecution **Arch**itecture, a PoC platform for semantic-based service composition

Bisimilarity as a semantic notion of compliance

What is this talk about?

Adaptation in **S**ervice **E**xecution **Arch**itecture, a PoC platform for semantic-based service composition

Bisimilarity as a semantic notion of compliance

$$\text{to} \left\{ \begin{array}{l} \text{search for} \\ \text{and compose} \end{array} \right. \text{distributed service}$$

with support for

multi-language programming

(language-independence via choreographic models)

Plan of the talk

An bird-eye watch of SEArch's choreographic model

Plan of the talk

An bird-eye watch of SEArch's choreographic model

An overview of SEArch's underlying theory

Plan of the talk

An bird-eye watch of SEArch's choreographic model

An overview of SEArch's underlying theory

A programming pattern for adaptation

Plan of the talk

An bird-eye watch of SEArch's choreographic model

An overview of SEArch's underlying theory

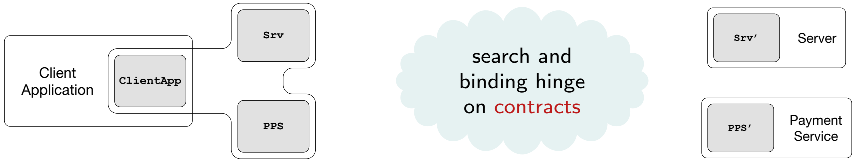
A programming pattern for adaptation

Conclusions

– The underlying theory of SEArch –

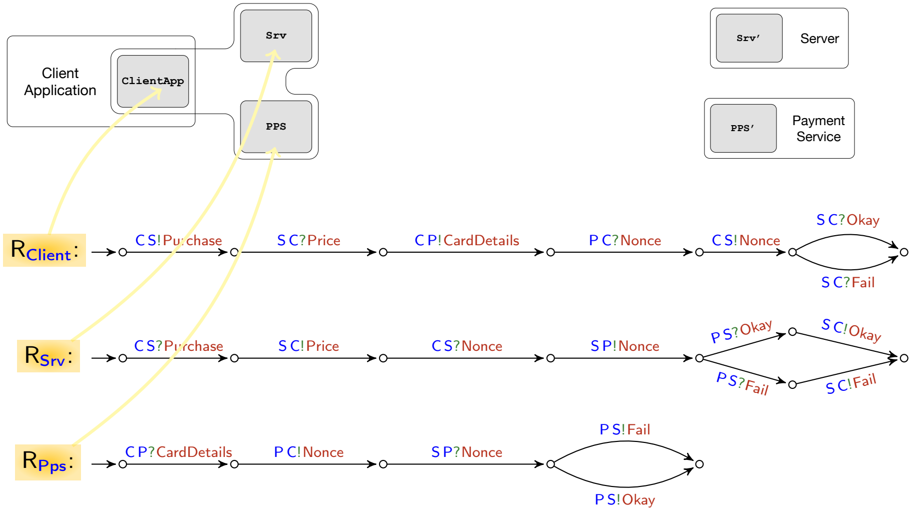
Asynchronous Relational Networks [FL:TCS (503) 2013]

A theory of software architectures of SOAs featuring provide and required interfaces



Asynchronous Relational Networks [FL:TCS (503) 2013]

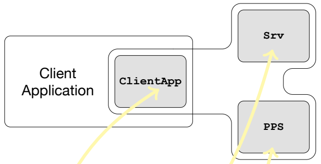
A theory of software architectures of SOAs featuring provide and required interfaces



Contracts as CF-SMs [BZ:JACM 1983] according to [PVT:PLACES 2015,Vis:PhD 2018]

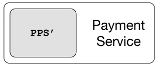
Asynchronous Relational Networks [FL:TCS (503) 2013]

A theory of software architectures of SOAs featuring provide and required interfaces



