DEVS Specification and Implementation of SIMAN Blocks Using Modelica Language

Victorino Sanz, Alfonso Urquia, and Sebastian Dormido
Departamento de Informática y Automática
Escuela Técnica Superior de Ingeniería Informática, UNED
Juan del Rosal 16, Madrid, 28040, SPAIN

ABSTRACT

Modelica is a general object-oriented simulation language mainly based on non-causal modeling with mathematical equations. The aim of our work is to develop a Modelica library, ARENALib, for discrete process-oriented system modeling with comparable functionalities to Arena Basic Process panel. It will provide, combined with the current Modelica components for continuous system modeling, a good tool for modeling hybrid systems. A first version of the library, with basic capabilities, is freely available under GPL license. In this contribution an specification of the Create, Dispose, Queue, Seize, Delay and Release SIMAN blocks using DEVS formalism is presented. The implementation, in Modelica, of SIMANLib library is based on these specifications. Create, Process and Dispose modules of ARENALib have been reimplemented using SIMANLib blocks. A single server system model is also discussed. Future work will consist of the development of more SIMAN blocks to complete ARENALib modules and functionalities

I. DEVS FORMALISM

DEVS (Discrete Event System Specification) is a modeling formalism proposed by B.P. Zeigler [1].
A model in DEVS is specified by the tuple M = {X,S,Y, δ_{int}, δ_{ext}, λ,ta}, where:

- X is the set of input values.
- S is the set of states.
- Y is the set of output values.
- δ_{int} : S -> S is the internal transition function.
- δ_{ext} : Q x X -> S is the external transition function, where
  Q = {(s,e)| s ∈ S, 0 ≤ e ≤ ta(s)} is the total state set and,
  e is the time elapsed since last transition.
- λ : S -> Y is the output function
- ta : S -> $\mathbb{R}^{+}_{0,\infty}$ is the set of positive reals with 0 and ∞.

DEVS models base their communication in sending and receiving events.

