Generating Heterogeneous Executable Specifications in SystemC from UML/MARTE Models

Authors:

P.Peñil J.Medina H.Posadas E.Villar

Index

- Motivation
- Design Flow
- UML/MARTE Methodology
- Application Examples
- Conclusions
- Future Work

Motivation

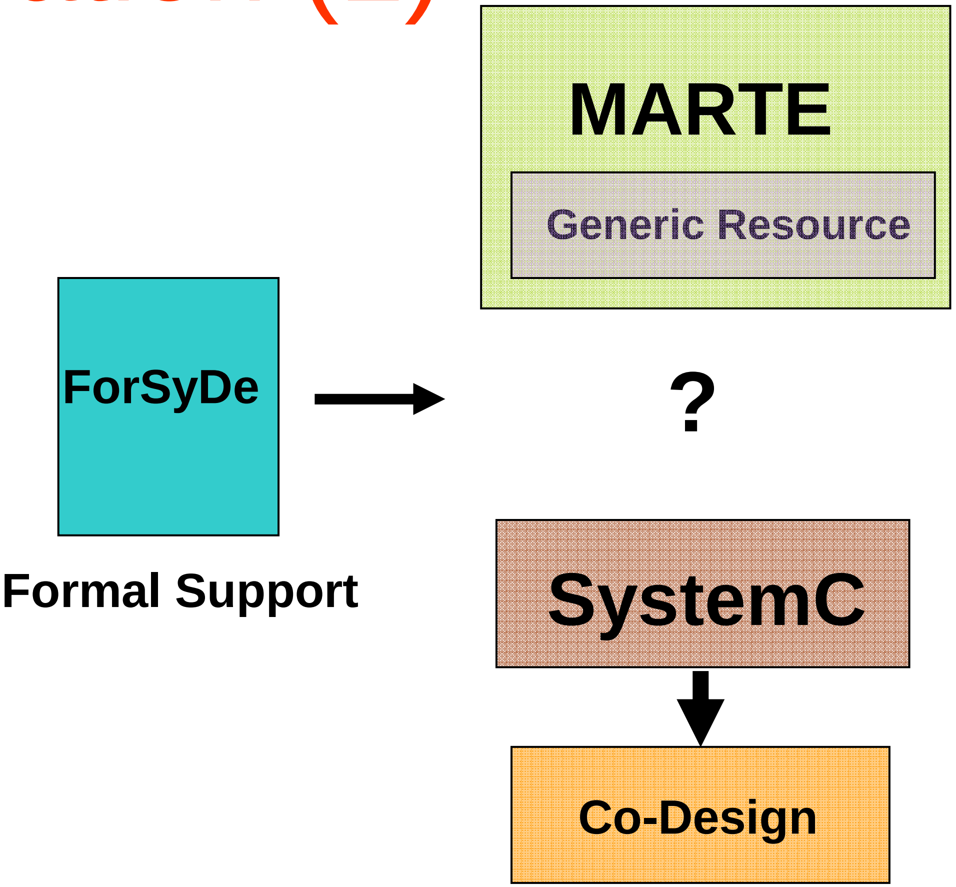
Motivation (1)

MARTE provides semantics to UML

Select a subset of MARTE

Relate UML/MARTE to SystemC

SystemC enables a link to Co-Design



MARTE

Generic Resource

ForSyDe

?

