Desarrollo de Sistemas Inteligentes aplicados a redes eléctricas industriales

Andrés Krapf y Ana Casali
{akrapf,acasali}@fceia.unr.edu.ar

Depto. de Sistemas e Informática
Facultad de Cs.Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Av. Pellegrini 250. 2000 Rosario, Argentina.

Abstract

Este trabajo sintetiza el desarrollo de un Sistema Inteligente de Control (SIC) para la Automatización de Sistemas Eléctricos de Potencia (SEPs) del tipo industrial. Las tareas automatizadas son detección y aislación de fallas, y automatización de métodos de back-up que soporten posibles pérdidas de integridad en los componentes involucrados. Para este sistema se propone una arquitectura multiagente altamente escalable y flexible, y se decriben los agentes necesarios para realizar las distintas tareas. Se implementan prototipos de dos de los agentes que integran el SIC utilizando el modelo BDI (Belief-Desire-Intention) y se evalúa su comportamiento.

Keywords: Diseño Orientado a Agentes, Agentes BDI, Sistemas Eléctricos de Protencia.

1. INTRODUCCIÓN

El objetivo de un Sistema Eléctrico de Potencia (SEP) es proveer energía eléctrica con criterios de calidad, seguridad y fiabilidad acorde a las normas vigentes.

En este trabajo se hace especial referencia a las redes eléctricas industriales[1]. En la actualidad estos sistemas presentan la posibilidad de mejorar la calidad de prestación. Para ello es necesario disponer de medios de protección y control que se adapten a distintas condiciones de hardware y de operación del SEP. La condición básica de todo SEP es el equilibrio entre generación y demanda. El mundo y en especial la Argentina se enfrenta al problema que los recursos energéticos son limitados y el margen de reserva de generación es cada vez menor, por lo que es necesario agudizar los mecanismos de control para evitar colapsos generales o parciales.

Para los ingenieros de SEPs el principal factor a tener en cuenta, es la calidad del servicio, en el cual uno de los elementos más importantes es la continuidad del mismo, siendo una preocupación la pérdida total del suministro y en un segundo lugar se encuentran las pérdidas parciales del servicio.

Para dar soporte a este problema, los sistemas inteligentes son una importante alternativa a explorar. Los sistemas basados en técnicas de la Inteligencia Artificial y en particular la tecnología de los sistemas multi-agentes, son una herramienta valiosa a la hora de desarrollar sistemas para problemas reales complejos, de carácter distribuido y donde se espera que sus componentes tengan cierta autonomía. Existen diversos trabajos enfocados al desarrollo de sistemas inteligentes en este dominio de aplicación, por ejemplo [14] y [15].

[1]Comprenden a los grandes consumidores de energía eléctrica, tales como las industrias del acero, químicas, papel, etc.; que generalmente reciben el suministro eléctrico en alta o media tensión.

1.1. Dominio de aplicación

Un SEP es un conjunto de equipos que permiten energizar cargas en forma segura y confiable, en distintos niveles de tensión, ubicados generalmente en diferentes lugares físicos.

Entre los componentes que se encuentran en un SEP, podemos mencionar principalmente a Protecciones Eléctricas (Relays), Interruptores, Transformadores, Barras, Generadores, Líneas y Cargas.

Las protecciones eléctricas tienen un rol fundamental en el proceso de adquisición y análisis de datos. Una protección eléctrica (también llamada *relé*) es un dispositivo que sensa variables de redes, para procesarlas mediante algoritmos adecuados. A su vez ordena acciones frente a situaciones anormales. Existen distintos tipos de protecciones que se clasifican en base a su función. Para el trabajo realizado se considera principalmente el relé denominado *de máxima corriente y direccional*. Cada relé tiene asignada una "corriente de seteo" (I_c). I_c es superior a la máxima corriente que puede circular por el relé en condiciones normales en al menos un 30 %, para tener en cuenta errores de calibración. Se pueden definir las principales señales provenientes de un relé de la siguiente manera:

$$S_1 = \begin{cases} 1 & \text{si } I \geq I_c \\ 0 & \text{si } I < I_c \end{cases}$$

$$S_2 = \begin{cases} 1 & \text{si } I \text{ va de Barra a Línea} \\ 0 & \text{en } I \text{ va de Línea a Barra} \end{cases}$$

$$S_3 = \begin{cases} 1 & \text{si el Relé presenta alguna falla interna} \\ 0 & \text{si no} \end{cases}$$

Si la corriente (I) que circula es mayor (o igual) que I_c, entonces el relé se *excita* (también suele decirse que opera). Esta información se obtiene mediante S_1. Mediante S_2 obtenemos la dirección en que circula la corriente y mediante S_3 podemos saber si el relé presenta alguna falla interna (por motivo desconocido). Otros tipos de protecciones que consideramos son las denominadas *protecciones propias de transformador (PPT)*. Estas presentan señales de excitación y falla-relé (falla interna). También ordenan disparos.

Las protecciones de máxima corriente, tienen un tiempo de seteo t_a. Si la protección se encuentra en estado de excitación durante el tiempo t_a, inmediatamente ordena la apertura del interruptor asociado (se dice que ordena disparo o "trip").

Un interruptor (de potencia) es un dispositivo utilizado para desconectar una carga o una parte del sistema eléctrico, tanto en condiciones de operación normal como en condición de cortocircuito. La operación de un interruptor puede ser automática o manual, accionada por la señal de un relé encargado de vigilar la correcta operación del sistema eléctrico donde está conectado o de un operador a distancia. Cada interruptor suele estar asociado a un relé.

