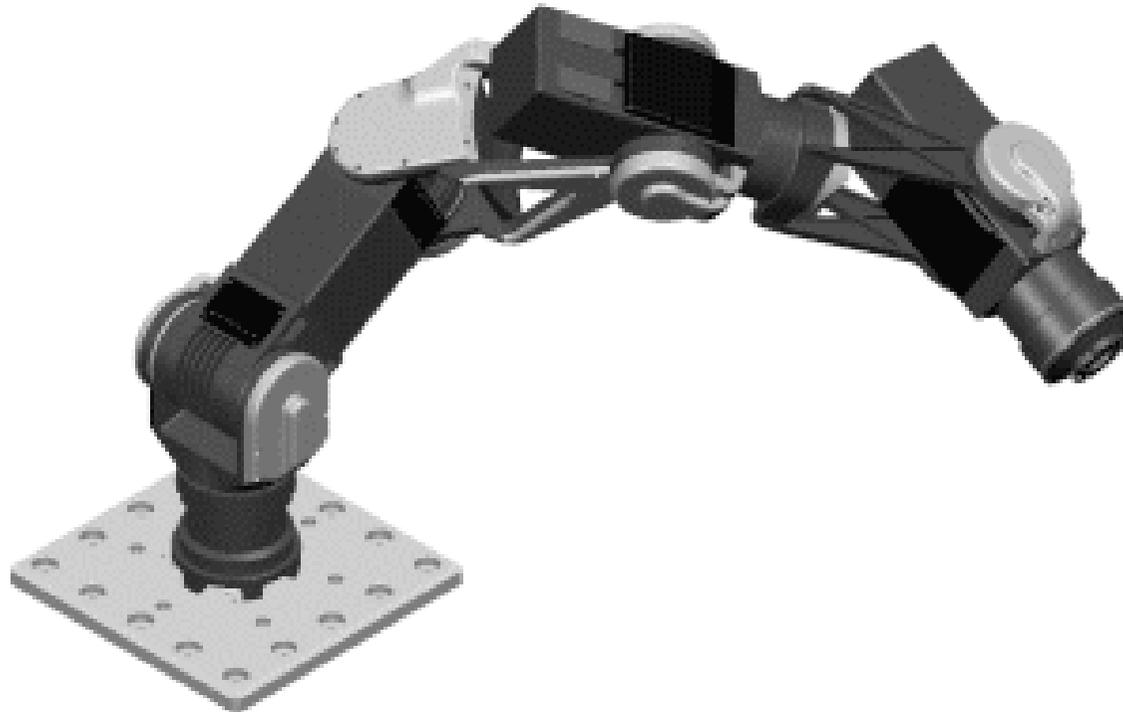


# El modelo WRSPM para requerimientos y especificaciones

## Introducción y designaciones



# Máquinas

- Cuando construimos software, construimos *máquinas* que pueden ser implementadas con una computadora de propósito general.
- No construimos el huésped físico, construimos el *comportamiento* y las *propiedades* que harán que la máquina sea útil para algún propósito específico.
- El propósito de la máquina es ser instalada en el mundo real e interactuar con aquel.

# ¿Qué hacen y cómo funcionan?

- ¿Qué hace un auto?
  - Un auto transporta personas y su equipaje a través de caminos de un lugar a otro”
  - Un auto se mueve, sobre cualquier superficie razonablemente llana y firme, hacia donde sea que su conductor lo dirija”
- Las respuestas hablan del *dominio de aplicación y uso* de la máquina donde los efectos y beneficios serán observados, interpretados, etc.

