# MAST: una experiencia en el modelado de sistemas de tiempo real con herramientas de libre distribución

*Grupo de Ingeniería Software y Tiempo Real, Universidad de Cantabria*
www.istr.unican.es

XXII Jornadas de Tiempo Real
Cercedilla, enero de 2023

Financiado en parte por MCIN/ AEI /10.13039/501100011033/ FEDER “Una manera de hacer Europa” a través del proyecto PID2021-124502OB-C42 (PRESECREL).

# Página web de MAST

UC UNIVERSIDAD DE CANTABRIA

MAST

**Modeling and Analysis Suite for Real-Time Applications**

**Contents**

- Introduction
- News
- MAST model
- Available tools
- Documents
- Downloading

**Key features**

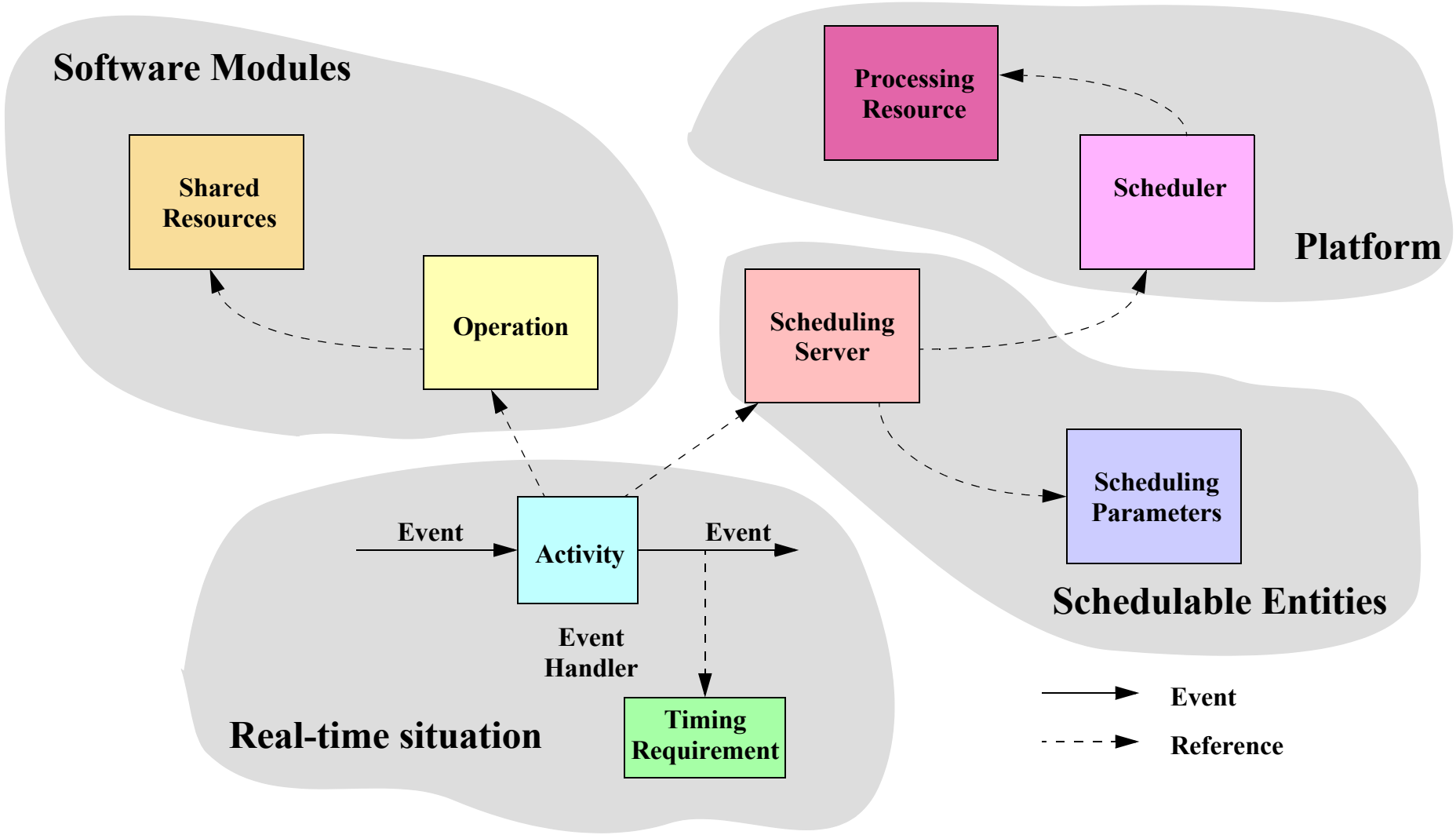
**MAST 1.5.1.0**, incuding multipath transactions and a corrected analysis of polling schedulers, is now available!

## Introduction

**MAST** is an open-source suite of tools to perform schedulability analysis of real-time distributed systems that assesses a rich variety of timing requirements. Via sensitivity analysis, you will know how far or close the system is from meeting those requirements. **MAST** uses a versatile and composable input model for the real-time behavior of the modules and platforms that form your system.

MAST is distributed under a GPL free software license. For commercial inquiries related to other licenses please contact us.