Formal Support

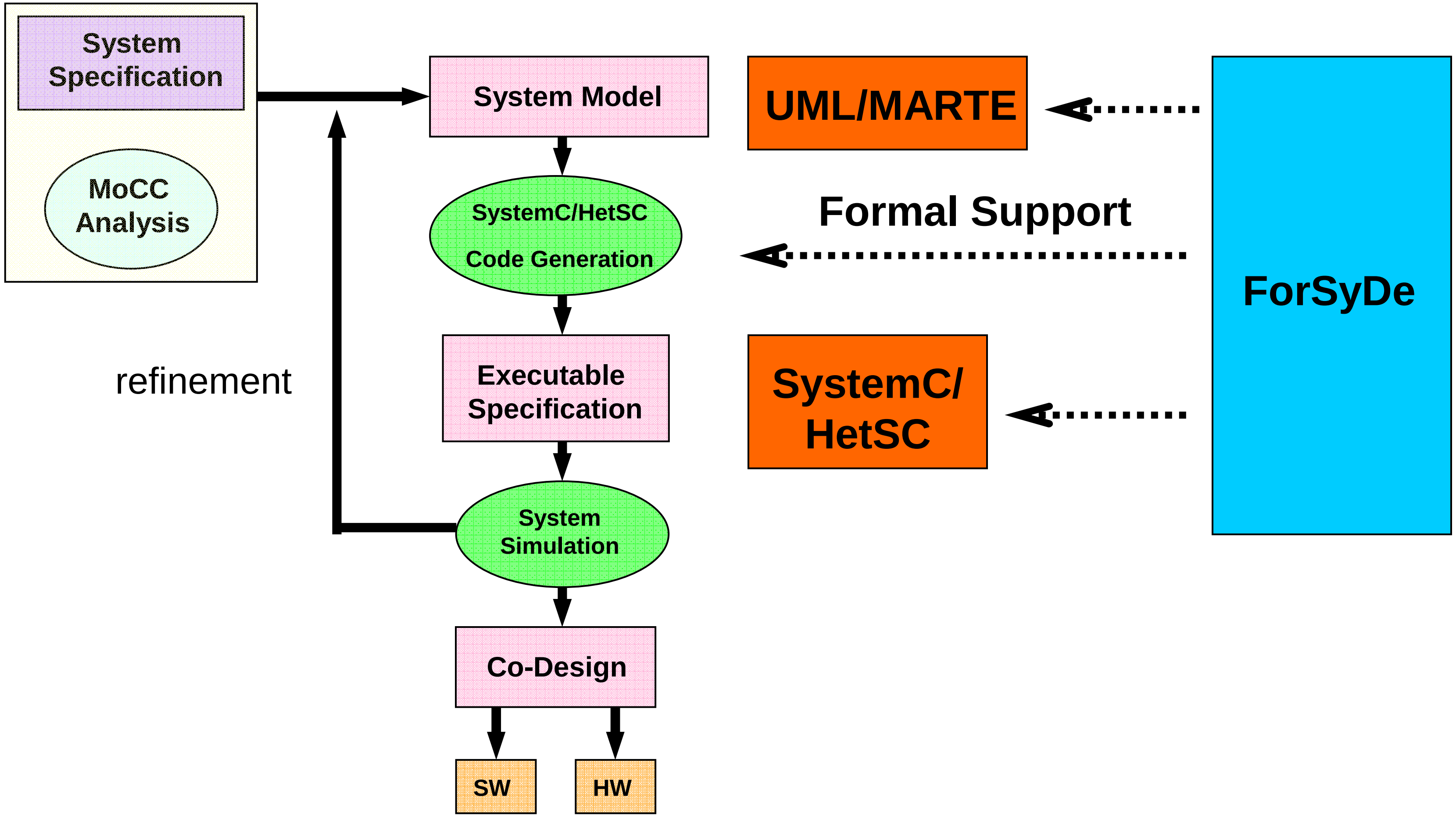
SystemC

Co-Design

Motivation (2)

- **Massive Concurrency**
  - Data Dependencies
  - Relations
- **Characteristic of the interactions**
  - Formal Semantics
  - Univocal Description
- **Models of Computation & Communication (MoCCs)**
  - Behaviors Semantics Heterogeneity

Design Flow Proposed



Formal System Design

- ForSyDe formal metamodel
  - *Process*
  - *Signals*
    - *Separation Communication-Computation*
  - *MoCC generic characteristics*
    - Untimed MoCs
      - No time information
      - Causality (cause and effects)