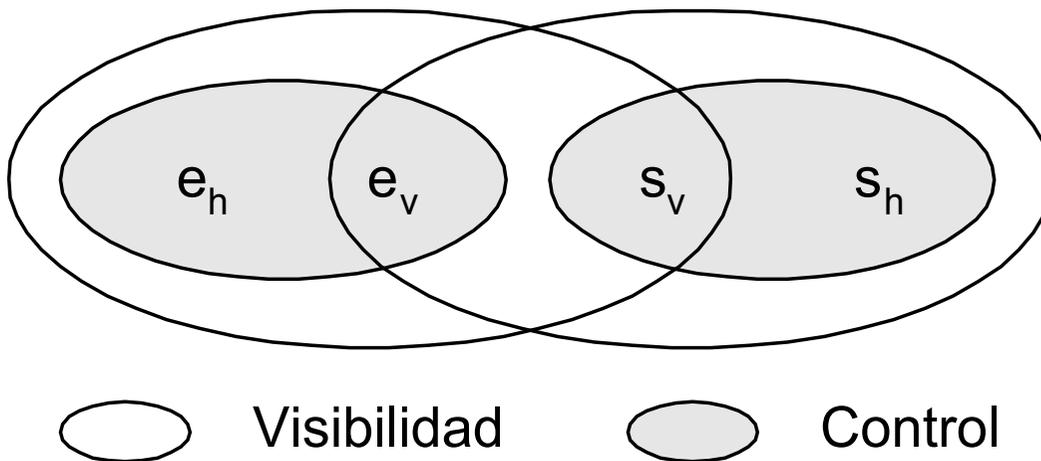
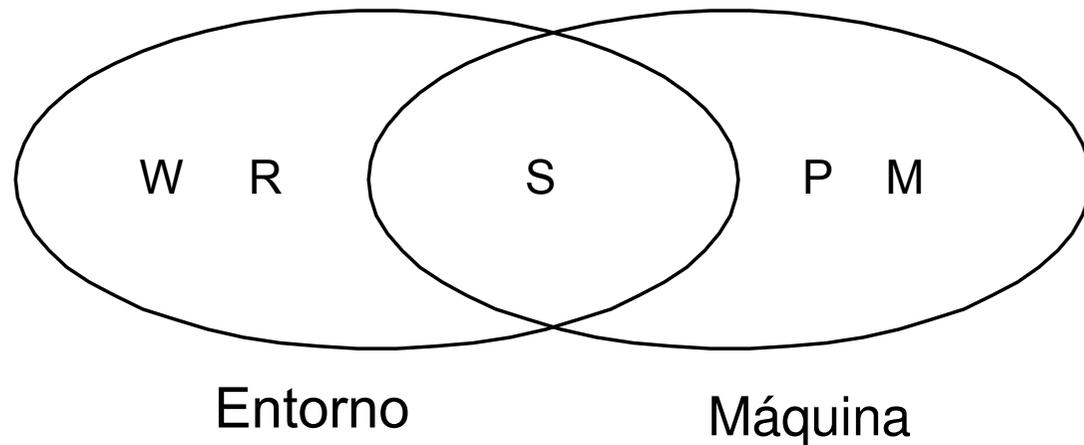
## ¿*Qué* hacen ...? (2)

- Las respuestas NO hablan de la máquina.
  - Personas, equipaje, caminos, movimiento, superficie llana y firme, conductor, etc. no son partes o propiedades del auto, pertenecen al *dominio de aplicación* en el cual se utilizan los autos.
- Cuándo hablamos sobre *qué* hace una máquina hablamos de su propósito, el cual siempre debe ser buscado fuera de ella.

## ¿*Qué* hacen ...? (3)

- El propósito de un sistema de cómputo (máquina) es causar, o cambiar o mantener relaciones en su *dominio de aplicación*. Eso es *qué* hace el sistema.
- Si hablamos de procesos y flujos de datos dentro de la computadora, o si hablamos sobre la estructura y estados de la base de datos, no estamos hablando del *qué* sino del *cómo*.