# El modelo


Software Modules
Shared Resources
Operation
Processing Resource
Scheduler
Platform
Scheduling Server
Scheduling Parameters
Schedulable Entities
Event
Activity
Event
Event Handler
Timing Requirement
Real-time situation
Event
Reference

# Publicaciones del modelo MAST

- **MAST: Modeling and Analysis Suite for Real Time Applications.**
  M. González Harbour, J.J. Gutiérrez García, J.C. Palencia Gutiérrez, and J.M. Drake Moyano.
  Proceedings of 13th Euromicro Conference on Real-Time Systems, Delft, The Netherlands, IEEE Computer Society Press, pp. 125-134, June 2001.
- **Modeling distributed real-time systems with MAST 2.**
  Michael González Harbour, J. Javier Gutiérrez, José M. Drake, Patricia López Martínez and J. Carlos Palencia.
  Journal of Systems Architecture, Vol 56(6), pp. 331-340, Elsevier, June 2013.
  - añade particionado tipo ARINC (procesadores y redes)
  - se alinea con el estándar MARTE de la OMG

# MAST: el origen

Dos presentaciones en:

- Jornadas de Automática 1996 (San Sebastián)
  - Grupo de Tiempo Real
- III Jornadas de Tiempo Real 2000 (Santander)

# Implementación de Sistemas Distribuidos de Tiempo Real Estricto en Ada 95

J. Javier Gutiérrez García (gutierjj@ctr.unican.es)
J. Carlos Palencia Gutiérrez (palencij@ctr.unican.es)
Michael González Harbour (mgh@ctr.unican.es)

Departamento de Electrónica y Computadores, Universidad de Cantabria

**III Jornada de Sistemas de Tiempo Real**
**Santander, Septiembre de 1996**

# Antecedentes

- El anexo de sistemas distribuidos (E) de Ada 95 no contempla tiempo real.
  - Construcción de particiones a partir de un programa no distribuido.
  - Declaración explícita de los módulos distribuidos.
  - Configuración en tiempo de compilación.

- Anexo implementado en GNAT (Sparc/Solaris, PC/Linux).
  - Servidor de nombres centralizado (no adecuado para tiempo real).

- Ada presenta buenas cualidades para programar sistemas distribuidos de tiempo real:
  - Implementación de redes de comunicación de tiempo real en Ada 83 con extensiones, basadas en las colas de mensajes del POSIX.
  - Facilidad para desarrollar aplicaciones que sean modelables y analizables.

# Objetivos de Nuestro Trabajo Futuro

**Implementación de sistemas distribuidos de tiempo real estricto:**

- Utilizar las colas de mensajes del POSIX para implementar las comunicaciones.

- Dos posibles implementaciones:

  - Estándar Ada 95:
    Colas de mensajes en el *Partition Communication Subsystem*.

  - Estándar POSIX:
    Definición de un prototipo Ada 95 que utilice las colas de mensajes.

- Entorno integrado para el desarrollo de aplicaciones distribuidas de tiempo real estricto.

# Implementación Bajo el Estándar Ada 95

- Implementación modelable y analizable.
- Evaluar si el anexo de sistemas distribuidos es adecuado para tiempo real:
  - Cumplimiento de las especificaciones.
  - Restricciones.
- Evaluar la eficiencia de la implementación del *Partition Communication Subsystem* con las colas de mensajes:
  - Overhead de una nueva capa software.
  - Adaptación a las especificaciones.

# Implementación Bajo el Estándar POSIX

- Colas de mensajes base de las comunicaciones en el sistema distribuido.

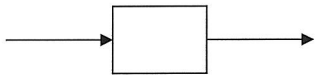
- Sencillez y eficiencia: las tareas de la aplicación utilizan directamente la interfase de colas de mensajes.

- Prototipo de sistema distribuido:
  - Modelo de representación del sistema distribuido:
    - Extensivo.
    - Soportar la utilización de colas de mensajes.
    - Análisis de planificabilidad.
    - Planificación (expulsora por prioridades fijas, servidores esporádicos).
  - Plantillas Ada 95 adaptadas al modelo del sistema.

- Herramienta de generación automática de código.