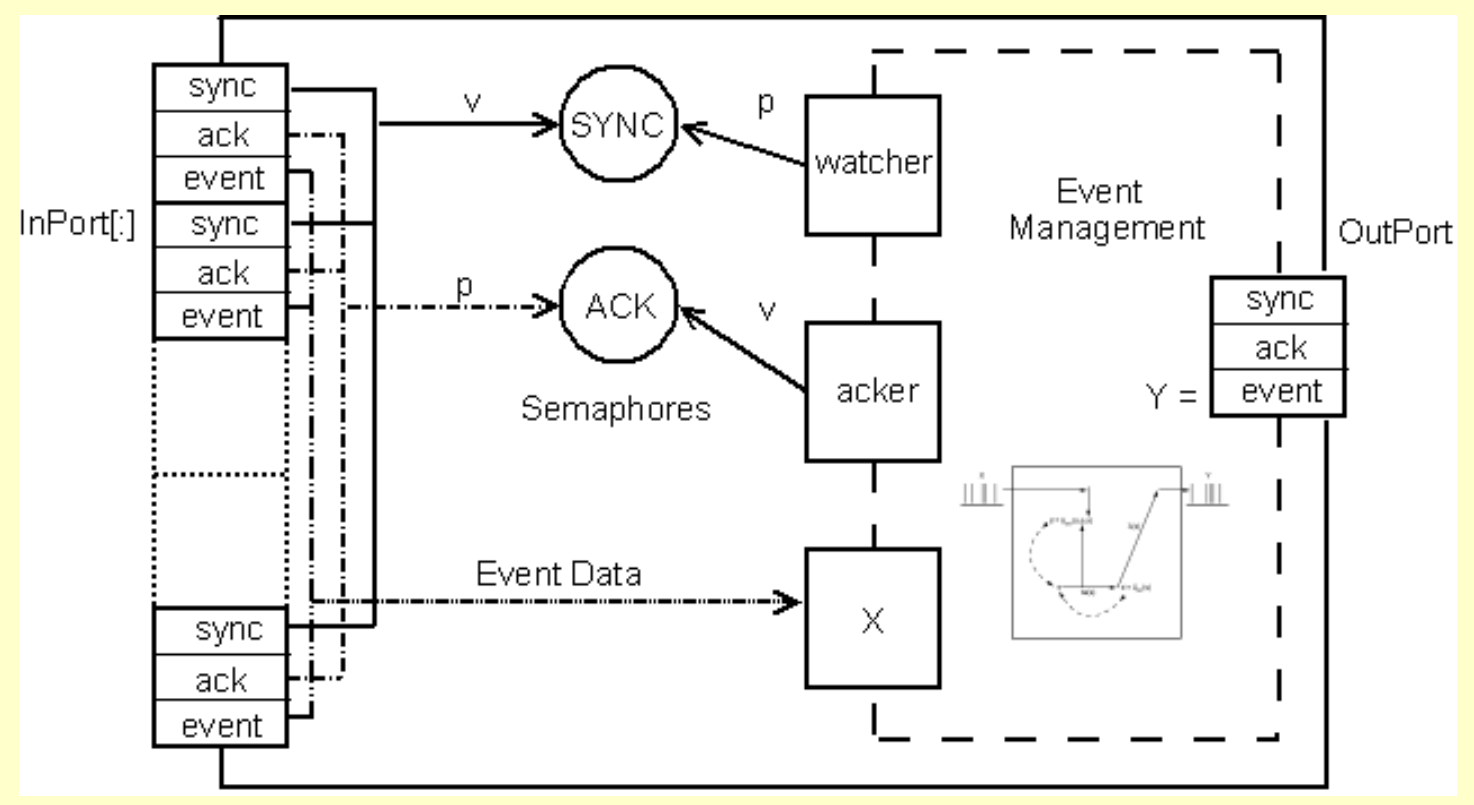
II. DEVS SPECIFICATION OF SIMAN BLOCKS

An specification of the Create, Dispose, Queue, Seize, Delay and Release SIMAN blocks have been described using the DEVS formalism.
This specification represents the management of entities performed by each SIMAN block, as described in [2].
Below an example of specification is shown. It corresponds to the Create block and its parameters are the following: *BatchSize*, *OffSet*, *Interval* and *MaxNumOfBatches*.

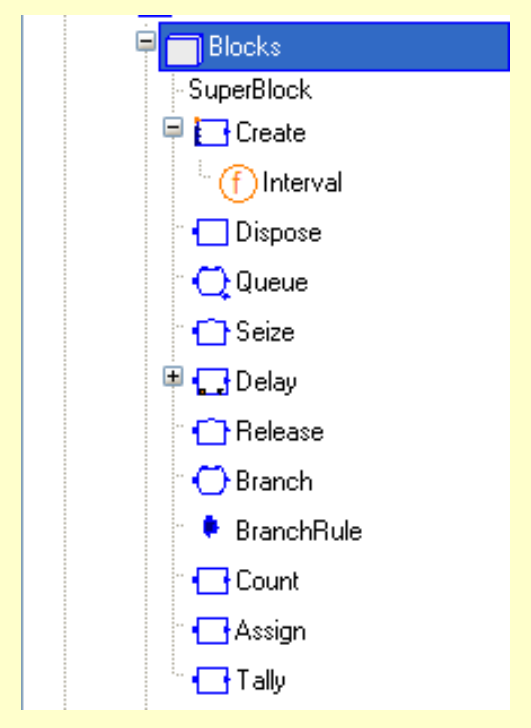
```
X = ∅
S = {start, wait, work, halt} x N⁺ x N⁺
y = N⁺
δint :(start,0,0) = (wait,0,0)
      (wait,n,0) = (work,n,BatchSize)
      (work,n,m) = (work,n+1,m-1) ⇔ n < MaxNumOfBatches
      (work,n,1) = (work,n+1,0) ⇔ n < MaxNumOfBatches
      (work,MaxNumOfBatches,m) = (halt,MaxNumOfBatches,m)
      (halt,n,m) = (halt,n,m)
δext : nothing since X = ∅
λ    :(work,n) = send(sn)
      (halt,n) = ∅
ta   :(start) = OffSet
      (work) = Interval
      (halt) = ∞
```