HetSC

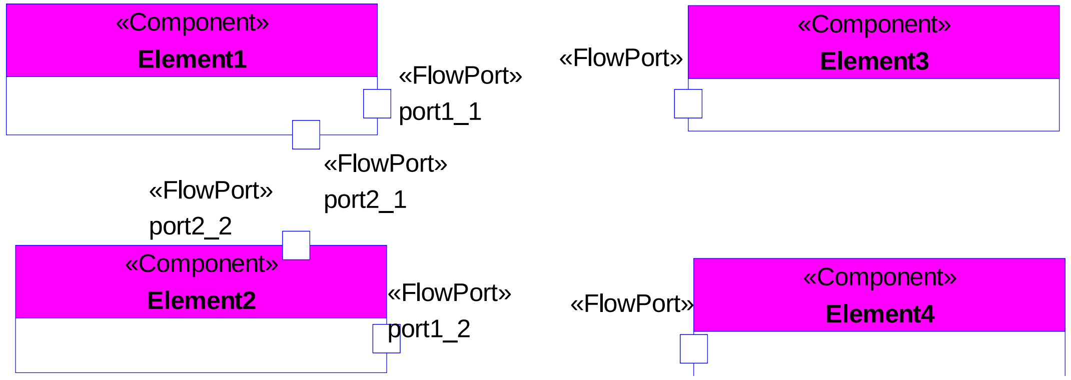
- Methodology for the specification of concurrent **Het**erogeneous embedded systems in **S**ystem**C**
- Clearly Separation between Communication and Computation
- Partitioning Decision not yet taken
- MoCC Semantics