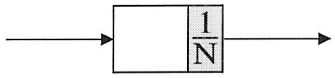
# Modelo de Sistema Distribuido

**Acción Lineal**

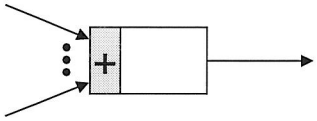


**Divisor de Frecuencia de Entrada**

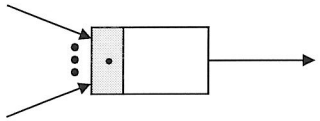
**Divisor de Frecuencia de Salida**



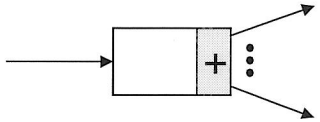

**Or de Entrada**



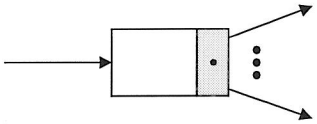

**And de Entrada**



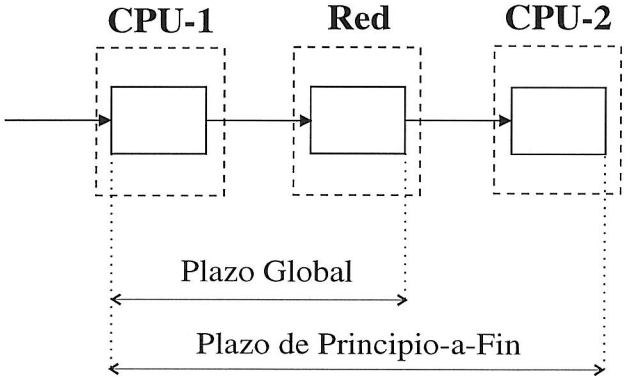

**Or de Salida**




**And de Salida**

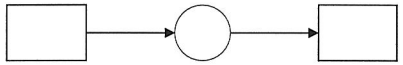


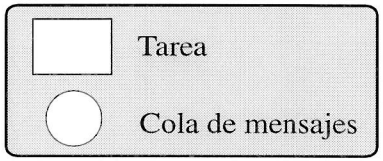

**Respuesta Lineal a un Evento**



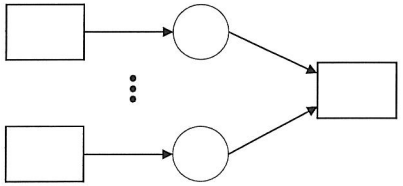
# Modelo de Colas de Mensajes

**Tarea lineal y divisor de frecuencia**

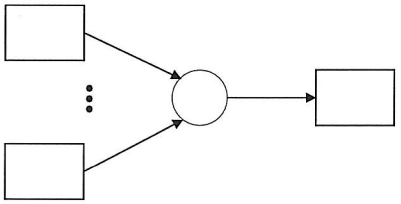




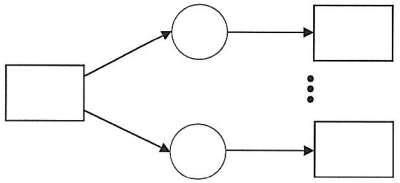

**Tarea que lee de varias colas (and de entrada)**



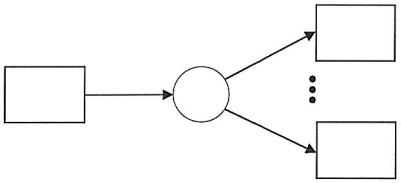
**Varias tareas que escriben en la misma cola (or de entrada)**



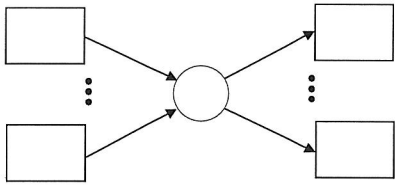
**Tarea que escribe en varias colas (and de salida)**



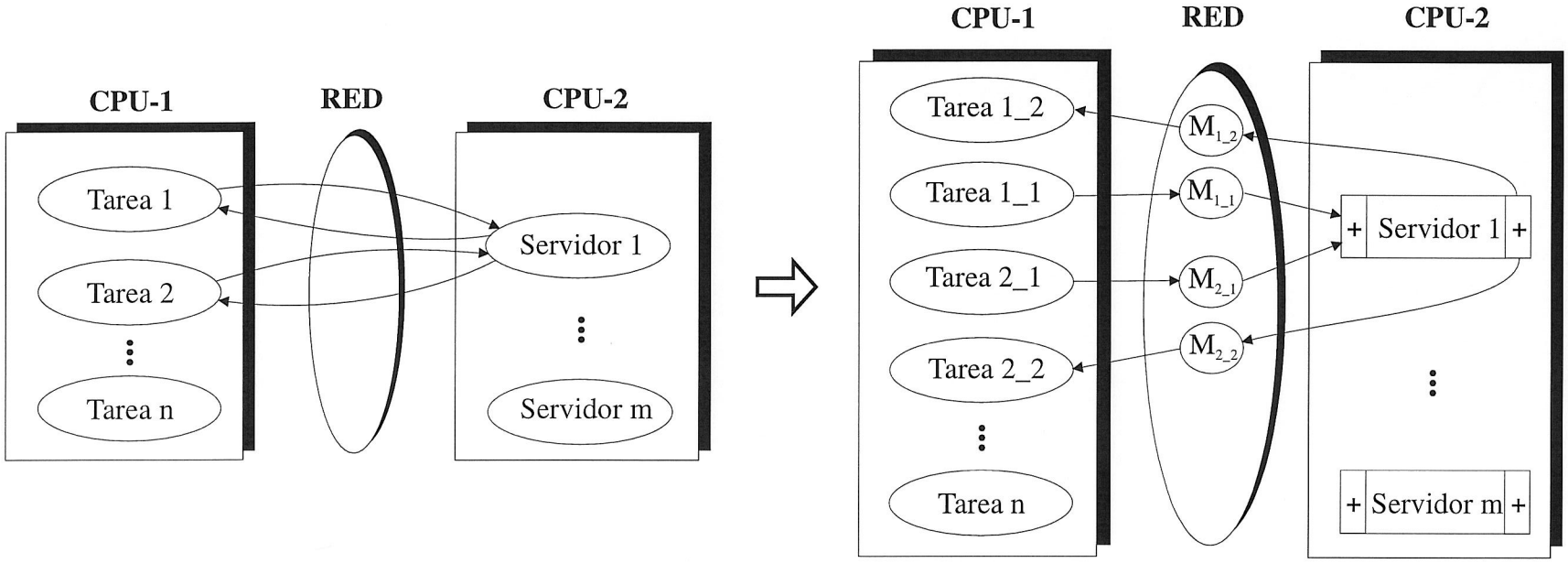
**Varias tareas que leen de la misma cola (or de salida)**