III. IMPLEMENTATION

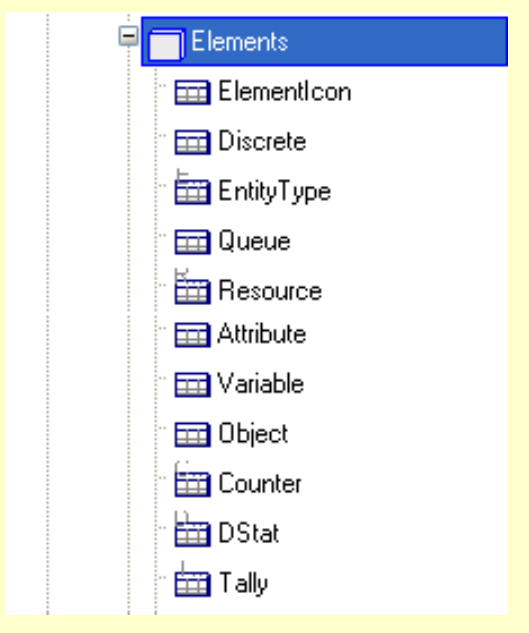
A general atomic DEVS model, called AtomicDEVS, has been implemented in Modelica [3,4].
SIMAN blocks have been implemented using AtomicDEVS model, using the previous specifications and redeclaring external and internal transitions, output and time advance functions.
SIMAN elements have also been implemented and joined to the blocks to develop the SIMANLib Modelica library.



general AtomicDEVS



Blocks



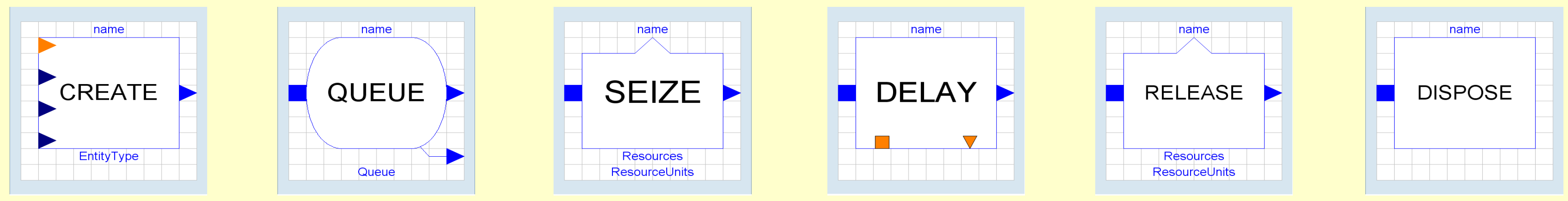
Elements

SIMANLib

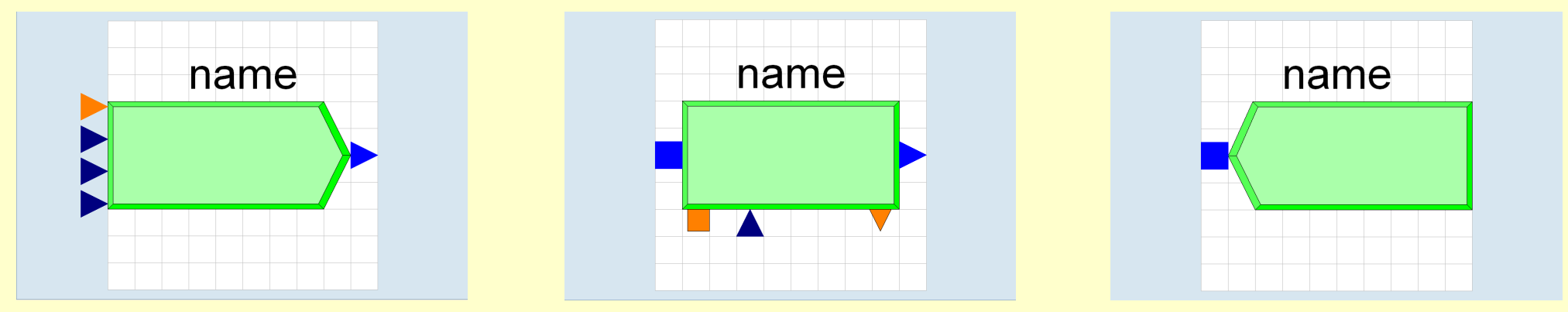
IV. SYSTEM DEVELOPMENT

Modeling and simulation of discrete-event systems using the process-oriented approach can be performed in Modelica using the ARENALib Modelica library [6].