UML/MARTE Methodology

UML/MARTE SystemC Interoperability

- Hierarchy
- Computation
- communication

Hierarchy



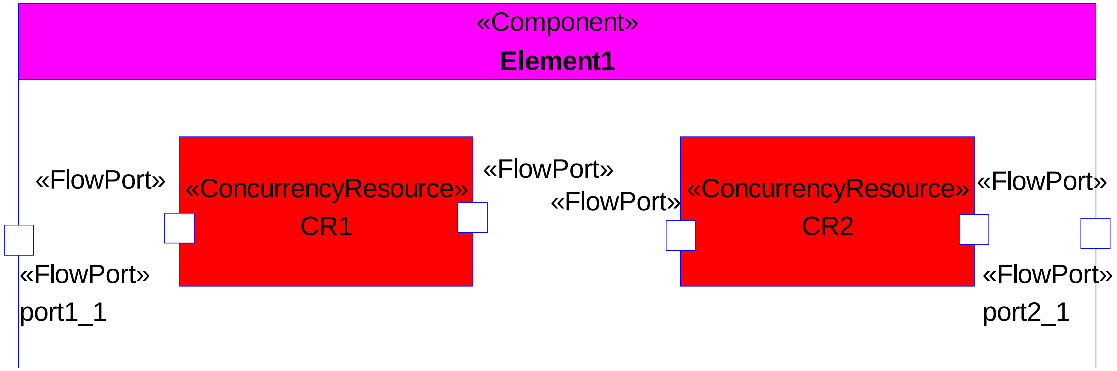
<<Component>> ↓ **sc_module**

FlowPort ↓ **sc_port**

Computation

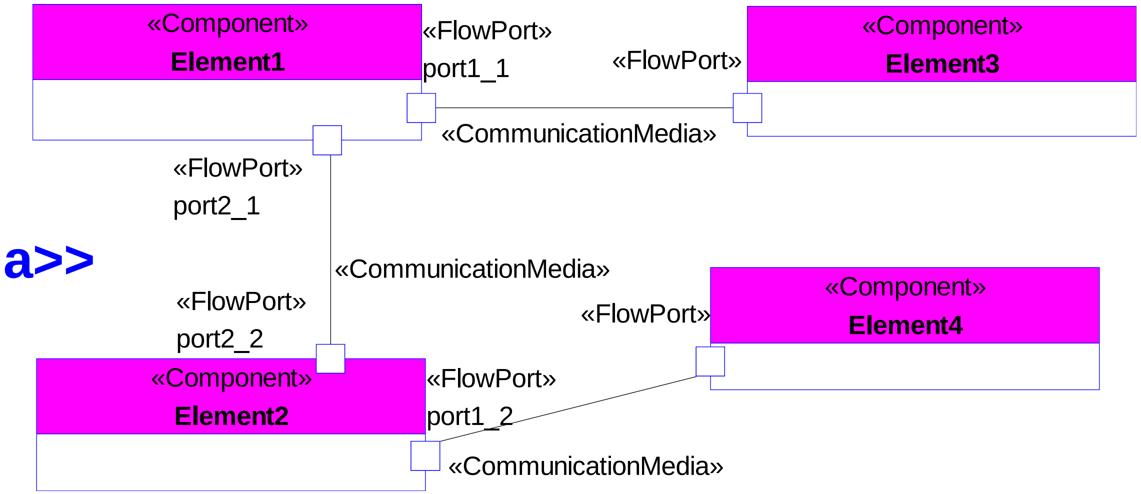
<<ConcurrencyResource>>

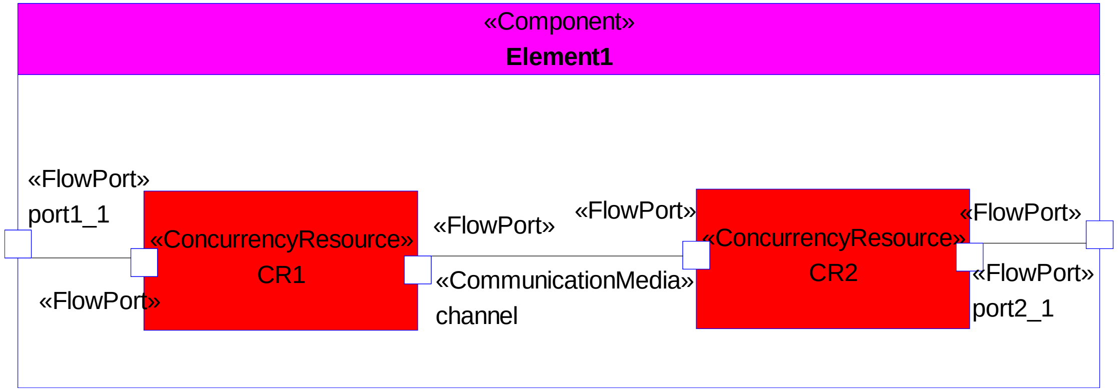
Sc_thread



Communication(1)

<<CommunicationMedia>>



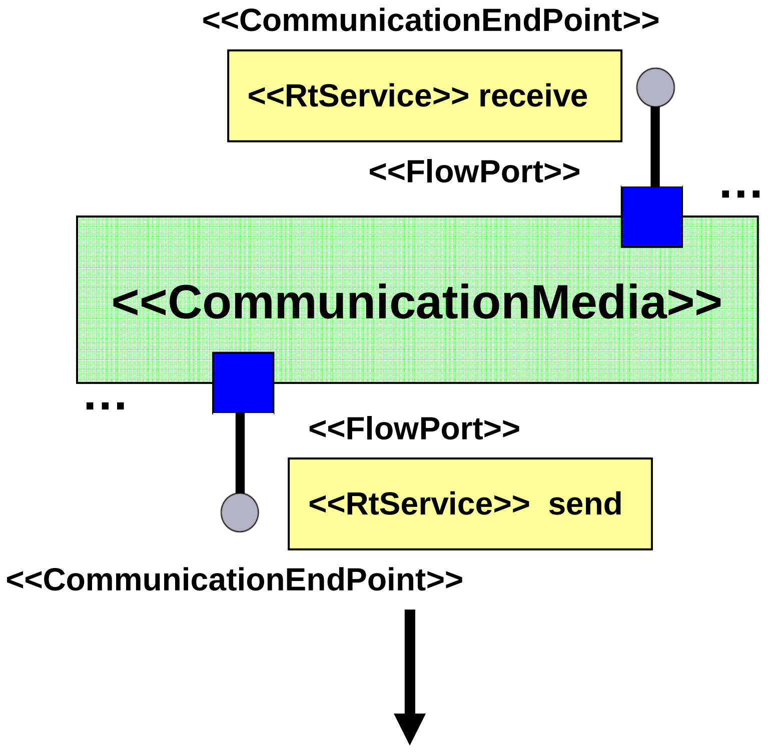


Communication (2)