Las barras pueden ser consideradas como nodos del SEP. Son puntos de maniobra pues a ella concurren las líneas, se conectan los transformadores y las cargas. Las cargas son elementos que transforman la energía eléctrica en otra forma de energía: mecánica, calórica, etc; pueden ser por ejemplo: motores, hornos, equipos de iluminación, etc. Las líneas cumplen el rol de conductores y transmisores de la energía. Un transformador es un equipo que permite aumentar o disminuir la tensión en un circuito eléctrico. Un generador es una fuente productora de energía eléctrica.

Es muy importante mantener el suministro de energía en forma adecuada, pues los daños ocasionados pueden ser de costos muy elevados e incluso se puede ver afectada la vida de seres humanos, si consideramos por ejemplo un sistema de ventilación. Resulta necesaria entonces la existencia de sistemas de control de SEPs.

1.2. Problemática actual

En la actualidad se cuenta con sistemas SCADA para la adquisición de datos, aplicaciones aisladas que realizan análisis de aspectos específicos y también software para soporte de decisión.

La complejidad actual de los SEPs radica en que no es posible disponer de una regulación (seteo) única para todas las condiciones operativas factibles. En las redes actuales se pueden presentar operaciones no consideradas previamente debido a servicios prestados bajo severidades extremas motivadas por salidas de servicio de una parte de las instalaciones y ante la necesidad de continuar con el suministro, se alteran las condiciones preestablecidas. Entre ellas se pueden citar sobrecargas en líneas que llevan a desconexiones en cascada.

Es necesario que para lograr un funcionamiento adecuado de las protecciones eléctricas, se produzca una variación en su seteo en forma automática (relé adaptivo), teniendo en cuenta estados de operación de elementos adyacentes y lejanos [2].

Con el fin de desarrollar un sistema de control acorde a las necesidades vigentes se utilizaron en este proyecto herramientas de la Inteligencia Artificial, principalmente se hizo uso de tecnología multi-agente y de Sistemas de Razonamiento Procedural (PRS) [12]. El resultado de este trabajo es el diseño de un Sistema Inteligente de Control (SIC) aplicado a redes eléctricas y la implementación de dos de sus agentes (el agente Detección y Aislación de Fallas (DAF) y el agente Back-Up) utilizando el modelo BDI. Esta presentación tiene la siguiente estructura: en la sección 2 se describen los requerimientos que debe verificar el SIC. En las secciones 3 y 4 se desarrollan respectivamente la arquitectura del SIC y el diseño de sus agentes. Finalmente, las secciones 5 y 6 corresponden a observaciones sobre la experimentación y conclusiones.

2. REQUERIMIENTOS

Por medio de un trabajo de Ingeniería de Conocimiento se realizó la extracción de requerimientos para el sistema planteado. Mediante sucesivas reuniones con el experto en el dominio de aplicación [2] se logró refinar y comprender los objetivos del mismo.

En primera instancia se identifica el siguiente objetivo general del sistema de control: *actuar ante una falla, tratando de mantener la integridad del mayor área posible bajo condiciones de calidad adecuadas*. Luego, se puede descomponer el objetivo global en las siguientes tareas o subojetivos:

- detectar fallas;
- aislar fallas;
- respaldar componentes que presenten falla interna (protecciones e interruptores);
- restaurar el servicio a zonas desenergizadas y
- mantener un modelo del SEP.

En este trabajo se centra la atención en las tres primeras tareas. También debe considerarse un importante requerimiento temporal. Los tiempos de actuación en caso de existir una falla en el SEP deben ser del orden de 100 milisegundos. Si bien el presente trabajo no pretende necesariamente que la implementación verifique este requerimiento dado que se trata de un prototipo, debe ser considerado, ya que una implementación final sí debería verificarlo.

[2]Ing. Luis A. Krapf , Escuela de Ingeniería Eléctrica, Facultad de Cs. Exactas, Ingeniería y Agrimensura, UNR

Un aspecto que también se debe tener en cuenta es la capacidad de adquirir datos provenientes del SEP, realizar algún tipo de preprocesamiento (e.g., filtrado[3]) y luego transmitir eventos significativos a las entidades interesadas.

Otro requerimiento, no funcional, involucra la posibilidad o facilidad para integrar sistemas de análisis preexistentes o "legacy". El sistema a desarrollar deber permitir una fácil integración.

Para continuar con el proceso de adquisición del conocimiento, se obtuvieron una serie de escenarios, también llamados *casos de uso*. Cada caso de uso representa una situación de falla y determina las acciones necesarias que deben efectuarse correspondientemente. Esto permite realizar un *refinamiento en los requerimientos*. Por cada falla analizada se genera un caso de uso, el cual a su vez es representante de un conjunto de situaciones que deben ser tratados de forma equivalente. Esta agrupación se debe sencillamente que dependiendo de la configuración de cada protección se puede excitar un mayor o menor número de protecciones. Las protecciones que deben tener un comportamiento común para toda instancia de una clase de casos de uso son aquellas más próximas a la falla, salvo en casos de incertidumbre en los que estas protecciones pueden no dar indicaciones (considerando de esta manera fallas internas, funcionamientos anómalos o errores de calibración).

