

Práctica 1. Sistema experto basado en reglas para el control de un Centro de Datos

Práctica 1. Sistema experto basado en reglas para el control de un Centro de Datos

1. Descripción y objetivos
 - Funcionamiento general del Sistema
 - Objetivos
2. Herramientas a utilizar
3. Desarrollo de trabajo
 - Especificaciones y simplificaciones
 - (a) Elementos del Centro de Datos
 - (b) Control de accesos
 - (c) Control de climatización
 - (d) Detección de "desastres"
 - (e) Control de alimentación y sistema eléctrico
 - Funcionamiento deseado
4. Arquitectura general del sistema simulado a implementar
5. Tareas a realizar
6. Normas de entrega

1. Descripción y objetivos

Se pretende desarrollar el **módulo de control** de la infraestructura de un Centro de Datos empleando un sistema basado en reglas programado con CLIPS.

- El módulo a desarrollar será parte de una aplicación de monitorización y control escrita en Python, con tres grandes componentes: *Subsistema de monitorización*, *Subsistema de actuación* y el **Módulo de control** a implementar en este entregable.
- Esta aplicación monitorizará constantemente el estado del Centro de Datos y consultará periódicamente al **módulo de control** implementado en CLIPS, que le proporcionará las acciones a ejecutar sobre los "actuadores" del Centro de Datos.

Funcionamiento general del Sistema

- El **estado del Centro de Datos** (lecturas de sensores, eventos, incidencias, peticiones de acceso, etc) será capturado por componentes Python del *Subsistema de monitorización* (externo al **módulo de control** a desarrollar) y convertido en una colección de **hechos CLIPS** que alimentarán al sistema basado en reglas
- Con esos hechos representando el **estado del Centro de Datos** en un instante determinando, se lanzará el motor de inferencia de CLIPS donde se habrán cargado las reglas y demás elementos (plantillas, hechos iniciales, etc) que conforman la lógica del **módulo de control**
- El resultado de la ejecución de las reglas activadas por el motor de inferencias de CLIPS se materializará en una serie de hechos CLIPS que representan las **acciones de control**. Estos hechos se extraerán de la Memoria Activa final y se pasarán al *Subsistema de actuación* (externo al **módulo de control**) para ejecutarlas sobre los actuadores del Centro de Datos (sistemas de calefacción/aire acondicionado, puertas, etc)

Objetivos

1. Realizar las fases de "Adquisición del conocimiento" y de "Representación de conocimiento" que típicamente forman parte de los proyectos de desarrollo de Sistemas Basados en el Conocimiento.
 2. Utilizar las Reglas de Producción y Marcos/Templates de CLIPS como formalismos de representación del conocimiento
 3. Experimentar con el desarrollo de Sistemas Basados en Reglas en un dominio de complejidad media-baja y resolver problemas como inconsistencia de reglas, necesidad (o no) de prioridades, etc
 4. Integar un módulo de control basado en reglas CLIPS en una aplicación Python
-

2. Herramientas a utilizar

- Lenguaje y motor de inferencias basado en reglas [CLIPS](#)
 - Librería [clipsy](#) para la integración de CLIPS con programas Python
-

3. Desarrollo de trabajo

Se pretende implementar un sistema sea capaz de conocer el estado del Centro de Datos y de actuar en consecuencia.

Como guía, en esta sección se describe el tipo de elementos presentes en los Centros de Datos que se pretendería monitorizar y un esbozo del funcionamiento general del sistema completo.

Nota: Las "especificaciones" de este documento son orientativas y se proponen como guía para el trabajo

- No es imprescindible manejar todos los elementos del Centro de Datos indicados en la siguiente subsección
- Tampoco es necesario implementar el control exactamente del modo que se describe.
- Es posible añadir otros elementos y/o hacer un control más detallado del descrito.

Especificaciones y simplificaciones

(a) Elementos del Centro de Datos

El Centro de Datos está organizado en una serie de **zonas** que cuentan con un nombre/etiqueta identificativo.