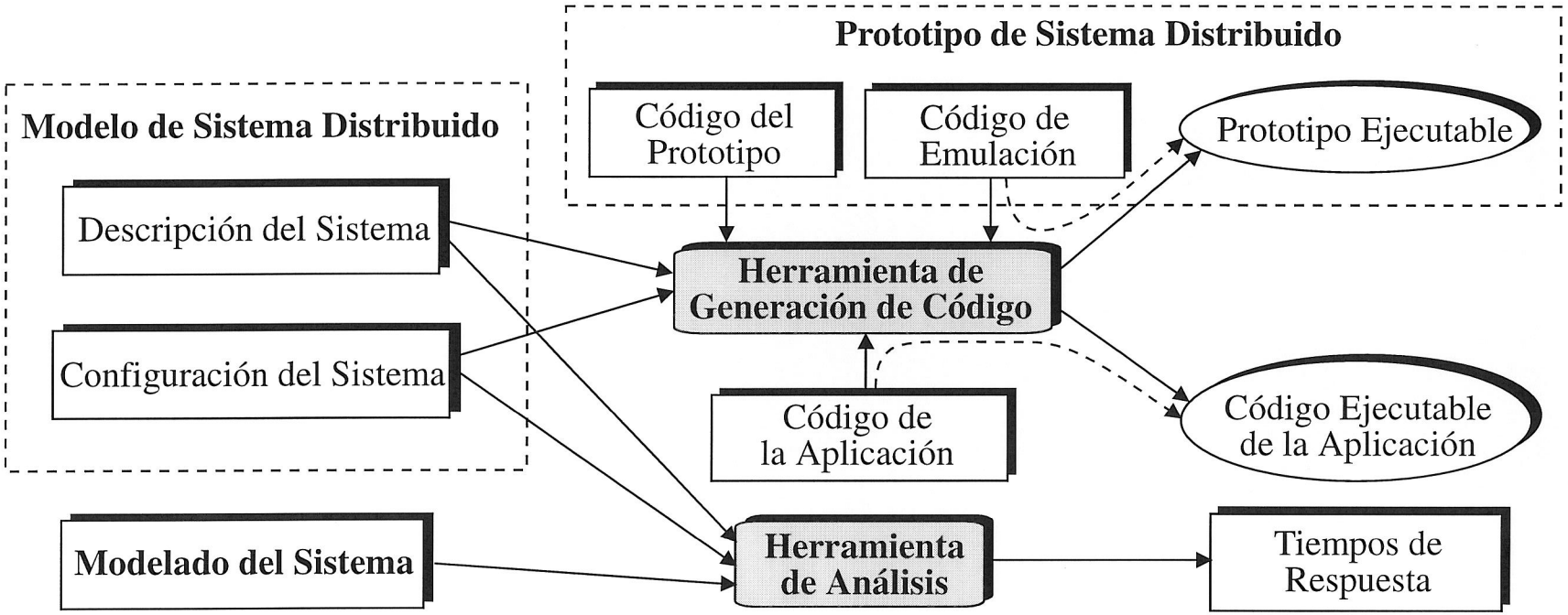
**Varias tareas que escriben y varias que leen de la misma cola (or de entrada y or de salida)**