A continuación se describe un escenario a modo de ejemplo. La situación que se desea detectar se describe en forma gráfica (ver Fig. 1). Para simplificar la notación, no se introducen los nombres de los interruptores. Se asume en términos generales que el relé R_j está asociado al interruptor I_j. Además pueden aparecer flechas debajo de algunos componentes. Estas corresponden a indicaciones (o señales) del sentido de la corriente provistas por los relés. Si bien un relé puede dar este tipo de indicación sin estar excitado, debe asumirse que lo está (también con el fin de simplificar la notación).

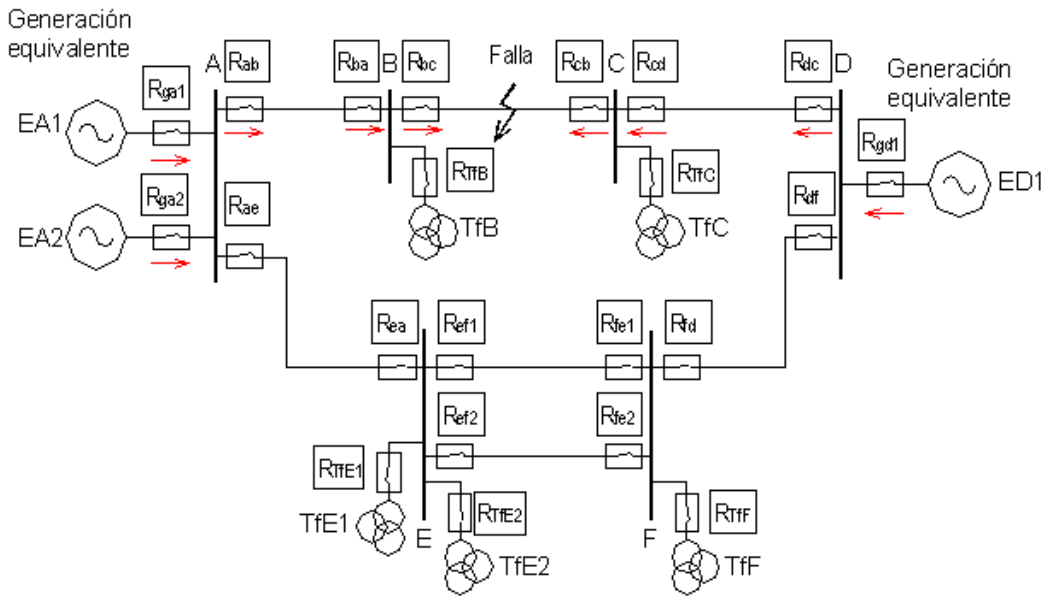

Figura 1: Caso 1. Falla en LINEA

En el caso ilustrado en la figura 1 se presenta la falla en una línea. Las **acciones** necesarias para resolverlo son:

- disparar los interruptores I_{bc} y I_{cb}.

[3]evitando por ejemplo, el envío de señales ante una intermitencia en las mismas

- si no abre I_{bc} se debe disparar I_{ba} y I_{ab}. También deben desconectarse las cargas involucradas, o sea, abrir I_{Tfb} y las cargas colgadas del transformador (TfB) deben desconectarse recursivamente (empezando por la más cercana).
- análogamente, si no abre I_{cb} se debe disparar I_{cd} y I_{dc}. También deben desconectarse las cargas involucradas, o sea, abrir I_{TfC}. Las cargas colgadas del transformador (TfC) deben desconectarse recursivamente (empezando por la más cercana).
- se deben impedir (bloquear) los disparos de los demás relés que se excitaron.

En este trabajo concretamente se identificaron nueve casos de uso, abarcando entre ellos existencia de incertidumbre. Se presenta incertidumbre cuando la información recolectada es incompleta para alcanzar una solución (por ejemplo el caso en que una protección no envíe señal cuando en realidad debiera hacerlo) o cuando la información obtenida es errónea y genera inconsistencias (por ejemplo el envío de señales incorrectas, como ser la dirección de corriente opuesta a la real) debiéndose esto a fallas internas o funcionamientos anormales.

2.1. Características del problema

El problema en el cual nos concentramos tiene algunas características importantes a considerar. Los datos utilizados para el análisis del SEP, se encuentran distribuidos geográficamente, puesto que la información puede ser recolectada de diferentes estaciones remotas. Estos datos tienen a su vez una naturaleza altamente dinámica. Por otro lado, la complejidad del problema puede ser atacada mediante tareas independientes. Estas características, claramente nos conducen a que la aplicación de la tecnología de sistemas Multi-Agentes [10] es altamente adecuada para resolver un problema como el propuesto.

3. ARQUITECTURA MULTI-AGENTE

Durante los últimos años se han desarrollado varias metodologías para el análisis y diseño orientado a agentes (AOSE). Ejemplos de esto son DESIRE [7], Gaia [11], Prometheus [8] y para el desarollo de agentes BDI [1]. Si bien hay similitudes, cada metodología tiene aspectos muy interesantes, por lo que se extrajeron conceptos de las mismas para avanzar hacia la etapa diseño.

3.1. De los Requerimientos hacia la Arquitectura

El primer paso seguido es la *descomposición de tareas* (según la metodología DESIRE [7]). Este proceso tiene como fin la obtención de tareas o funcionalidades requeridas.

Las funcionalidades deben ser lo más simples posible de manera de tratar con un único aspecto o comportamiento esperado del sistema. Por lo tanto podemos identificar las siguientes funcionalidades del sistema: *identificar fallas*; *bloquear disparos innecesarios*; *aislar fallas identificadas*; *administrar eventos*; *respaldar componentes que presenten falla (Back-Up)* y *restaurar el servicio*.