- Las **zonas** están delimitadas por "jaulas" y cuentan con mecanismos de control de acceso (puertas automáticas) que exigen la identificación (tarjeta, biometría, etc) del usuario cuando accede y cuando sale
- En relación a las restricciones de acceso, las **zonas** se categorizan en 3 niveles (de menor a mayor grado de restricciones):
 - *zona fría*: alberga elementos de acceso restringido a los que se precisa acceso frecuente (switches, router, paneles de cableado)
 - *zona templada*: elementos con restricciones y frecuencia de acceso intermedia (racks y servidores de uso habitual)
 - *zona caliente*: elementos especialmente sensibles de acceso más restringido y menos frecuente (racks, servidores, dispos. almacenamiento)

Las **zonas** albergan colecciones de **racks** donde se ubican los equipos del Centro de Datos (servidores, cabinas de discos, switches, ...)

(b) Control de accesos

- Cada **usuario** tiene un nivel de acceso acorde a los niveles de las **zonas** (*fría, templada, caliente*) que se codifica, junto con su identificación, en el mecanismo de identificación (tarjeta, biometría, etc) que utiliza
- Los intentos de acceso o salida de una **zona** tienen en cuenta esos niveles, permitiendo a los usuarios de niveles más altos acceder a las **zonas** de niveles iguales o menores que el suyo propio
- Cada **zona** tiene un detector de presencia que indica si hay personas dentro o no

(c) Control de climatización

- Cada **zona** del Centro de Datos tiene una serie de sensores de climatización:
 - temperatura (media de la zona): con valores de -20° a +80° y resolución/"saltos" de 1° C
 - humedad: con valores de 0% a 100% y resolución/"saltos" de 10%
- Todas las **zonas** del Centro de Datos tienen el mismo *rango ideal* de temperatura mínima y máxima y el mismo *rango ideal* de humedad mínima y máxima
- Cada **rack** del Centro de Datos tiene un sensor de temperatura (de -20° a +80°, con resolución/"saltos" de 1° C) y un rango de temperaturas de trabajo ideales (temp. mínima y temp. máxima)
- El Centro de Datos cuenta con un sistema de **calefacción** global que se puede encender y apagar a demanda
- Cada **zona** del Centro de Datos cuenta con un módulo de **aire acondicionado** propio. Este **aire acondicionado** puede encenderse, apagarse y aumentar o reducir intensidad en tramos de +/- 1 °C
- Cada **zona** del Centro de Datos cuenta con un **ventilador** propio que fuerza la circulación de aire afectando a la temperatura y humedad de la **zona** de forma independiente. Este **ventilador** puede encenderse, apagarse y aumentar o reducir la velocidad en tramos de +/- 20 rpms (con un máximo de 100 rpms)

(d) Detección de "desastres"

- Cada **zona** del Centro de Datos cuenta con un detector de humo y un detector de filtraciones de agua.
- Cada **zona** tiene vinculado un tipo de contenido, que puede ser *normal* o *sensible* (material electrónico, productos químicos, etc)
- Cada **zona** cuenta con un doble sistema de extinción de incendios (con agua o con gas).
 - Extinción por agua: aplicable *sólo* si el contenido de la **zona** *no* es *sensible*
 - Extinción por gas: aplicable *sólo* si no hay personal dentro de la **zona**

Existe un sistema de megafonía para todo el Centro de Datos que permite emitir mensajes de aviso (por ejemplo: "evacuación del edificio", "alerta por inundación en ...", etc)

(e) Control de alimentación y sistema eléctrico

- Cada **zona** tiene un sistema de iluminación automático que se encienderá cuando haya gente dentro de la misma (se ha producido una entrada) y se apaga al abandonar la zona la última persona que accedió.
- Cada **rack** dispone de un monitor de la corriente de alimentación, que indica el voltaje de entrada en cada instante, con valores entre 100 V y 300 V.
- Cada **rack** tiene un rango de alimentación ideal (210-230 V), los valores fuera de ese rango generarán alertas.

Funcionamiento deseado

El **módulo de control** CLIPS contendrá las definiciones de hechos que representan los conceptos descritos anteriormente (`deftemplate` de **zonas**, **sensores**, **racks**, **acciones de control**, etc) y las reglas de control implementadas.

- Habrá un conjunto de *hechos iniciales* que describirán la estructura y los elementos del Centro de Datos de trabajo: **zonas** concretas, **racks**, etc

En cada iteración se le presentará al **módulo de control** CLIP un conjunto de hechos representando el estado de los **sensores** y demás elementos del Centro de Datos en un instante dado.