$R_{Srv} \overset{?}{\sim} P'_{Srv}$

$R_{Pps} \overset{?}{\sim} P'_{Pps}$



R_{Client}:

$$\begin{tikzcd}[column sep=large] {} \arrow[r] & \circ \arrow[r, "C\,S!Purchase"] & \circ \arrow[r, "S\,C?Price"] & \circ \arrow[r, "C\,P!CardDetails"] & \circ \arrow[r, "P\,C?Nonce"] & \circ \arrow[r, "C\,S!Nonce"] & \circ \arrow[r, bend left, "S\,C?Okay"] \arrow[r, bend right, "S\,C?Fail"'] & \circ \end{tikzcd}$$

R_{Srv}:

$$\begin{tikzcd}[column sep=large] & & & & & & \circ \arrow[dr, "S\,C!Okay"] & \\ {} \arrow[r] & \circ \arrow[r, "C\,S?Purchase"] & \circ \arrow[r, "S\,C!Price"] & \circ \arrow[r, "C\,S?Nonce"] & \circ \arrow[r, "S\,P!Nonce"] & \circ \arrow[ur, "P\,S?Okay"] \arrow[dr, "P\,S?Fail"'] & & \circ \\ & & & & & & \circ \arrow[ur, "S\,C!Fail"'] & \end{tikzcd}$$

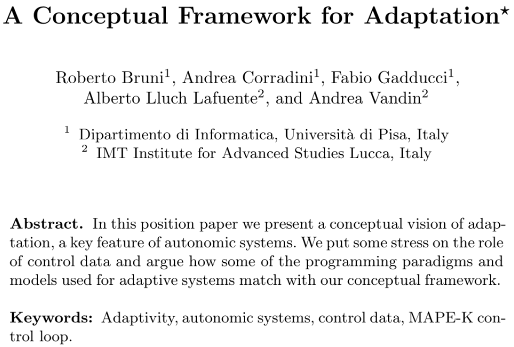
R_{Pps}:

$$\begin{tikzcd}[column sep=large] {} \arrow[r] & \circ \arrow[r, "C\,P?CardDetails"] & \circ \arrow[r, "P\,C!Nonce"] & \circ \arrow[r, "S\,P?Nonce"] & \circ \arrow[r, bend left, "P\,S!Fail"] \arrow[r, bend right, "P\,S!Okay"'] & \circ \end{tikzcd}$$

Contracts as CF-SMs [BZ:JACM 1983] according to [PVT:PLACES 2015,Vis:PhD 2018]

– Adaptation –

A conceptual framework for adaptation



A Conceptual Framework for Adaptation*

Roberto Bruni[1], Andrea Corradini[1], Fabio Gadducci[1], Alberto Lluch Lafuente[2], and Andrea Vandin[2]

[1] Dipartimento di Informatica, Università di Pisa, Italy
[2] IMT Institute for Advanced Studies Lucca, Italy

Abstract. In this position paper we present a conceptual vision of adaptation, a key feature of autonomic systems. We put some stress on the role of control data and argue how some of the programming paradigms and models used for adaptive systems match with our conceptual framework.

Keywords: Adaptivity, autonomic systems, control data, MAPE-K control loop.