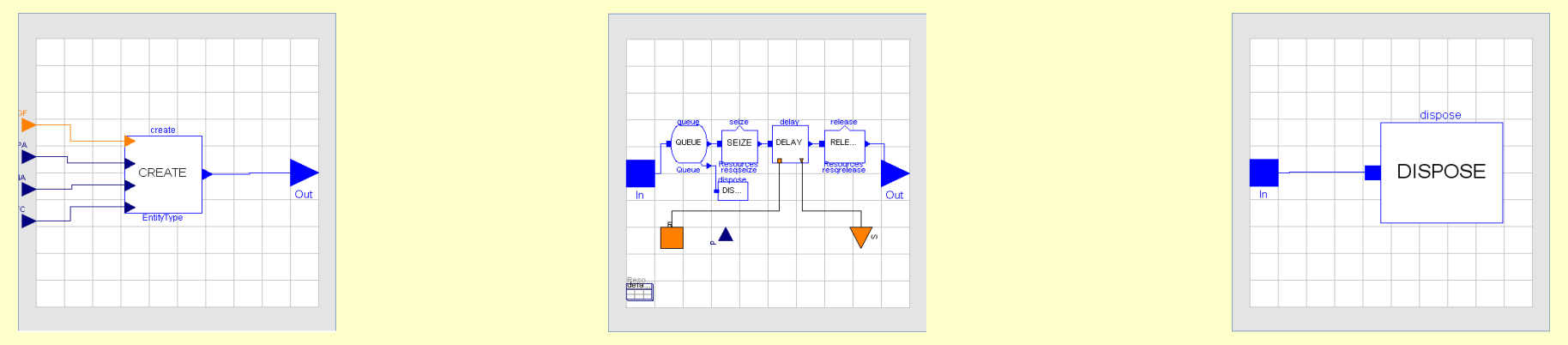
SIMANLib blocks and elements can be used for modeling systems. However, higher level modeling modules, like the ones in the Basic Process panel of Arena, can also be composed using SIMANLib components.
Thus, Create, Queue, Seize, Delay, Release and Dispose blocks have been used to reimplement the Create, Process and Dispose modules of ARENALib.



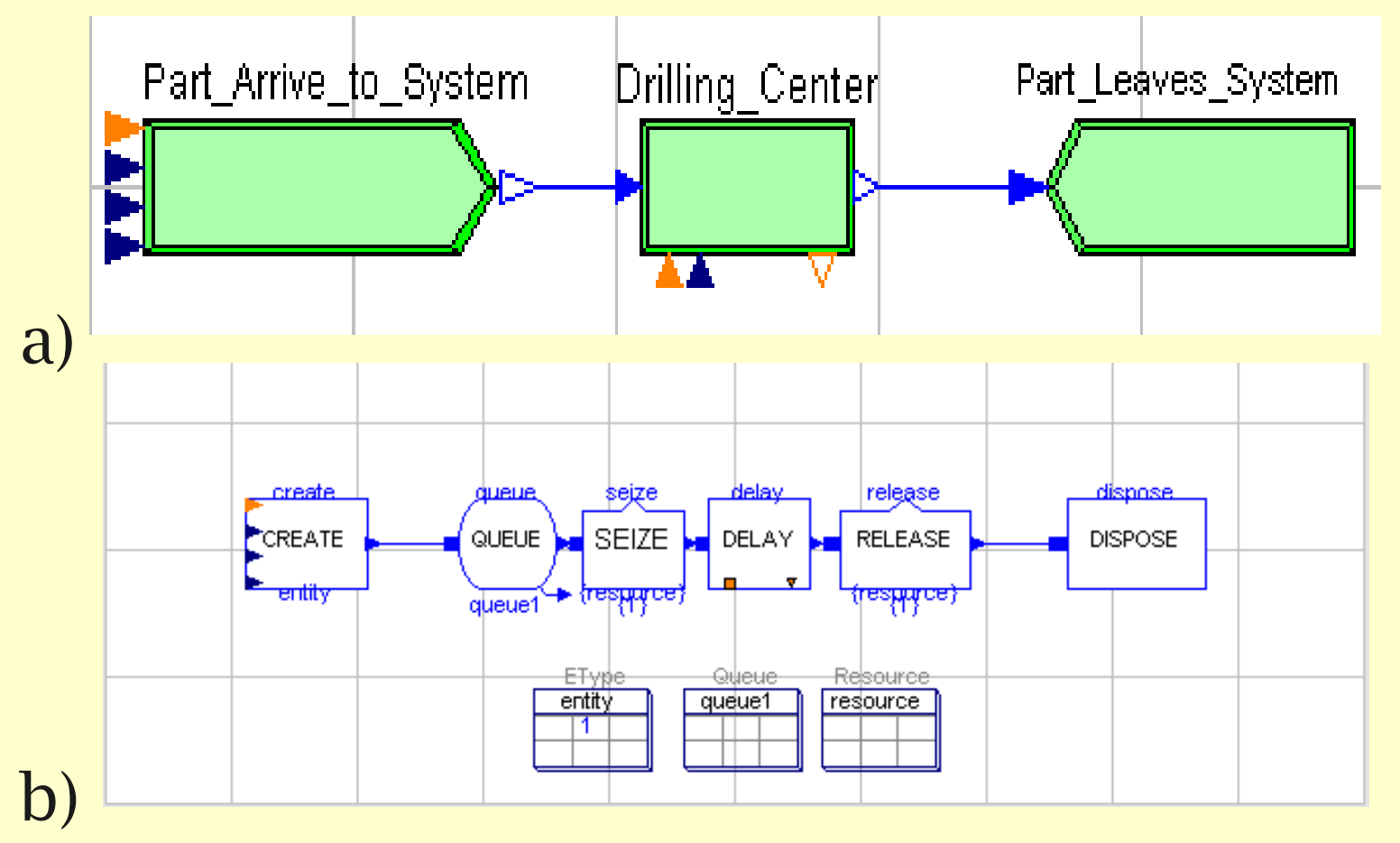
SIMANLib blocks used to reimplement the ARENALib modules.