Sobre esa colección de hechos, se aplicarán las reglas CLIPS y el resultado final será el conjunto de acciones a ejecutar sobre el Centro de Datos

No se implementarán ni simularán la ejecución de esas acciones

- se asumen que serán puestas en acción por el *Subsistema de actuación*, que no se implementará
- el **módulo de control** actúa de modo "reactivo", ante una descripción de un estado concreto del Centro de Datos, propone una lista de acciones a realizar

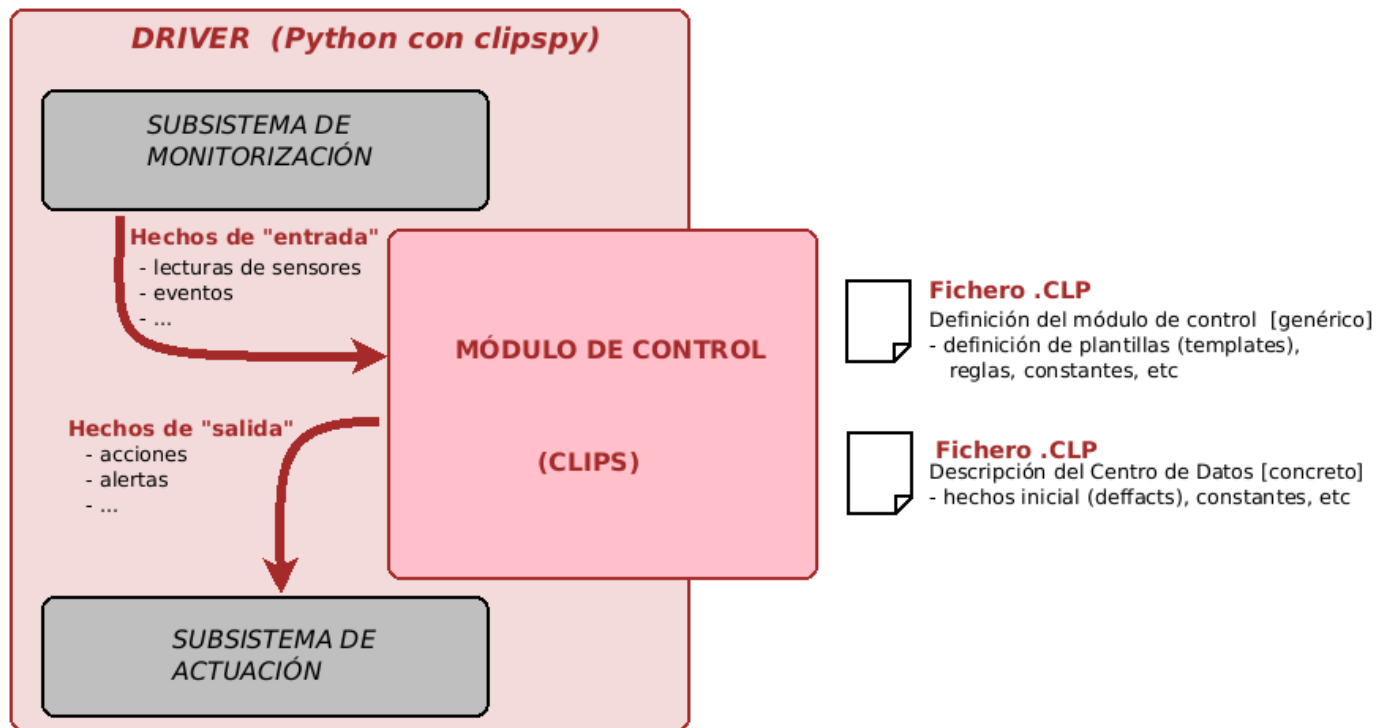
Se pretende que el sistema proponga acciones para:

- mantener las restricciones de control de accesos de las **zonas** y **usuarios**
- mantener los niveles de temperatura y humedad de las **zonas** en los valores deseados, actuando sobre la calefacción, el mecanismo de aire acondicionado y los ventiladores
- monitorizar los niveles de temperatura de cada **rack** creando alertas en el caso de que estos se salgan de los valores óptimos
- monitorizar la alimentación eléctrica de cada **rack** creando alertas en el caso de que el voltaje de alimentación se salgan de los valores óptimos
- implementar la detección y actuación ante desastres (fuego e inundaciones) en cada **zona** actuando sobre los mecanismos de extinción y creando las correspondientes alertas y avisos por megafonía