# Modelo Cliente-Servidor



# Entorno Integrado de Desarrollo



UNIVERSIDAD DE CANTABRIA

# Modelo abierto de sistema de tiempo real para herramientas automáticas

**J. Javier Gutiérrez García (gutierjj@ctr.unican.es)**

**Michael González Harbour (mgh@ctr.unican.es)**

**Departamento de Electrónica y Computadores, Universidad de Cantabria**

**3as Jornadas de Tiempo Real**

**Santander, 10-11 de Febrero de 2000**

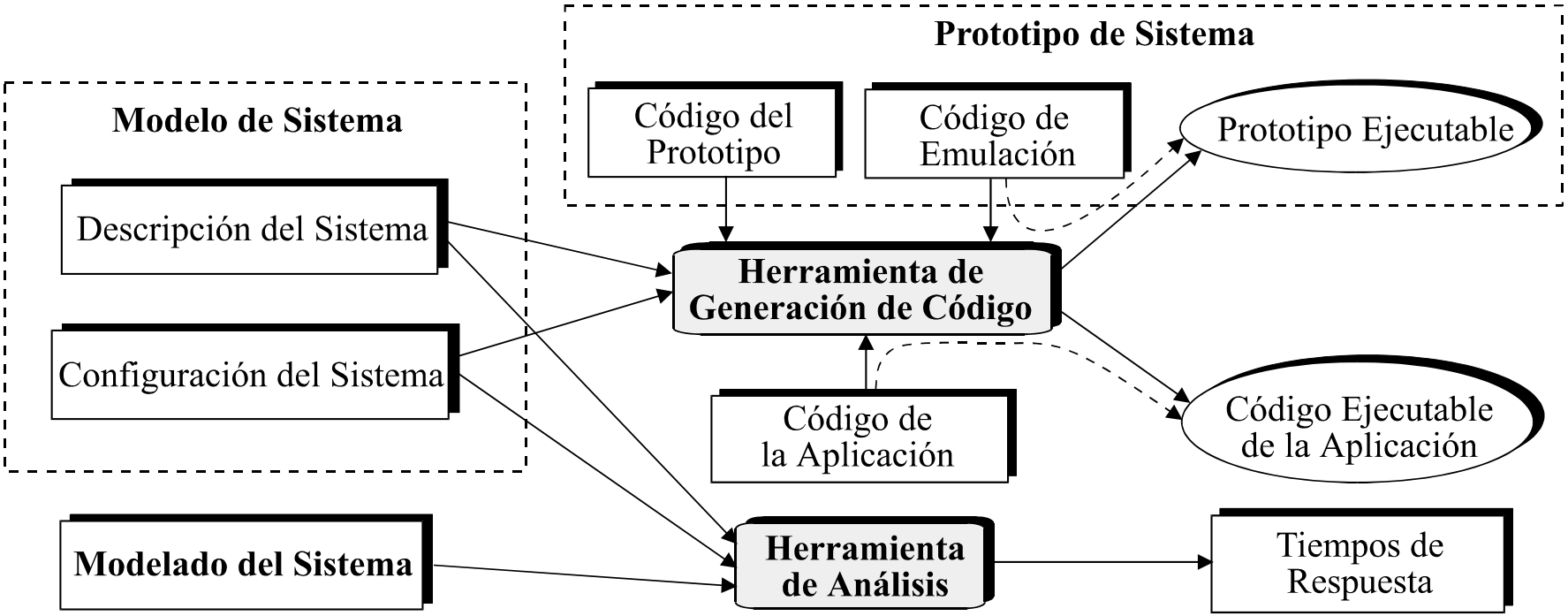
# Modelo descriptivo de sistema de tiempo real

UNIVERSIDAD DE CANTABRIA

1. Motivación
2. Objetivos
3. Características generales del sistema de tiempo real
4. Características particulares del modelo
5. Fichero de especificación
6. Ejemplo

# 1. Motivación

UNIVERSIDAD DE CANTABRIA



# 2. Objetivos

UNIVERSIDAD DE CANTABRIA

- Desarrollar un modelo de descripción de sistemas de tiempo real gobernados por eventos
- *Modelo abierto* para que pueda incorporar nuevas características o puntos de vista sobre el sistema
- Implementar el modelo:
  - Fichero de especificación
    - Descripción sencilla y extensible del sistema
  - Diseñar la estructura de datos (Ada 95)
    - Estructura de datos soporte del modelo
  - Implementación de la herramienta:
    - Fichero de especificación --> Estructura de datos
    - Estructura de datos --> Fichero de especificación

# 3. Características generales del sistema de tiempo real

UNIVERSIDAD DE CANTABRIA

Recursos de procesado:
- Procesadores
- Redes de comunicaciones

Políticas de planificación

Recursos de datos:
- Política de acceso

**Sistema de tiempo real**

Eventos

Tareas/Procesos/Trabajos

Requisitos temporales

# 4. Características del modelo

UNIVERSIDAD DE CANTABRIA

**Elementos del sistema:**

- **Recursos:**
  - **Recursos de procesado (procesadores y redes)**
  - **Recursos de datos**
- **Acciones (procedimientos, funciones, mensajes,...)**
- **Eventos (interrupciones, timers,...)**
- **Transacciones:**
  - **Combinaciones de acciones con requisitos temporales**
- **Servidores (tareas, procesos, threads,...)**

# Recursos

UNIVERSIDAD DE CANTABRIA

**Todos tienen nombre**

**Recursos de procesado:**

- **CPU de prioridad fija**
  - **Tiempos de cambio de contexto (mejor, peor, promedio), overhead y período del ticker**
- **Red de prioridad fija**
  - **Overhead de envío y recepción (mejor, peor, promedio), y tiempo de bloqueo de un paquete**

**Recursos de datos:**

- **Techo de prioridad (techo)**
- **Herencia de prioridad**

# Servidores

UNIVERSIDAD DE CANTABRIA

**Nombre**

**Parámetros de planificación:**

- **Prioridades fijas (prioridad)**
- **Polling (prioridad y período)**
- **Servidor esporádico (prioridades normal y background, período de relleno y máximo número de rellenos pendientes)**

**Recurso de procesado**

# Acciones

UNIVERSIDAD DE CANTABRIA

**Todas tienen nombre**

**Simple:**

- **Parámetros de planificación para modificar los del servidor**
- **Tiempos de ejecución peor, mejor y promedio**
- **Lista de recursos de datos**