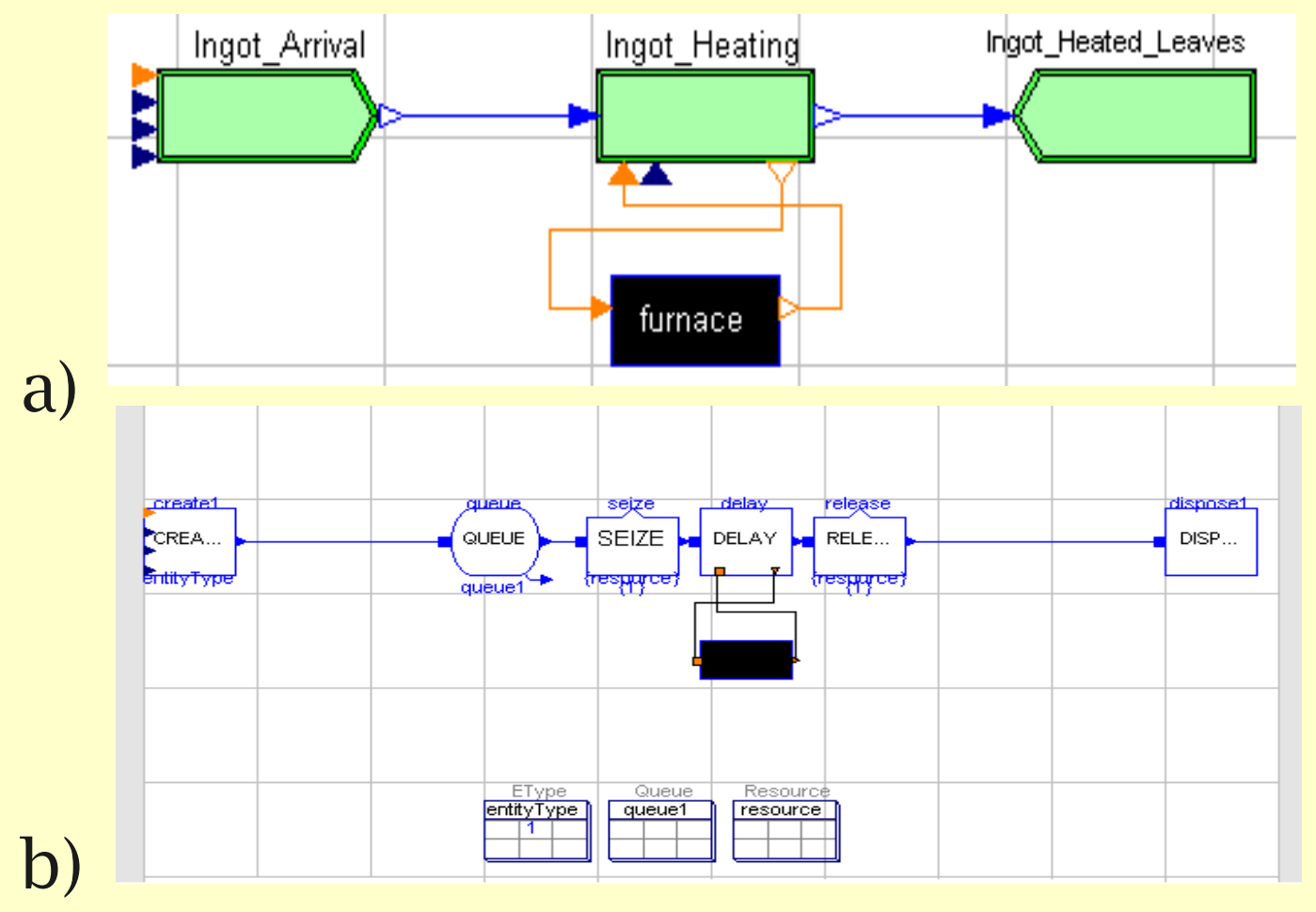
Reimplemented ARENALib modules.