# El modelo WRSPM



W: dominio de aplicación

R: requerimientos

S: especificación

P: programa que implementa S

M: plataforma de cómputo

$e$  y  $s$ : fenómenos del entorno y del sistema

# DK: conocimiento de dominio

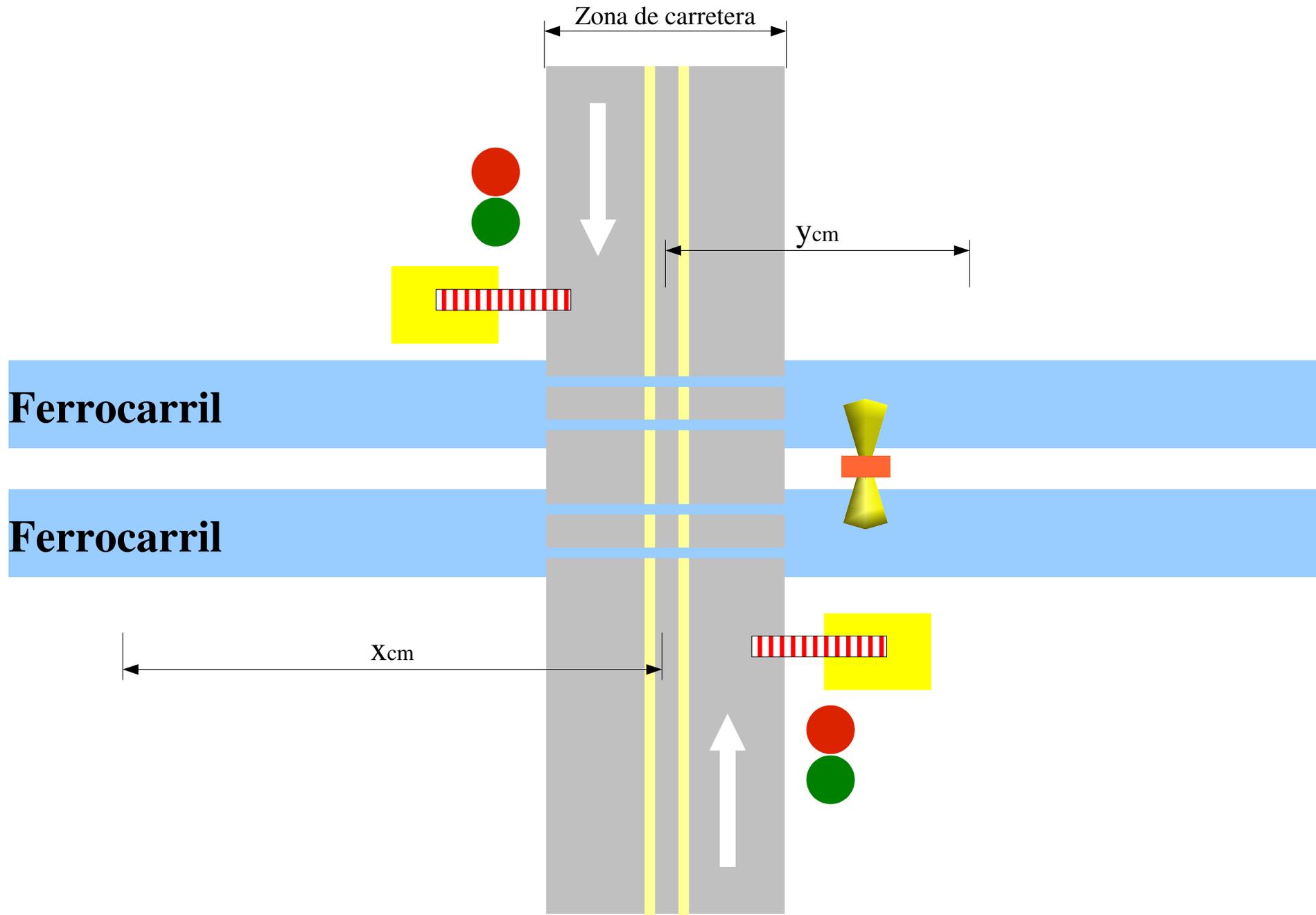
- Es el conocimiento sobre el mundo real relevante al proyecto de desarrollo.
- Provee el contexto donde las acciones de la máquina tienen significado o son útiles.
- Puede ser intangible o físico, estático o dinámico.
- Todo proyecto debe incluir una descripción del conocimiento del dominio independiente de la máquina.

## DK (2)

- El IS es libre de describir toda la información sobre el dominio que considere interesante aun sin saber si la necesitará o no.
- Básicamente, se trata de describir las propiedades del entorno que restringen el funcionamiento de la máquina, propiedades que no pueden ser violadas u olvidadas debido a su carácter *natural* o *físico*.