<<Flowports>>

Provided Interface
<<CommunicationEndPoint>>

Method
<<RtService>>



User Channels

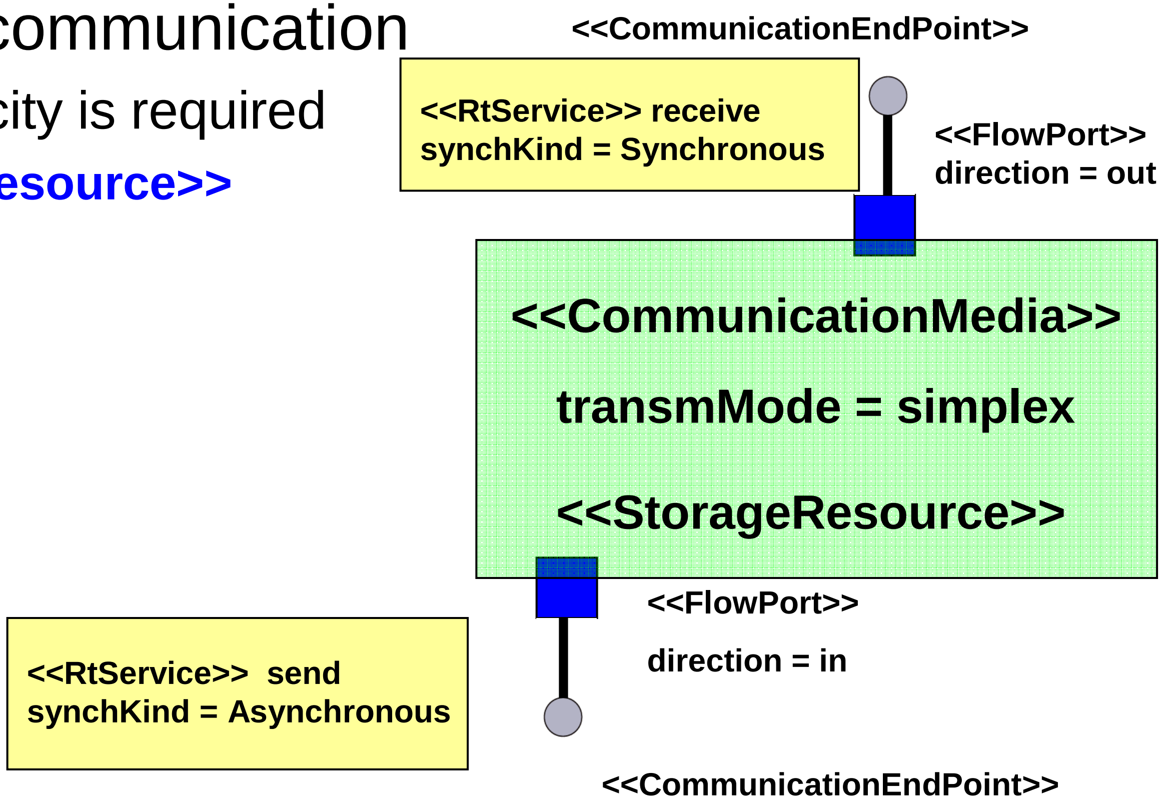
Predefined Channels

SystemC channels

Application Examples

Kahn process network(1)

- Asynchronous communication
  - Buffering capacity is required
    - **<<StorageResource>>**
      - *resMult*



Kahn process network(2)

- HetSC channel:
  - uc_inf_fifo and uc_fifo

```
// module channels
uc_fifo *channel_1;
uc_inf_fifo *channel_1
```