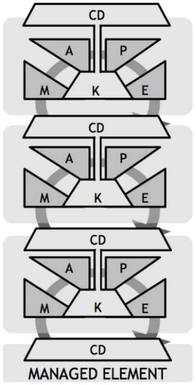


Fig. 2. Tower of adaptation

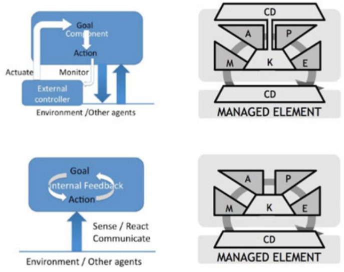


Fig. 3. External (top) and internal (bottom) patterns

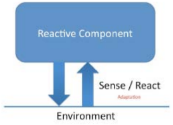


Fig. 4. Reactive pattern

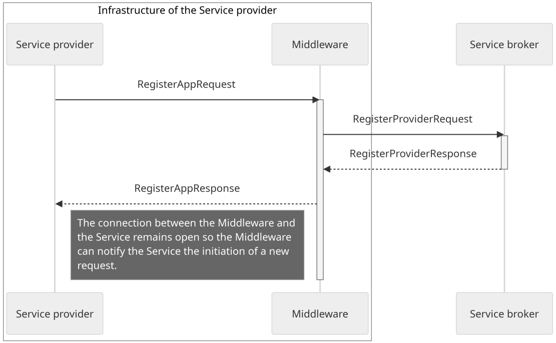
Adapting SEArch to adaptation

A programming pattern for adaption

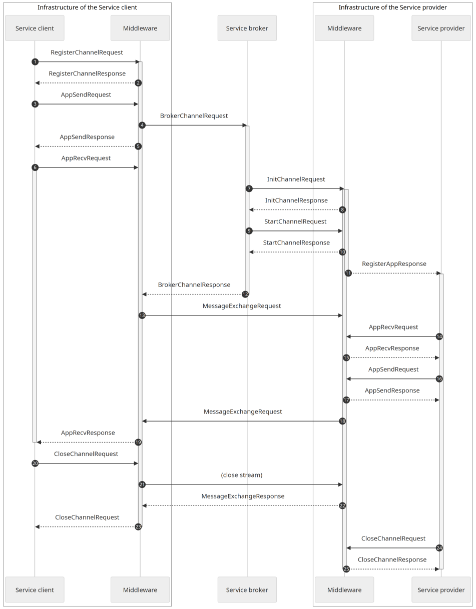
1. a service invocation triggers the registration of the necessary communication and monitoring channels (if any)
2. the control loop continuously

   processes monitoring information
   evaluates environmental conditions based on this information
   chooses an execution scenario
   dispatches the execution of the relevant code
3. the monitor senses the environment and triggers adaptation on environmental changes

SEArch's two protocols

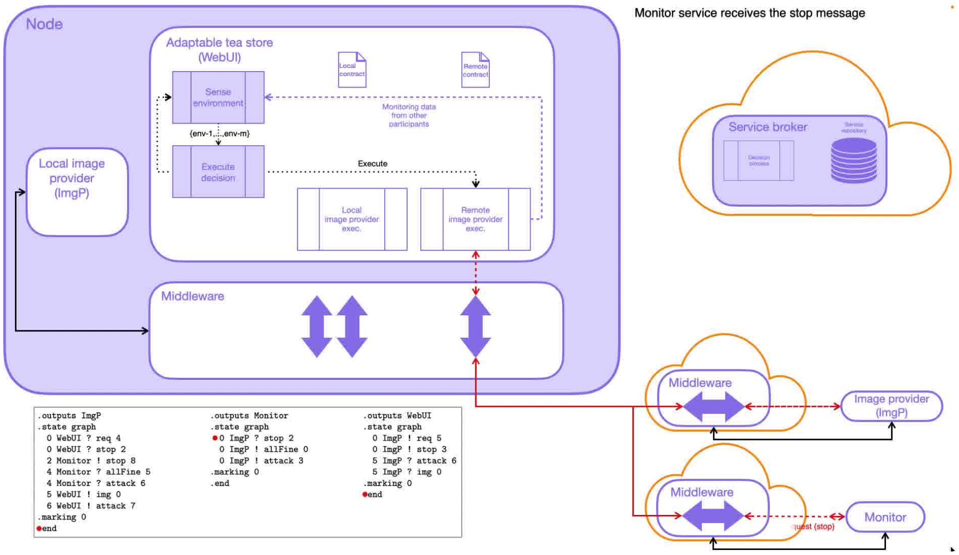


The connection between the Middleware and the Service remains open so the Middleware can notify the Service the initiation of a new request.