4. Arquitectura general del sistema simulado a implementar

El componente a implementar (el **módulo de control** CLIPS) se materializará en al menos dos ficheros `.clp`:

- Un fichero `.CLP` con (1) la definición de los tipos de hechos (`deftemplate`) que se usarán para almacenar la información del Centro de Datos (**zonas**, **racks**, **usuarios**, **sensores**, **acciones**, etc) y (2) la definición de las reglas (`defrule`) que implementarán el **módulo de control** y, como resultado del proceso de inferencia, generarán los hechos que representen las **acciones de control**
- Un fichero `.CLP` con las definiciones de hechos iniciales (`deffacts`) que describan el Centro de Datos concreto donde se va a llevar a cabo la monitorización (hechos con las **zonas** y **racks** concretos que forman el Centro de Datos, **usuarios** del Centro de Datos, **sensores** presentes en el Centro de Datos, etc)



El sistema completo de monitorización de Centros de Datos se simulará con un programa Python que actuará como "driver" del **módulo de control** CLIPS que conforman los ficheros .CLP anteriores

- Ese "driver" Python usará la librería `clipspy` para tres funciones:
 1. Cargar los ficheros .CLP que implementan el **módulo de control**
 2. Insertar en la *memoria activa* del Sistema Experto cargado los hechos que representan el **estado actual** del Centro de Datos monitorizado (valores/lecturas concretas de **sensores**, acciones de accesos de **usuarios** concretos, **eventos**, etc) en un momento dado y sobre los que las reglas del **módulo de control** decidirán las acciones a tomar
 3. Ejecutar el motor de reglas con los hechos anteriores cargados y, una vez finalizado el razonamiento, recuperar los hechos que representan las **acciones de control** decididas por las reglas y mostrar por pantallas dichas acciones

5. Tareas a realizar

Las tareas concretas a realiza en la práctica serán las siguientes:

1. Definir el alcance del problema a resolver, tomando como base las descripciones de la Sección 3
 - Especificar el sistema "real" a controlar (pueden simplificarse los elementos descritos en el enunciado, añadir otros nuevos o incorporar un control más detallado)
 - Características de los Centros de Datos considerados en **módulo de control** implementado
 - Descripción de los **sensores** y de las **acciones de control** manajados
2. Conceptualización del dominio
 - Identificar los conceptos relevantes y diseñar las plantillas de los hechos a considerar en el **módulo de control**
 - Diseñar las reglas a incluir en el **módulo de control**
3. Implementación en CLIPS del problema
 - Representar en el formato de CLIPS las definiciones de hechos (`deftemplate`) y las reglas a implementar (`defrule`)

- Implementar el "driver" Python con la librería `clipsy` responsable de cargar la implementación CLIPS del **módulo de control**, cargar los hechos con el **estado actual** y recuperar y mostrar las **acciones de control**

4. Ejecución del sistema y pruebas

- Definir un fichero con la definición de los hechos (`deffacts`) que caracterizan un Centro de Datos de pruebas
- Incluir en el "driver" Python código para insertar hechos de prueba y verificar el funcionamiento del **módulo de control** desarrollado

6. Normas de entrega

Trabajo **individual** o en **parejas**

- **Importante:** ambos miembros del grupo deberán subir su entrega a la tarea de Moovi
- Una vez entregado en Moovi se habilitará un cuestionario de Moovi de 6-7 preguntas de respuesta múltiple (con 4 opciones posibles y penalización por fallos [3 fallos = 1 correcto])

Entregable:

- Ficheros que conforman el sistema desarrollado:
 - fichero .CLP con `deftemplate` y `defrule`
 - fichero .CLP con `deffacts` de prueba
 - fichero .py con el código del "driver" Python
- Memoria explicativa con la siguiente estructura:
 - Descripción del problema
 - Detallar el alcance del sistema propuesto
 - Elementos considerados
 - Simplificaciones impuestas
 - Ampliaciones realizadas
 - Descripción de la implementación en CLIPS
 - Tipos de hechos y/o clases usados
 - Breve descripción de las reglas usadas y de su estructura
 - Ejemplo de funcionamiento y descripción de las pruebas realizadas
 - Conclusiones y problemas encontrados (posibles mejoras, idoneidad de la herramienta, etc)
 - Bibliografía consultada

Nota: Incluir nombre, DNI y e-mail de todos los alumnos del grupo en la portada

Fecha de entrega: `<por determinar>` (principios de noviembre)