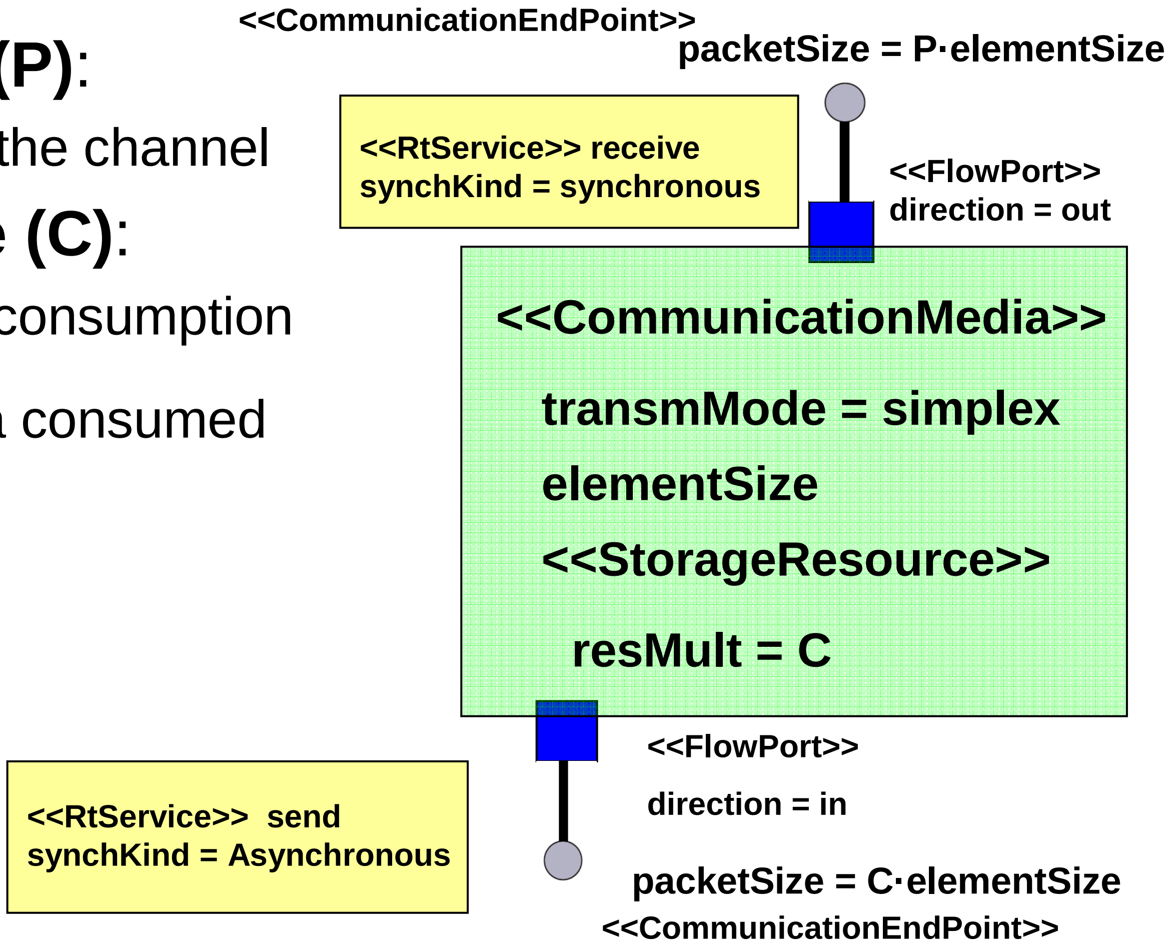
```
// instances channels

channel_1=new uc_fifo<dataType>(“Name, N”)

channel_1=new uc_inf_fifo<dataType>(“Name”)
```

Synchronous Data Flow(1)

- **Producer rate (P)**:
  - Data written in the channel
- **Consumer rate (C)**:
  - Data to trigger consumption
  - Number of data consumed



Synchronous Data Flow(3)

- HetSC channel:
  - uc_arc_seq

uc_arc_seq *channel1;

channel1=new **uc_arc_seq** <dataType, *production_rate*, *consumption_rate*>(“Name”);

Conclusions

- Interoperability between SystemC and the system level modeling with MARTE
  - supported by formal bases
- Generation executable Specifications from UML/MARTE models Identify different MoCCs in MARTE and SystemC
- Automatic Transformation

Future Work

- Synchronous MoC
  - *Synchronous Reactive*
  - *Clocked Synchronous*
- Functionality Description
  - *Activity Diagram* combined *with* Time Modeling and CCSL
- Automatic Generation