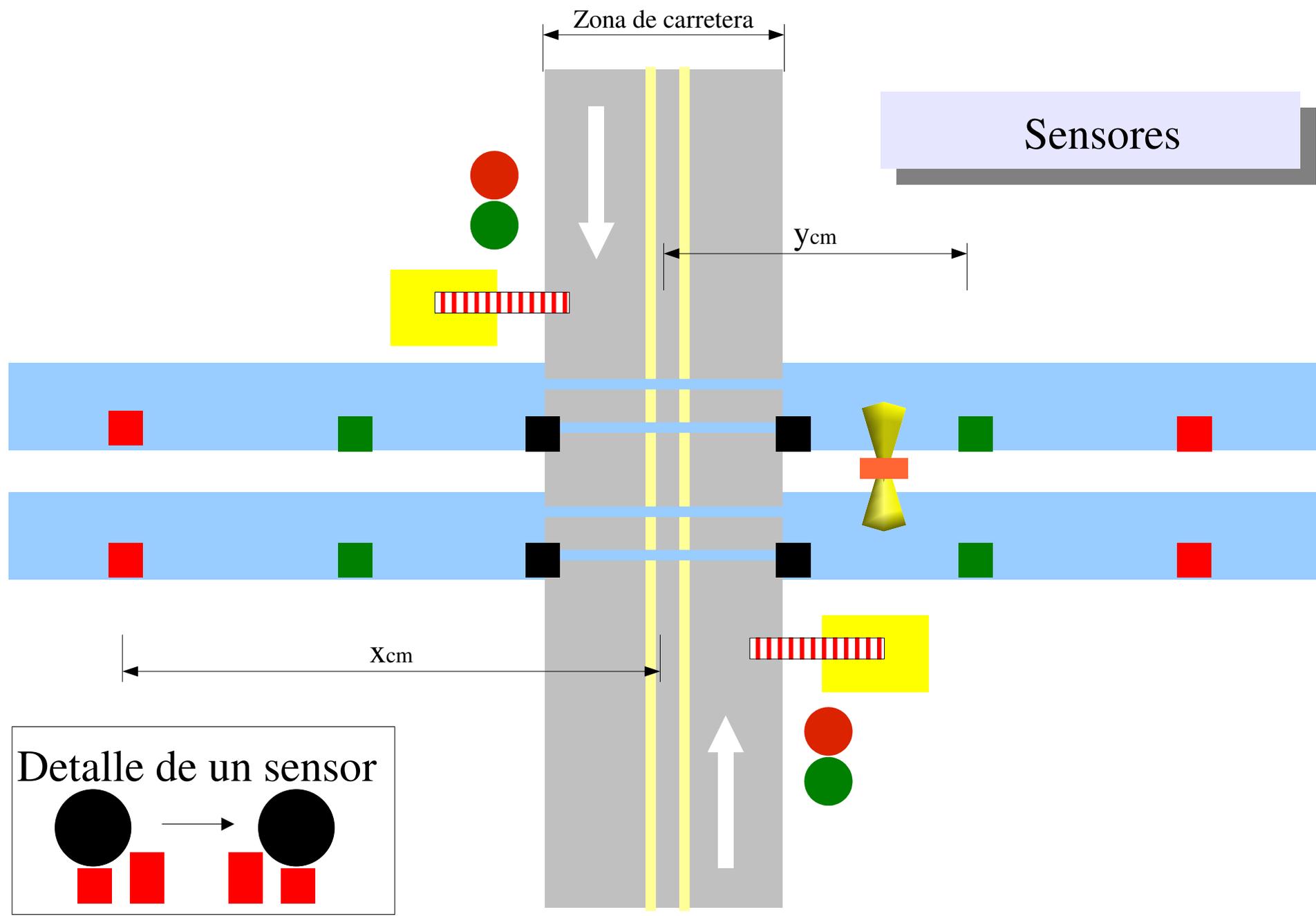
# Ejemplo guía (DK)

- Se trata de controlar la barrera y otros dispositivos de un cruce de ferrocarril en dos sentidos.
- Los dispositivos físicos son: dos barreras, dos semáforos, una alarma y sensores.
- El dominio se completa con los diagramas de las dos transparencias siguientes.



# Algo de DK sobre el cruce

- La alarma se enciende y apaga alternadamente.
- Los semáforos pasan de rojo a verde alternadamente.
- Lo mismo ocurre con la barrera.
- Sólo un tren entra en la zona de interés a la vez.
- Todo tren que entra a la zona de interés eventualmente la abandona.