Al definir una funcionalidad también es importante definir la información que requiere y la información que produce. Para describir o especificar una funcionalidad se utilizan descriptores que contienen el *nombre* de la misma, una *descripción* informal breve, una lista de *acciones*, una lista de *percepciones*, *datos usados* y *producidos*, y una breve descripción de las interacciones con otras funcionalidades.

Esta forma de identificar funcionalidades, da la posibilidad de estudiar la interacción entre ellas, lo cual es necesario para el siguiente paso. Este consiste en identificar cuáles serán los agentes que van a existir en el sistema.

Veamos la descripción de una funcionalidad:

Nombre: Detector de fallas.
Descripción: Determina la presencia de falla, su localización y su tipo.
Acciones: -.
Percepción. Estado de protecciones.
Acceso a Datos: Lectura del estado de protecciones, topología de red. Escritura de datos con la región donde se encuentra la falla y su clasificación.
Interacciones: Provee la región generada a las funcionalidades Bloqueo y Aislación. A esta última también le provee la clase y locación de falla.

En el diseño de un sistema Multi-Agente, resulta fundamental definir los agentes que lo componen, así como también las interacciones entre ellos. Se puede considerar a un agente como una combinación de funcionalidades. Luego, se puede determinar qué agentes existirán evaluando diferentes combinaciones según criterios provenientes del diseño orientado a objetos y reutilizados en AOSE. Se pueden asignar funcionalidades a agentes de modo de encontrar una configuración en la que se consiga un bajo nivel de acoplamiento y un gran nivel de cohesión. Por ejemplo, si distintas funcionalidades usan los mismos datos entonces es una indicación de que se deben agrupar dado que habrá una gran interacción entre ellas. Si distintas funcionalidades están claramente no relacionadas es una razón para no agruparlas, así como también si existen en distintas plataformas de hardware.

Utilizamos un diagrama de interacción entre funcionalidades (ver figura 2) para mostrar las funcionalidades (rectángulos), los datos (elipses) y conexiones entre ellos (flechas). Una flecha desde una funcionalidad hacia datos indica que la funcionalidad produce los datos.

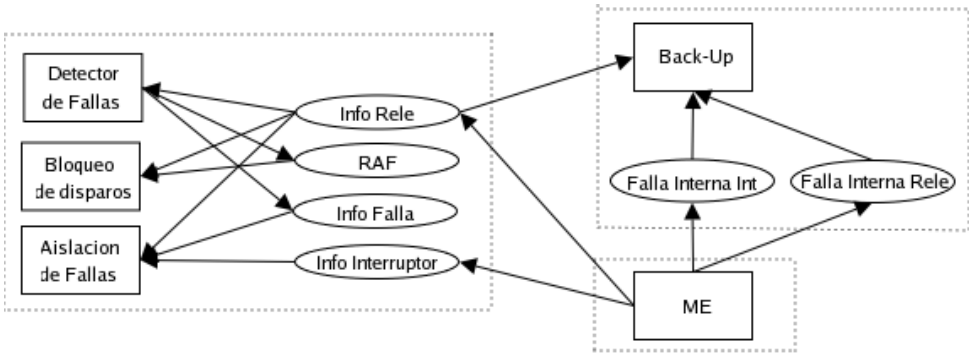

Figura 2: Interacción entre funcionalidades

Del análisis de las funcionalidades antes identificadas y del diagrama de interacción resulta claro y natural agrupar las funcionalidades de forma tal que se obtienen los siguientes agentes:

- Detección y Aislación de fallas (DAF)
- Back-Up (BK)
- Manejador de Eventos (ME)
- Reposición de Servicio (RS)

Además de los mencionados consideramos dos tipos de agente, cuya representación está motivada por correspondencia directa con elementos muy importantes para el dominio de aplicación:

- Agente de Protección (AP)
- Agente de Interrupción (AI)

En el presente trabajo no se describe el diseño de los agentes RS y ME. Sí se consideran aspectos relacionados a la interaccón con el agente ME, dado que éste constituye el nexo entre los agentes DAF y Back-Up y el sistema eléctrico subyacente (SEP). El Agente de Protección es un tipo de agente que representa una protección eléctrica. Puede actuar autónamente para proteger dispositivos del SEP. Consiste principalmente de reglas, lógica de relés, y características de operación. El Agente de Interrupción representa el comportamiento de un interruptor.

A continuación se describe un agente tal como aparece en *la guía de agentes*, siguiendo el estilo de notación empleado en Prometheus [8].

Nombre: Agente de Detección y Aislación de Fallas (DAF)
Descripción: Analiza los eventos provenientes de una capa inferior con el objetivo de detectar y encontrar con la mayor precisión posible la locación de una falla eventual para luego poder aislarla y evitar una desenergización innecesaria.
Cardinalidad: 1
Funcionalidades incluidas: Identificar región de falla, bloquear disparos innecesarios, aislar fallas
Lee datos: Estado de los componentes (modelo propio)
Escribe datos: Estado de los componentes (modelo propio)
Interactúa con: ME (para obtener eventos de interés y ordenar acciones), Back-Up (obtener correcciones en la información sobre componentes con falla interna)

3.2. Interacción entre Agentes

Tan importante como definir los agentes que forman el sistema, es describir la interacción que habrá entre ellos. La figura 3 muestra un diagrama de interacción entre los agentes propuestos. Este diagrama simplemente vincula agentes que tienen algún tipo de interacción, ya sea mediante datos compartidos o mensajes. La existencia de múltiples agentes de un mismo tipo se representa en el diagrama mediante rectángulos superpuestos. Resulta útil este diagrama para considerar el nivel de acoplamiento entre agentes.