Internal structure of the reimplemented modules.



Dicrete-event model of a drilling center developed using: a) ARENALib; and b) SIMANLib.

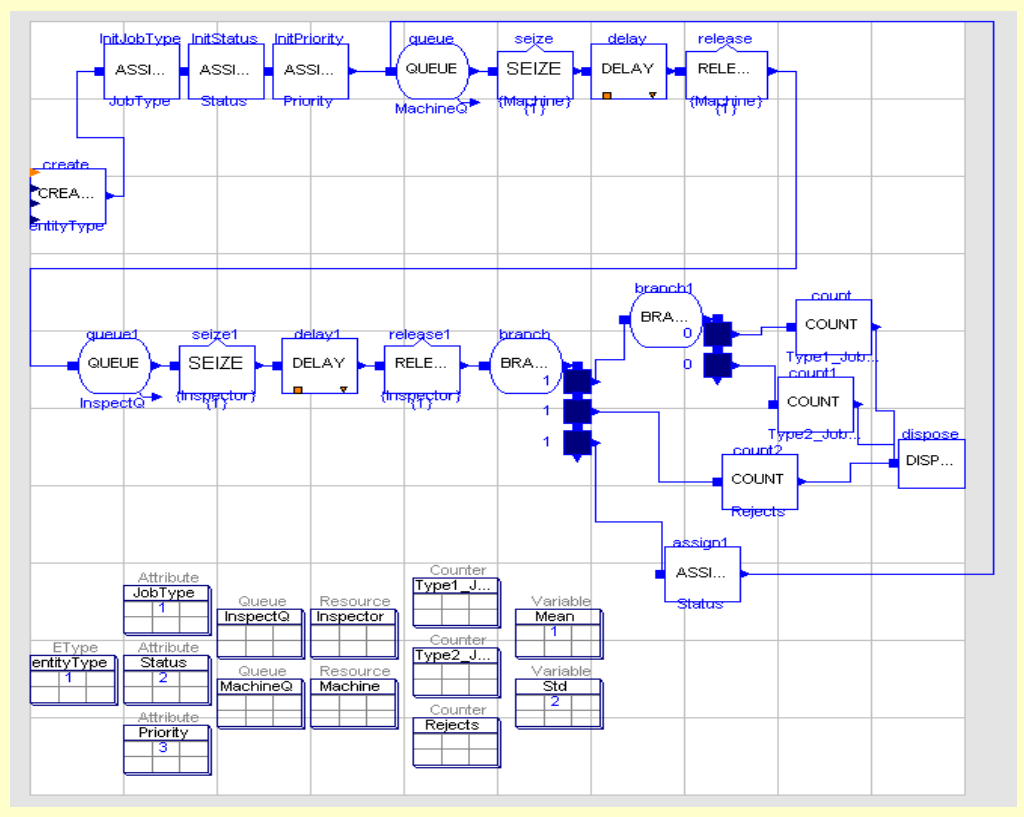


Hybrid model of furnace process developed using: a) ARENALib; and b) SIMANLib. The "Furnace" block encapsulates a continuous time model.