**Compuesta:**

- **Formada por acciones simples**

# Eventos

UNIVERSIDAD DE CANTABRIA

**Periódico:**

- **período, máximo jitter, y fase**

**Esporádico:**

- **tiempo promedio de llegada, función de distribución, mínimo intervalo entre llegadas**

**Ilimitado:**

- **tiempo promedio de llegada, y función de distribución**

**A ráfagas:**

- **tiempo promedio de llegada, función de distribución, intervalo de tiempo y máximo número de llegadas**

# Transacciones

UNIVERSIDAD DE CANTABRIA

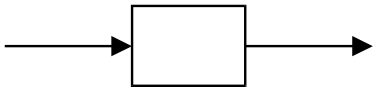
**Se describen a través de grafos:**

- **Lista de arcos**
  - **Normal (nombre)**
  - **Entrada (nombre y evento)**
  - **Salida (nombre)**
- **Lista de nudos**
  - **Contiene una acción (acción, arcos de entrada y salida, servidor, y requisitos temporales)**
  - **Estructurales (arco y/o arcos de entrada y/o salida)**
    - **Input_1_of_N, Input_N_of_N, Output_1_of_N, Output_N_of_N, Rate_Divisor, Delay, y Offset**

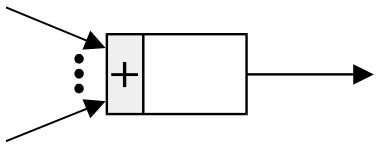
# Transacciones (cont.)

UNIVERSIDAD DE CANTABRIA

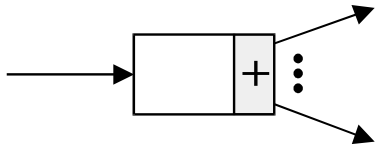
**Action / Rate Divisor / Delay / Offset**



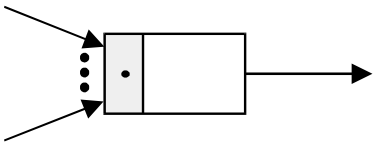
**Input 1-of-N**



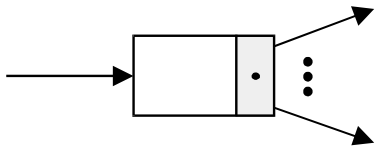

**Output 1-of-N**




**Input N-of-N**




**Output N-of-N**



# 5. Fichero de especificación

UNIVERSIDAD DE CANTABRIA

**Reglas de especificación:**

- **Formato de objeto: Nombre_de_objeto (Argumentos);**
- **Los objetos tienen *tipo* y/o *nombre* (obligatorios)**
- **No se consideran espacios, tabuladores y retornos de carro**
- **Nombres como Ada: letra+(letra/número/subrayado)**
- **Nombres con o sin comillas (obligatorias para palabras reservadas)**
- **Los nombres referenciados deben aparecer previamente**
- **Comentarios como en Ada: "--"**
- **La descripción no es sensible a las mayúsculas**