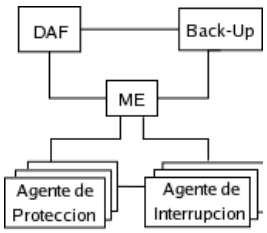

Figura 3: Interacción básica entre Agentes

Mediante un mecanismo de suscripción, el Agente Manejador de Eventos (ME), mantiene registros de cuáles son los eventos de interés para cada agente existente en el sistema y es su función enviarlos. Por lo tanto es importante, definir cuáles serán los eventos generados como resultado de la información obtenida del entorno (SEP). Los eventos generados serán los percibidos por los agentes

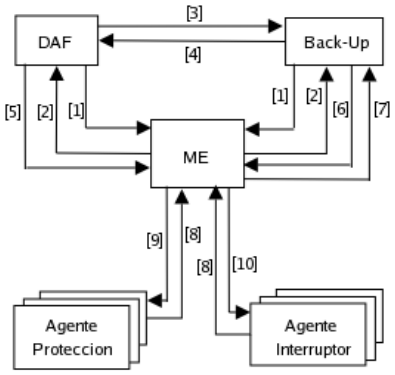

Figura 4: Diagrama de Colaboración.

Id	*Descripción del mensaje*
1	Suscripción por eventos de interés.
2	Informe a los agentes suscriptos de nuevo evento de interés.
3	Suscripción por corrección de lecturas (de relés con falla).
4	Informe a los agentes suscriptos de correcciones.
5	Orden de acción (por ejemplo disparo).
6	Solicitud de información (por ejemplo: estado de un relé).
7	Informe con respuesta a solicitud.
8	Informe con señales o datos propios de componentes eléctricos.
9	Solicitud de bloqueo.
10	Solicitud de disparo o cierre.

Figura 5: Mensajes entre agentes.

de análisis como por ejemplo, los agentes DAF y Back-Up. También se deben definir los eventos generados por estos últimos. Los agentes se comunican entre sí mediante mensajes. Un lenguaje para la comunicación de agentes (ACL) es un medio adecuado de encapsulación de mensajes para su transporte [6].

La interacción entre agentes puede ser capturada mediante distintas representaciones. Una representación posible está dada por "diagramas de colaboración", otra esta dada por "diagramas secuenciales". Los diagramas secuenciales hacen énfasis en la secuencia cronológica de la comunicación mientras que los diagramas de colaboración hacen énfasis en las asociaciones entre agentes.

La figura 4 y el cuadro de la figura 5 ejemplifican cómo es la interacción entre los agentes. Los números de referencia que se encuentran sobre los líneas de asociación dan una orden a una secuencia general de interacción posible, que está descripta en la tabla adyacente.

Mediante un diagrama secuencial (ver figura 6) se puede permitir una mejor visualización de la forma en que interactúan o colaboran los agentes en el caso particular en que el agente DAF (suscripto al servicio de corrección de lecturas provista por el agente Back-Up) ordena la apertura de un interruptor con falla interna. En este tipo de diagramas, se puede leer la secuencia temporizada de interacción desde arriba hacia abajo (o sea, el tiempo se incrementa hacia abajo). En este caso se simplificó la notación agrupándose a los Agentes de Protección y Agentes de Interrupción dentro de SEP.

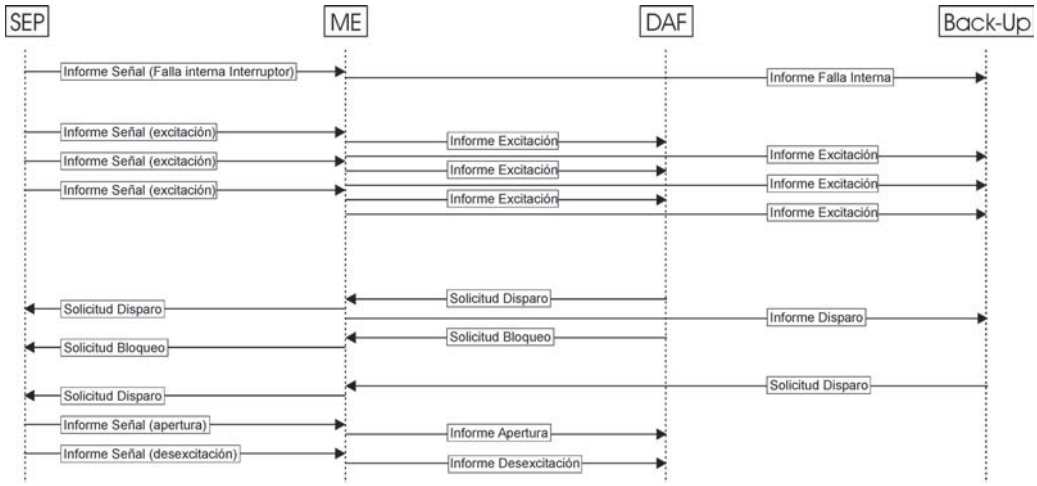

Figura 6: Diagrama secuencial. Presencia de falla interna en interruptor.

A partir de la arquitectura de este sistema multiagente denominado SIC, en la próxima sección se avanza en el diseño e implementación de dos de sus agentes: el agente DAF y el agente Back-Up.