Architectural Adaptation in SEArch

Monitor service receives the stop message



```
.outputs ImgP
.state graph
  0 WebUI ? req 4
  0 WebUI ? stop 2
  2 Monitor ! stop 8
  4 Monitor ? allFine 5
  4 Monitor ? attack 6
  5 WebUI ! img 0
  6 WebUI ! attack 7
.marking 0
●end
```

```
.outputs Monitor
.state graph
●0 ImgP ? stop 2
  0 ImgP ! allFine 0
  0 ImgP ! attack 3
.marking 0
.end
```

```
.outputs WebUI
.state graph
  0 ImgP ! req 5
  0 ImgP ! stop 3
  5 ImgP ? attack 6
  5 ImgP ? img 0
.marking 0
●end
```

PYTHON,

```
async def main(grpc_channel):
    stub = search.PrivateMiddlewareServiceStub(grpc_channel)
    registered = False
    logger.info("Connected to middleware. Waiting for registration...")
    async for r in stub.register_app(
        search.RegisterAppRequest(
            provider_contract=search.LocalContract(
                format=search.LocalContractFormat.LOCAL_CONTRACT_FORMAT_FSA,
                contract=PROVIDER_CONTRACT,
            )
        )
    ):
        if registered and r.notification:
            logger.info(f"Notification received: {r.notification}")
            # Start a new session for this channel.
            asyncio.create_task(session(grpc_channel, r.notification))
        elif not registered and r.app_id:
            # This should only happen once, in the first iteration.
            registered = True
            logger.info(f"App registered with id {r.app_id}")
            # Create temp file for Docker Compose healthcheck.
            with open("/tmp/registered", "w") as f:
                f.write("OK")
        else:
            logger.error(f"Unexpected response: {r}. Exiting.")
            break

    grpc_channel.close()
```

PYTHON, GO

```
const ppsContract = `
.outputs PPS
.state graph
q0 ClientApp ? CardDetailsWithTotalAmount q1
q1 ClientApp ! PaymentNonce q2
q2 Srv ? RequestChargeWithNonce q3
q3 Srv ! ChargeOK q4
q3 Srv ! ChargeFail q5
.marking q0
.end
`                                                           // the CFSM in ChorGram syntax
func main() {
        flag.Parse()
        var logger = log.New(os.Stderr, fmt.Sprintf("[PPS] - "), log.LstdFlags|log.Lmsgprefix|log.Lshortfile)
        var opts []grpc.DialOption
        opts = append(opts, grpc.WithTransportCredentials(insecure.NewCredentials()))
        conn, err := grpc.Dial(*middlewareURL, opts...)
        if err != nil {
                logger.Fatalf("Error connecting to middleware URL %s", *middlewareURL)
        }
        defer conn.Close()
        stub := pb.NewPrivateMiddlewareServiceClient(conn)

        // Register provider contract with registry.
        req := pb.RegisterAppRequest{
                ProviderContract: &pb.LocalContract{
                        Contract: []byte(ppsContract),    // passed to the broker upon registration
                        Format:   pb.LocalContractFormat_LOCAL_CONTRACT_FORMAT_FSA,
                },
        }
        streamCtx, streamCtxCancel := context.WithCancel(context.Background())
        defer streamCtxCancel()
        stream, err := stub.RegisterApp(streamCtx, &req)
        if err != nil {
```