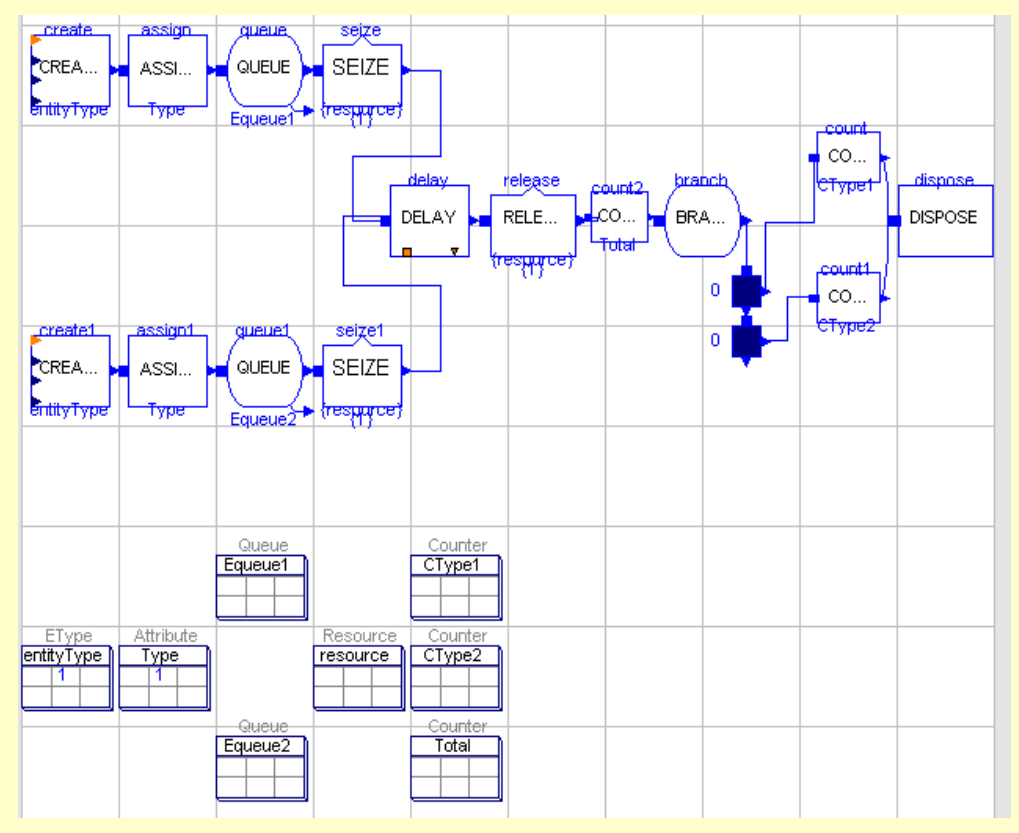
Discrete-event models can be developed using SIMANLib and ARENALib. Hybrid systems can also be developed using these libraries in combination with the already existing Modelica components for modeling continuous time systems.

The Modelica source code for SIMANLib and ARENALib can be found at **http://www.euclides.dia.uned.es**