4. AGENTES DAF Y BACK-UP

En esta sección se describen los puntos relevantes del diseño e implementación de los agentes DAF y Back-Up. Del proceso de adquisición del conocimiento resulta natural pensar en un conjunto o jerarquía de procedimientos para determinar cómo estos agentes llevarán a cabo sus tareas. Este proceso deliberativo para decidir qué acción deberá tomar el agente en cada caso, se puede trasladar a la toma de decisión en torno a las intenciones en un sistema intencional. Además, los agentes DAF y Back-Up deben combinar acciones de naturaleza reactiva y proactiva, por lo cual se decide modelizarlos bajo el paradigma BDI, utilizando una arquitectura PRS.

La arquitectura PRS[4] (originalmente desarrollada Georgeff y Lansky [12]) fue quizás la primera arquitectura basada en el paradigma BDI y se ha convertido en una de las más conocidas. Ha sido utilizada en varias aplicaciones reales (e.g. [9], [13]). Entre las ventajas principales de PRS se destacan: la representación de planes y procedimientos, la posibilidad de utilizar planes parciales, comportamiento Reactivo y Proactivo, y la capacidad de Meta-Razonamiento, es decir planes que razonan o gestionan otros planes.

Durante los últimos años se ha construido un gran número de plataformas o entornos para el desarrollo de agentes y en particular de PRS. De las plataformas existentes se buscaron aquellas de código abierto[5] y escritas fundamentalmente en lenguaje C. Esta última preferencia se basa en que el código C ha mostrado no tener inconvenientes bajo restricciones de tiempo real duro. La plataforma de desarrollo elegida resultó ser OpenPRS (OPRS) [4], una versión de código abierto de PRS [5].

4.1. Sistema de Razonamiento Procedural (PRS)

Un Sistema de Razonamiento Procedural es un conjunto de herramientas y métodos para la representación y ejecución de planes y procedimientos. Una arquitectura PRS consiste básicamente en: (1) una base de datos con las *creencias* actuales sobre el entorno; (2) un conjunto de *objetivos o deseeos* actuales a alcanzar; (3) una librería de planes o *procedimientos*, que describen secuencias particulares de acciones y pruebas que pueden realizarse para alcanzar ciertos objetivos o para reaccionar ante ciertas situaciones; y (4) una *estructura de intenciones*, que consiste de un conjunto ordenado (parcialmente) de todos los planes elegidos para ejecución. Estos componentes y sus interacciones se ilustran en la figura 7.

Un *intérprete* (mecanismo de inferencia) manipula estos componentes. Recibe nuevos eventos y objetivos internos; selecciona un plan apropiado teniendo en cuenta los nuevos eventos, objetivos y creencias; ubica el plan dentro de la estructura de intenciones (grafo); elije un plan (intención) en la estructura y finalmente ejecuta un paso del plan activo. Esto puede resultar en una acción primitiva o en la formulación de un nuevo objetivo. El sistema interactúa con el entorno a través de su base de datos y de las acciones básicas o primitivas.

Los algoritmos utilizados y el ciclo principal que presenta el Intérprete de OPRS, considerando algunos supuestos razonables, permiten garantizar una cota superior sobre el tiempo de reacción. Por ejemplo, se debe considerar un máximo en la frecuencia de llegada de eventos y se debe verificar que la cardinalidad del conjunto de planes aplicables se decrezca monótonamente. Un análisis detallado de esto puede encontrarse en [3].

[4] Procedural Reasoning System

[5] Open Source

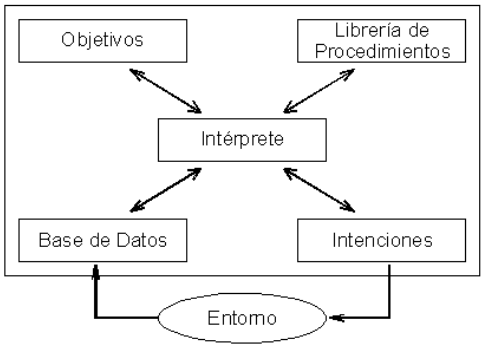

Figura 7: PRS. Una arquitectura basada en el paradigma BDI

4.2. Diseño

Para diseñar un agente con arquitectura PRS es necesario y fundamental especificar sus *creencias* y también su *librería de planes*. Con estos componentes, el *intérprete* (cuya implementación está provista por la plataforma OPRS) genera dinámicamente los componentes de *objetivos* e *intenciones*.

Las creencias de un agente son básicamente una base de datos de formato simple, la cual consiste en una lista de expresiones (predicados). Un ejemplo de archivo que representa parte de la base de datos del agente DAF en un instante dado es:

```
(
(EXCITED R3 1)
(EXCITED R4 1)
(POSITION I3 CLOSED)
(POSITION I4 CLOSED)
(ASSOCIATED-RI R5 I5)
(ASSOCIATED-RI R3 I3)
)
```

Podemos observar entonces que el agente DAF cree o tiene conocimiento de los siguientes hechos: los relés $R3$ y $R4$ se encuentran excitados, los interruptores $I3$ e $I4$ están cerrados y los relés $R3$ y $R5$ están asociados a los interruptores $I3$ e $I5$ respectivamente.

La plataforma OPRS permite especificar los planes mediante una herramienta gráfica. Un ejemplo de plan generado con esta herramienta es la figura 8.

Por último, se puede considerar a cada agente como un núcleo (OPRS Kernel) compuesto por un conjunto de planes y por una base de conocimientos.