# Fichero de especificación (cont.)

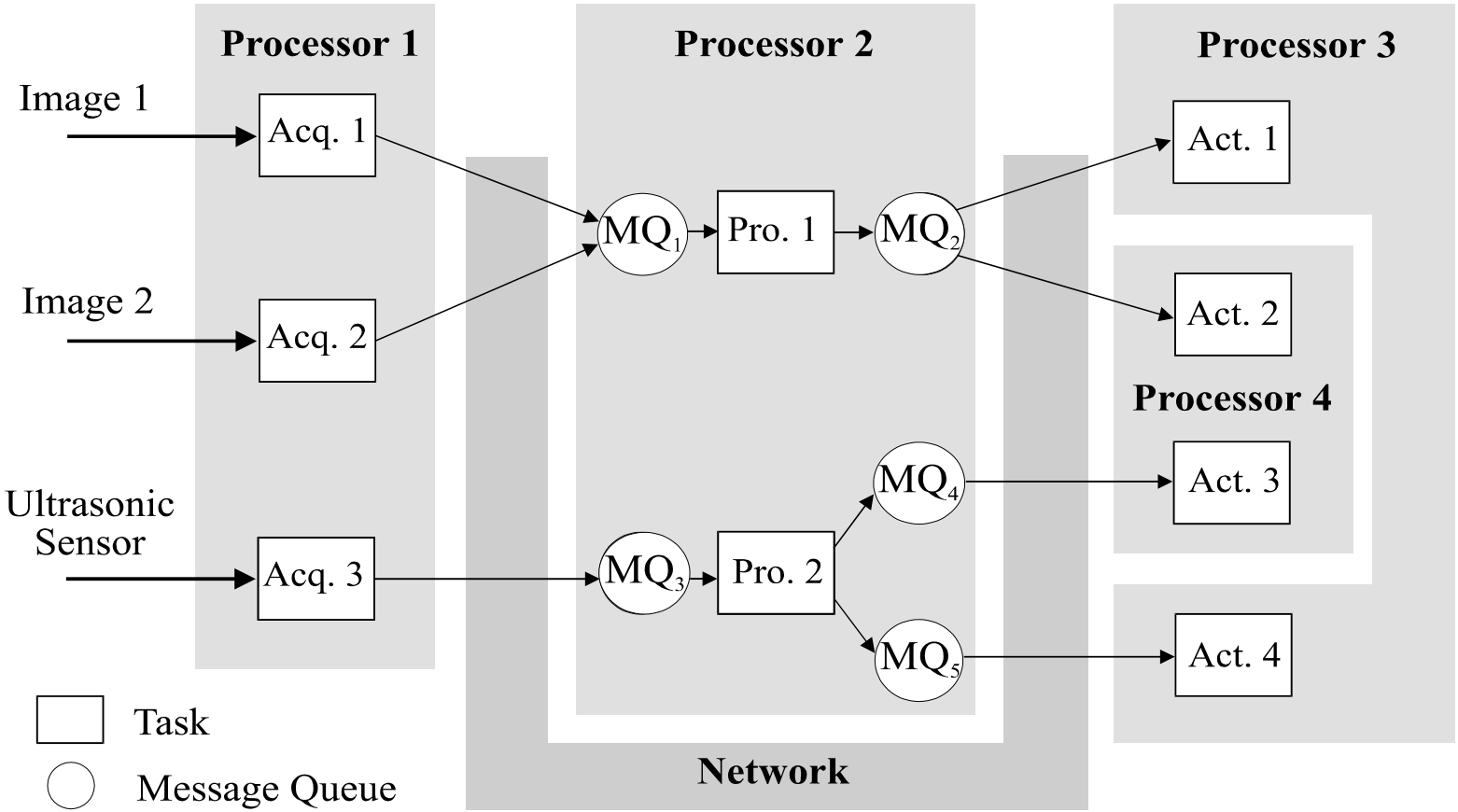
UNIVERSIDAD DE CANTABRIA

```
Action (
   Type                          => Simple,
   Name                          => Name of the action,
   Overrides_Sched_Parameters => Yes/No,
   New_Sched_Parameters          => Scheduling_Parameters,
   Worst_Case_Execution_Time  => WCET,
   Avg_Case_Execution_Time    => ACET,
   Best_Case_Execution_Time   => BCET,
   Data_Resources_List        => (
                                 Data Resource Name 1,
                                 Data Resource Name 2,
                                 ...));


Scheduling_Parameters = (
   Type          => Fixed_Priority_Scheduler,
   The_Priority => Priority)
```

# 6. Ejemplo

UNIVERSIDAD DE CANTABRIA



# Componentes del ejemplo

UNIVERSIDAD DE CANTABRIA

5 recursos de procesado:

- 4 CPUs
- 1 red)

No hay datos compartidos

16 acciones:

- 9 procedimientos
- 7 mensajes

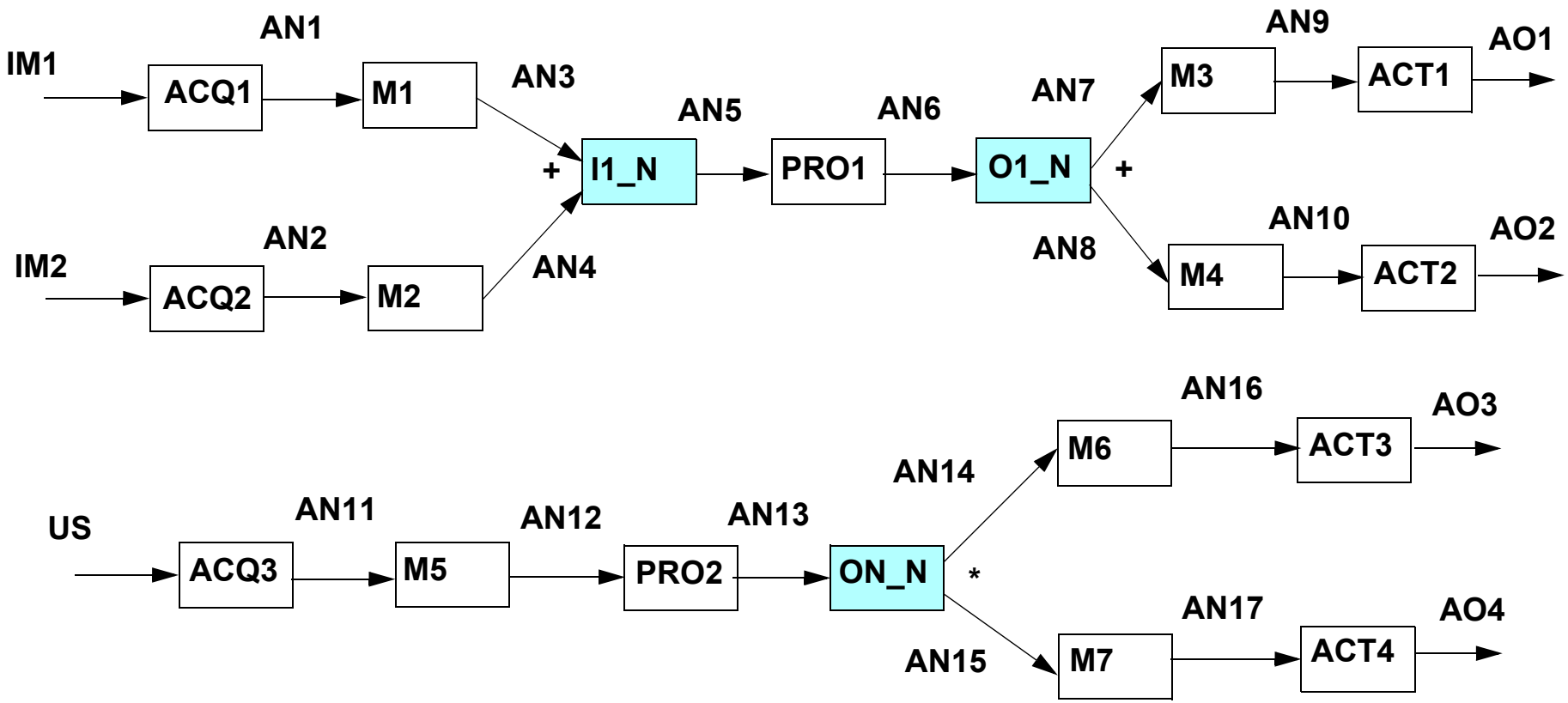
3 eventos

2 transacciones

16 servidores (en este caso uno por acción)

# Grafo para el ejemplo

UNIVERSIDAD DE CANTABRIA



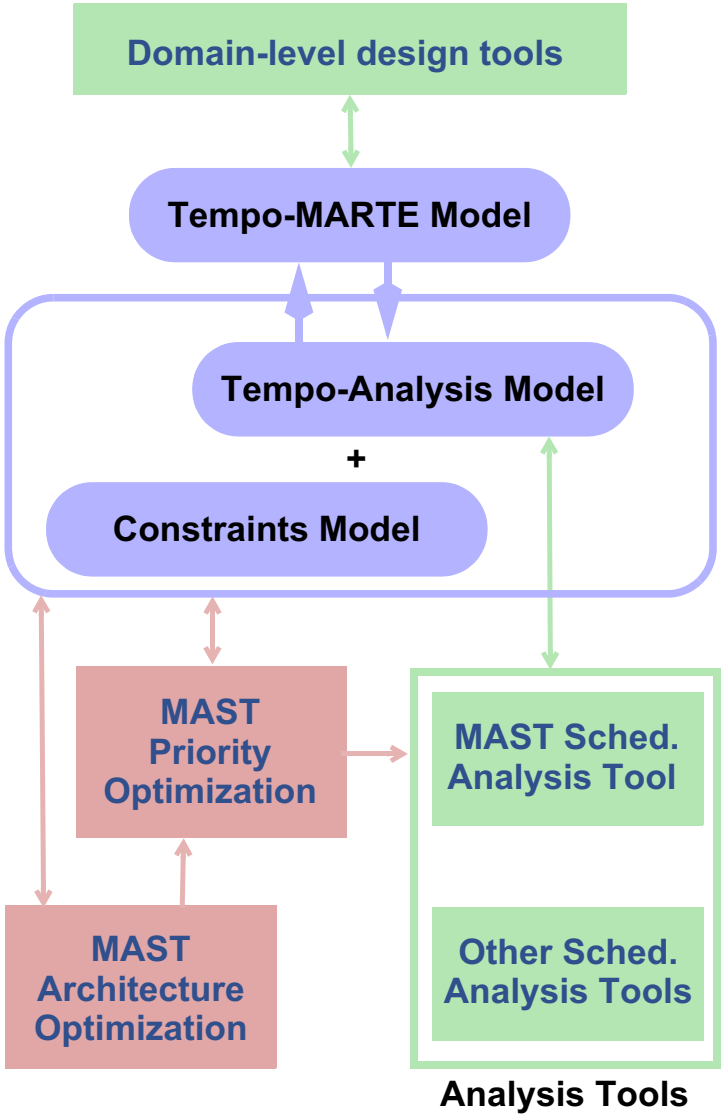
# Fichero para el ejemplo

UNIVERSIDAD DE CANTABRIA

**Entradas del ejemplo:**

- 5 Recursos de procesado
- 0 Recursos de datos
- 16 Acciones
- 3 Eventos
- 16 Servidores
- 2 Transacciones
  - 1 --> 14 arcos: 2 entrada, 10 normales, 2 salida
  - 11 nudos: 9 son acciones
  - 2 --> 10 arcos: 1 entrada, 7 normales, 2 salida
  - 8 nudos: 7 son acciones

# MAST como parte de MDE

- **Schedulability analysis and optimization in a model-based integrated tool-chain: Synthetic MARTE models for optimizing real-time design with MAST and TEMPO.**
  Laurent Rioux, Rafik Henia, Nicolas Sordon, Michael González Harbour, J. Javier Gutiérrez, Juan M. Rivas, César Cuevas, José M. Drake, and Julio L. Medina.
  The 2015 Conference on Forum on specification & Design Languages, FDL 2015 , Barcelona (Spain), Demo Night Session, September 2015.

# MAST como parte de MDE (cont.)