# R: requerimientos

- Indican lo que el cliente necesita del entorno.
- *Toda* la terminología usada en R debe basarse en la *realidad* del dominio de aplicación.
- Los requerimientos describen las relaciones que se *desea* que la máquina mantenga cuando sea instalada en el dominio.
- La máquina es sólo un medio para satisfacer los requerimientos.

# Ejemplo guía (requerimientos)

- Si un convoy llega a  $x$  metros del centro del cruce, se deben bajar las barreras, poner los semáforos en rojo y debe sonar la alarma.
- Una vez que el último vagón del convoy está a más de  $y$  metros del centro del cruce y alejándose, se debe subir la barrera, poner el semáforo en verde y apagar la alarma.
- Si por alguna razón un convoy llega a  $x$  metros del cruce pero nunca alcanza la zona de la carretera, se debe proceder como si el tren hubiese superado los  $y$  metros.
- Si, luego el convoy reanuda la marcha, se deberán activar las protecciones inmediatamente y el sistema continuará con su funcionamiento normal.

# S: especificación

- Describe el comportamiento *deseado* de la máquina a nivel de su interfaz con el dominio.
- Al igual que los requerimientos este comportamiento se expresa únicamente en términos de los fenómenos del dominio.
- Vista desde la máquina, una especificación es el punto de partida para programar; vista desde el dominio, es una clase restringida de requerimientos.

# Comenzar a describir

- Entonces debemos describir tres cosas: DK, R, S.
- Además queremos describirlas formalmente.
- ¿Por dónde empezamos?
- Por las designaciones.

# Algunas designaciones

- La distancia de seguridad para bajar la barrera  $\approx X$
- La distancia de seguridad para subir la barrera  $\approx Y$
- Un tren cruza la distancia  $X$  acercándose al cruce por la vía  $v \approx \textit{TrenX}(v)$
- Un tren alcanza la zona de carretera por la vía  $v \approx \textit{TrenC}(v)$
- Un tren se detiene en la vía  $v$  en la zona de peligro  $\approx \textit{TrenD}(v)$
- El último vagón de un tren pasa la distancia  $Y$  alejándose del cruce por la vía  $v \approx \textit{TrenY}(v)$

# Algunas designaciones (2)

- El sistema enciende la alarma  $\approx AOn$
- El sistema apaga la alarma  $\approx AOff$
- El sistema baja la barrera  $b \approx BBarrera(b)$
- El sistema sube la barrera  $b \approx SBarrera(b)$
- El sistema pone el semáforo  $s$  en rojo  $\approx Rojo(s)$
- El sistema pone el semáforo  $s$  en verde  $\approx Verde(s)$

# Control de fenómenos (1)

- Es esencial identificar los fenómenos *controlados* por la máquina, los fenómenos *controlados* por el entorno y los fenómenos *compartidos* entre la máquina y el entorno.
- Controlar significa "llevar a cabo", "tomar la responsabilidad de", "iniciar".
- Compartir significa "pertenecer a" o "ser observable por".

# Control (2)

- No puede haber en una descripción, fenómenos controlados por ambos (entorno y máquina).
  - Si esto sucede el fenómeno debe ser dividido en partes más pequeñas de forma tal que cada una de ellas esté controlada por el entorno o la máquina.
- Es muy común que existan fenómenos controlados por el entorno pero que no son compartidos con la máquina.

# Control (3)

- Por lo tanto quedan tres combinaciones:
  - controlado por el entorno/no compartido
  - controlado por el entorno/compartido
  - controlado por la máquina/compartido
- Cualquier lenguaje de especificación debe tener la expresividad suficiente como para determinar a cuál categoría pertenece cada término.
  - No es el caso de la mayoría de los lenguajes

# Control (4)

- Además, el lenguaje debe permitir expresar restricciones sobre fenómenos en cualquiera de las combinaciones.
- La "cuarta" categoría (controlado por la máquina/no compartido) es de interés sólo en la fase de programación.
- Si un fenómeno es de la 2ª categoría y está restringido, la máquina puede asumir esa restricción para cumplir con su cometido.

# Control (ejemplo guía)

- Controlados por el entorno, no visibles a la máquina: *X*, *Y*, *TrenX*, *TrenC*, *TrenD*, *TrenE*, *TrenY*
- Controlados por la máquina, visibles al entorno: *AOn*, *AOff*, *Rojo*, *Verde*, *BBarrera*, *SBarrera*