Siguiendo la metodología Prometheus [8], una forma de documentar elegantemente el diseño de un agente, consiste en determinar sus *capacidades*. Estas pueden pensarse como módulos dentro del agente y pueden estar anidadas dentro de otras capacidades. En el nivel inferior de anidamiento las capacidades se describen en términos de *eventos internos*, *planes* y *estructuras de datos*. Para fines de documentación, cada capacidad puede ser definida mediante un *Descriptor de Capacidad*, el cual brinda información sobre la interfaz de la capacidad (los eventos que sirven como entrada y los eventos producidos), información sobre interacción con otras capacidades, las capacidades incluidas, y referencias a datos de lectura y escritura.

Las funcionalidades descriptas en la sección 3.1 forman un conjunto inicial de capacidades, el cual puede ser refinado si así se quisiera. A continución se describe una de las capacidades del agente DAF a modo de ejemplo y también uno de los planes que comprende esta capacidad (figura 8).

Nombre: Aislación.

Descripción: Aisla la falla detectada, con posible manejo de incertidumbre.

Eventos de salida: Ordena disparos (apertura de interruptores).
Eventos de entrada: Estado de las protecciones.
Capacidades incluidas: -
Planes: AISLAR-F-B, AISLAR-F-L, AISLAR-F-T, AISLAR-F-TB, APERTURA-CNTRL, APERTURA-REC, APERTURA-VEC, APERTURA-MUL-VEC, ASSOCIATED-LRI, INIT-AIS, INIT-AIS-INCERT, INIT-AIS-INCERT2, OBTENER-RAF-TOTAL
Acceso a Datos: Lectura del estado de las protecciones, topología de red.
Interacciones: Necesita la información sobre la falla (clase y locación) producida por Detector de Fallas.

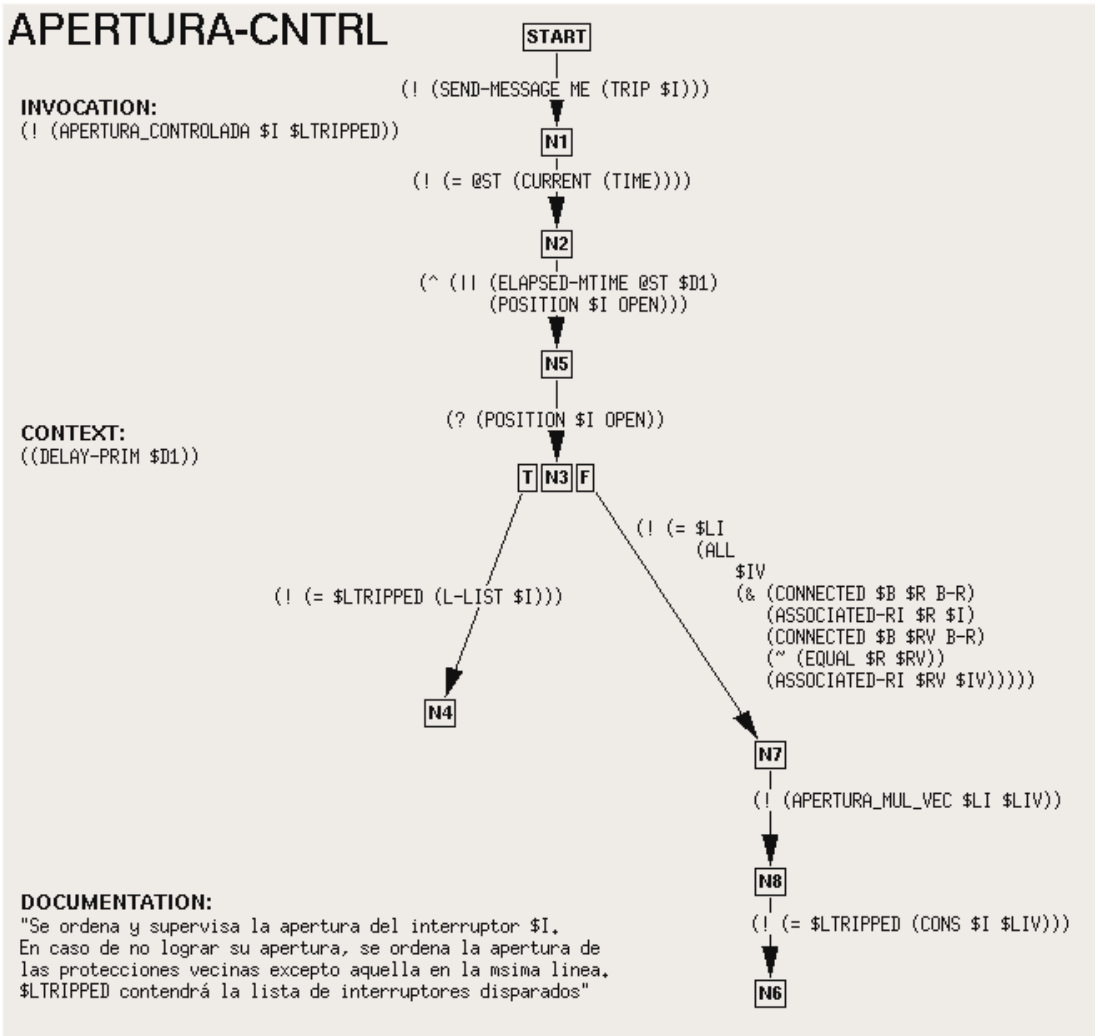

Figura 8: Un plan para controlar la apertura de un interruptor

5. EXPERIMENTACIÓN Y SIMULACIÓN

Se consideraron dos modalidades de prueba. En la primera se supone que el agente DAF parte de un estado en que cuenta con la información correspondiente a situaciones específicas de interés. Basta con enviarle un mensaje de **excitación** del relé que dará inicio a la detección. En la segunda se considera que el agente DAF no tiene la información que describe cada situación sino que el agente

ME envía la información simulando un comportamiento posible del SEP para alcanzar dicha situación de prueba. Todos los casos tratados tuvieron la respuesta esperada.