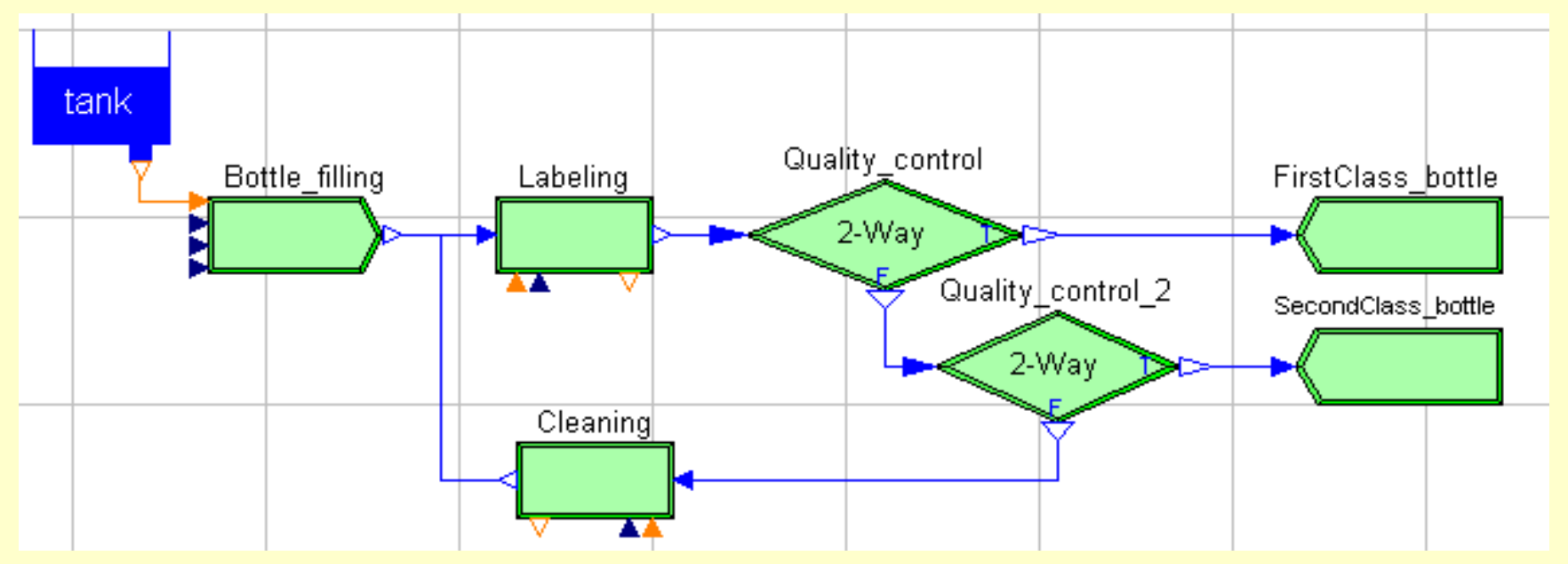
Other examples of systems developed using SIMANLib and ARENALib:



Two workstation model developed with SIMANLib.



Shared workstation model developed with SIMANLib.



Bottle-filling hybrid model developed with ARENALib.

V. CONCLUSIONS

An specification of the Create, Dispose, Queue, Seize, Delay and Release SIMAN blocks has been proposed. This specification follows the DEVS formalism. A general atomic DEVS model, called AtomicDEVS, has been implemented in Modelica. It can be used to develop other DEVS models, by redeclaring its components. Using the given specifications and the AtomicDEVs model, the SIMANLib Modelica library has been developed. It provides similar modeling capacities to its equivalent SIMAN blocks. SIMANLib components have been used to reimplement ARENALib modules. SIMANLib and ARENALib can be used and combined to develop models of discrete-event systems following the process-oriented approach. These libraries can also be used to develop hybrid models, in combination with the already existing Modelica language components. Discrete-event systems, like the DrillingCenter, the TwoWorkstation and the SharedWorkstation, and hybrid systems, like the BottleFilling and the Furnace, have been modeled using SIMANLib, ARENALib and other Modelica models.

REFERENCES

[1] Zeigler, B.P., T.G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. Orlando, FL, USA: Academic Pres, Inc.

[2] Pegden, C.D., R.P. Sadowski, and R.E. Shannon. 1995. *Introduction to Simulation Using SIMAN*. New York, NY, USA: McGraw-Hill, Inc.

[3] Modelica Association web site. http://www.modelica.org

[4] Fritzson, P. 2003. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Computer Society Press.

[5] Sanz, V., A. Urquia, and S. Dormido. 2006. *ARENA*Lib: A Modelica Library for Discrete-Event System Modeling and Simulation. In *Proceedings of the 5th International Modelica Conference*, 539-548.

ACKNOWLEDGMENTS

This work has been supported by the Spanish CICYT, under DPI 2004-01804 grant, and by the IV PRICIT (Plan Regional de Ciencia y Tecnología de la Comunidad de Madrid, 2005-2008), under S-0505/DPI/0391 grant.