PYTHON, GO to JAVA!

```java
public class Main {
  public static void main(String[] args) {
        ...// get book selection and shipping address from the user
        ByteString contractBytes = null; // Load file contract.fsa into a GlobalContract
        try {
               contractBytes = ByteString.readFrom(new FileInputStream("contract.fsa"));
        } catch (IOException e) {
               e.printStackTrace();
        }
        GlobalContract contract = GlobalContract.newBuilder().setContract(contractBytes).setFormat(
          GlobalContractFormat.GLOBAL_CONTRACT_FORMAT_FSA
        ).setInitiatorName("ClientApp").build();
        ...
  }
}
```

where in contract.fsa we find:

```
.outputs ClientApp
.state graph
q0 Srv ! PurchaseRequest q1
q1 Srv ? TotalAmount q2
q2 PPS ! CardDetailsWithTotalAmount q3
q3 PPS ? PaymentNonce q4
q4 Srv ! PurchaseWithPaymentNonce q5
q5 Srv ? PurchaseOK q6
q5 Srv ? PurchaseFail q7
.marking q0
.end
```

```
.outputs Srv
.state graph
q0 ClientApp ? PurchaseRequest q1
q1 ClientApp ! TotalAmount q2
q2 ClientApp ? PurchaseWithPaymentNonce q3
q3 PPS ! RequestChargeWithNonce q4
q4 PPS ? ChargeOK q5
q4 PPS ? ChargeFail q6
q5 ClientApp ! PurchaseOK q7
q6 ClientApp ! PurchaseFail q8
.marking q0
.end
```

```
.outputs PPS
.state graph
q0 ClientApp ? CardDetailsWithTotalAmount q1
q1 ClientApp ! PaymentNonce q2
q2 Srv ? RequestChargeWithNonce q3
q3 Srv ! ChargeOK q4
q3 Srv ! ChargeFail q5
.marking q0
.end
```

– Epilogue –

Recap

SEArch combines

- SOAs
- semantic models (ARNs + CFSMs)
- and tools for choreographic development (eg the data-aware bisimulation on CFSMs in an extension of **ChorGram**)

[1] Apologies for the blunt commercial ;-)

Recap

SEArch combines

- SOAs
- semantic models (ARNs + CFSMs)
- and tools for choreographic development (eg the data-aware bisimulation on CFSMs in an extension of **ChorGram**)

to enable

dynamic and semantic-based discovery and composition of distributed services
and now transparent architectural adaptation

[1] Apologies for the blunt commercial ;-)

Recap

SEArch combines

- SOAs
- semantic models (ARNs + CFSMs)
- and tools for choreographic development (eg the data-aware bisimulation on CFSMs in an extension of ChorGram)

to enable

dynamic and semantic-based discovery and composition of distributed services
and now transparent architectural adaptation

There's space for improvement

- decouple broker and service repository
- $\implies$ distributed bisimulation checks!
- parameterise the compliance check
- ...and QoS! (checkout our COORDINATION 2025 paper[1])

[1]Apologies for the blunt commercial ;-)

For the reviewers

bisimilarity??? really???
“ A discussion would be welcome to argue why bisimilarity is used in this work. ”

For the reviewers

bisimilarity??? really???
“ A discussion would be welcome to argue why bisimilarity is used in this work. ”

“Is this feasible in practice?”

For the reviewers

bisimilarity??? really???
“ A discussion would be welcome to argue why bisimilarity is used in this work. ”

“Is this feasible in practice?”

more contributions

For the reviewers

bisimilarity??? really???
“ A discussion would be welcome to argue why bisimilarity is used in this work. ”

“Is this feasible in practice?”

more contributions

“ This is specific, any other kind of adaptation is possible? ”

For the reviewers

bisimilarity??? really???
“ A discussion would be welcome to argue why bisimilarity is used in this work. ”

“Is this feasible in practice?”

more contributions

“ This is specific, any other kind of adaptation is possible? ”

compare to aspect-oriented middleware

For the reviewers

bisimilarity??? really???
“ A discussion would be welcome to argue why bisimilarity is used in this work. ”

“Is this feasible in practice?”

more contributions

“ This is specific, any other kind of adaptation is possible? ”

compare to aspect-oriented middleware

“ Regarding the contribution of the proposal (section 3), it is presented only abstractly without further discussion or validation. This limited development, combined with the extensive repetition of earlier work, significantly undermines the impact of what could have been an intriguing contribution. ”