Por último se destaca que la herramienta OPRS utilizada brinda una interfaz gráfica (X-OPRS) la cual permite interactuar con cada agente, analizar su respuesta ante estímulos manuales y también observar la evolución de las estructuras cognitivas que lo representan. Esta herramienta visual, disminuye notoriamente la performance (tiempos de respuesta), pero esto carece de importancia puesto que la utilización de la interfaz gráfica es opcional y sus fines consisten en permitir una mejor concepción y comprensión del comportamiento del SIC.

6. CONCLUSIONES

En este trabajo se analizó y describió la arquitectura general de un Sistema Inteligente de Control (SIC) para SEPs y la arquitectura propia de dos agentes, el agente de Detección y Aislación de Fallas y el agente de Back-Up. Como resultado se obtiene una arquitectura sumamente flexible, ya que permite realizar cambios en cualquiera de los componentes estudiados en este trabajo, sin afectar de manera importante al resto. Esto vale para el caso del sistema multiagente y también a nivel de agente.

A nivel de sistema multi-agente, se cuenta con características importantes como por ejemplo: cooperación de agentes autónomos, lo cual permite el manejo de tareas independientemente, clara interacción entre los agentes a través de mensajes y la posibilidad de trabajar con información de manera descentralizada.

A nivel de diseño de los agentes, la arquitectura PRS hace que los agentes desarrollados posean cualidades muy ventajosas. En primer lugar su incrementabilidad, pues permite muy fácilmente satisfacer nuevos requerimientos o cambios en los mismos. Además los procedimientos gozan de una representación muy ventajosa ya que pueden ser parciales, delegando al intérprete la responsabilidad de satisfacer los subobjetivos que estos requieran alcanzar y permitiendo un refinamiento en las tareas. La noción de objetivo es fuerte en PRS, dado que el intérprete considera todos los procedimientos que unifiquen con él para alcanzarlo, antes de considerar que falla. Por otra parte, la combinación de comportamiento proactivo y reactivo (en tiempo acotado) es fundamental para este desarrollo, pues se necesita comportamiento reactivo cada vez que el SEP de indicaciones de fallas potenciales y se requiere comportamiento proactivo para realizar por ejemplo, tareas de detección bajo incertidumbre. Otra ventaja es la posibilidad de controlar parte del ciclo principal mediante meta-razonamiento, utilizando meta-planes. Por último, se destaca que como resultado final de la etapa de diseño se obtiene directamente la implementación del sistema.

REFERENCIAS

[1] Casali A., Godo L., and Sierra C. A methodology to engineer graded bdi agents. In *WASI - CACIC Workshop.XII Congreso Argentino de Ciencias de la Computación*, 2006.

[2] Phadke A.G. and Horowitz S.H. Adaptive relaying. *IEEE Computer Applications in Power*, 3(3):47–51, 1990.

[3] Ingrand F. and Coutance V. Real-time reasoning using procedural reasoning. Technical report 93-104, LAAS/CNRS, Toulouse, France, 1993.

[4] Ingrand F.F. OPRS development environment, 2004. http://softs.laas.fr/openrobots/php/download.php/openprs-1.0b1-doc.pdf.

[5] Ingrand F.F., Georgeff M.P., and Rao A.S. An architecture for real-time reasoning and system control. *IEEE Expert: Intelligent Systems and Their Applications*, 7(6):34–44, 1992.

[6] FIPA. *FIPA ACL Message Structure Specification*. FIPA, 2001.

[7] Brazier F.M.T, Dunin-Keplicz B.M., Jennings N.R., and Treur J. DESIRE: Modelling multi-agent systems in a compositional formal framework. *Int Journal of Cooperative Information Systems*, 6(1):67–94, 1997.

[8] Padgham L. and Winikoff M. Prometheus: A methodology for developing intelligent agents. In *AAMAS 2002*, Bologna, Italy, July 2002.

[9] Ljungberg M. and Lucas A. The OASIS air-traffic management system. In *PRICAI '92*, Seoul, Korea, 1992.

[10] Wooldridge M. *An Introduction to Multi-Agent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[11] Wooldridge M., Jennings N.R., and Kinny D. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.

[12] Georgeff M.P. and Lansky A.L. Reactive reasoning and planning. In *Proc. of AAAI-87*, pages 677–682, Seattle, WA, 1987.

[13] Georgeff M.P. and Ingrand F.F. Monitoring and control of spacecraft systems using procedural reasoning. Technical Report 03, Australian Artificial Intelligence Institute, Melbourne, Australia, November 1989.

[14] McArthur S.D.J., Davidson E.M., Hossack J.A., and McDonald J.R. Automating power system fault diagnosis through multi-agent system technology. In *HICSS*, 2004.

[15] Thorp J.S. Giovanini R. Birman K. Coury D. Wang X.R., Hopkinson K.M. Developing an agent-based backup protection system for transmission networks. *Power Systems and Communications Infrastructures for the Future, Beijing, September 2002*.