

# Integración de la visión en un sistema de automatización global

Autor: Octavi Estapé

Tutor: Giuseppe Menga

Doble titulación Ingeniería de Telecomunicación: Politecnico di Torino y ETSETB (UPC)

Abril 2002

La difusión de Internet en los últimos años ha extendido ampliamente la utilización de los sistemas de automatización industrial. Los principios de estos sistemas de automatización, que en sus orígenes estaban solamente en las celdas de manufactura de los talleres de las fabricas, fueron llevados fuera de ellas y de esta manera, se abrió la posibilidad de integrar y hacer interaccionar sistemas pertenecientes a dominios diferentes (por ejemplo, la integración de la cadena logística entre empresas). Bajo esta visión se desarrolló **Global Automation Platform (GAP)**, una arquitectura software capaz de gestionar de forma uniforme y dinámica la integración de diferentes sistemas de automatización.

GAP ofrece los instrumentos necesarios para crear un canal de comunicación en Internet, entre diferentes dispositivos y sistemas que deben cooperar para llevar a cabo su actividad. También proporciona los mecanismos de oferta y contratación de servicios entre los diferentes módulos de control, en un ambiente distribuido. Su aplicación no atañe a un ámbito específico de acción, es decir, GAP puede gestionar servicios de naturaleza muy variada.

Este proyecto trata el argumento de integrar sistemas de visión (captura, transmisión y procesado de imagen), como servicios que puedan ser ofrecidos en GAP. En el ámbito industrial estos servicios son utilizados en aplicaciones muy diferentes, como cámaras de vigilancia, control remoto de procesos, control automático de robots, controles de calidad y videoconferencias.

El objetivo específico de este trabajo es valorar a través de la creación de prototipos, el uso de los instrumentos ofrecidos por **Java** para la creación de servicios de transmisión y procesado de imagen en GAP.

Java ofrece diversas características que lo hacen especialmente apto para un ambiente como la plataforma de automatización global: Java es seguro, escalable, independiente de la plataforma, e ideado para la red. Se han estudiado dos tecnologías diferentes proporcionadas por Sun Microsystems para la gestión de la imagen: **Java Media Framework** (para la captura, procesado y transmisión de datos de audio y video en tiempo real) y **Java Advanced Imaging** (para el procesado avanzado de imágenes independientes).

Se han realizado diferentes prototipos para identificar las problemáticas de los sistemas de transmisión y procesado de imagen en tiempo real usando la tecnología Java. Dos de los prototipos realizados permiten capturar un flujo de video de una cámara y procesarlo en tiempo real, uno utiliza Java Advanced Imaging y el otro no. Estos prototipos se han testado para en control de un robot móvil que debía reaccionar según el movimiento de los objetos definidos por el usuario, por ejemplo, para seguir un objeto de un color predefinido. El tercer prototipo permite transmitir un flujo de vídeo (proveniente de un archivo o capturado por una cámara) en tiempo real a través de una red.

Con la realización de los prototipos, se ha visto que Java Media Framework (JMF) es un buen punto de partida para implementar cualquier tipo de servicio para el que sean necesarias las fases de captura, procesado y transmisión de imagen. JMF es suficientemente rápido, flexible y extensible, y soporta una gran variedad de formatos de captura y almacenamiento de datos de audio y video. El protocolo utilizado por esta arquitectura para la transmisión de flujos de audio y vídeo es el estándar de Internet Real-Time

Transport Protocol (RTP). Actualmente JMF (version 2.1.1) soporta solamente los formatos JPEG y H.263 para la transmisión de vídeo sobre RTP, pero debido a la continua evolución en esta área, probablemente las versiones sucesivas incluirán nuevos formatos. Con los formatos actuales se pueden realizar comunicaciones de vídeo en tiempo real con calidad y velocidad de transmisión configurables por el usuario. La ventaja de la arquitectura de JMF, bajo este punto de vista, es que las aplicaciones existentes podrán utilizar cualquier nuevo formato, sin tener que someterlas a cambios importantes.

Java Advanced Imaging (JAI), es una librería bastante completa y potente para el procesamiento de imágenes sueltas (con operaciones avanzadas como la deformación de la imagen y operaciones en el dominio de la frecuencia). Permite realizar cualquier tipo de procesamiento usando una cadena de operaciones básicas, que están optimizadas para aprovechar las peculiaridades de cada plataforma y acelerar el procesamiento con código nativo.

Después de la realización del prototipo se ha visto que JAI no es apto para el procesamiento continuo de imágenes. El inconveniente de JAI en este caso es que genera una imagen nueva para cada operación. Esta imagen se almacena en un nuevo espacio de memoria, y no existe la posibilidad de reutilizar el espacio de imágenes precedentes. Para liberar la memoria se usa el mecanismo estándar de Java: el garbage collector. Algunas ejecuciones del garbage collector bloquean el funcionamiento de toda la aplicación por un tiempo proporcional a la memoria ocupada, sin que haya ningún método para evitarlo o ejecutar el garbage collector en un proceso separado. El tiempo mínimo de estas pausas es del orden de los 0,3 segundos, pero dependiendo de algunos parámetros puede ser mucho superior. Por ejemplo, con los parámetros de ejecución habituales, el garbage collector se ejecuta después del procesamiento de cada 35 fotogramas y provoca una pausa de 0,69s. Modificando los parámetros de ejecución de Java se pueden obtener pausas más cortas pero más frecuentes o viceversa. Los desarrolladores de JAI están valorando la posibilidad de incluir la opción de reutilizar el espacio de memoria de imágenes precedentes para almacenar las nuevas (así se resolvería el problema). Aparte del problema de las pausas, la velocidad máxima de procesamiento no supera los 7 fotogramas por segundo. JAI optimiza cada operación básica, pero para el procesamiento de interés se ha debido construir una cadena bastante larga de operaciones simples, que hacen que el procesamiento total sea lento.

Las aplicaciones basadas en JAI tienen que ser capaces de tolerar pausas relativamente largas en el procesamiento (del orden de un segundo). JAI se podría usar para aplicaciones con procesamiento a baja velocidad o que no sean en tiempo real. Por ejemplo, podría usarse de modo ventajoso para algunos tipos de aplicaciones para el control de calidad para las que sea necesaria un procesamiento de mucha precisión que tolere pausas. Para aplicaciones de control de robots móviles, el uso de JAI podría acarrear accidentes impredecibles (durante las pausas, el robot funcionaría sin información de control).

Se ha implementado otro prototipo del sistema de procesamiento de imagen que utiliza Java puro, para compararlo con el que utiliza JAI. Se ha implementado una sola operación que realiza el procesamiento de interés (con calidad ligeramente inferior a la de JAI). El problema de este enfoque es que el código no es fácilmente aprovechable para otro tipo de procesamiento. Este prototipo basado en Java puro ha demostrado ser mejor, porque consigue detectar a la vez objetos de diferentes colores, sin pausas, y a una velocidad de hasta 24 fotogramas por segundo. Otra alternativa no analizada, pero que podría ser interesante, es la implementación de las mismas operaciones en C, que deberían ser más rápidas, pero menos portables.

Los desarrollos futuros basados en la tecnología estudiada pueden llevar a la implementación de un conjunto de clases para la definición de servicios de transmisión y procesamiento de vídeo dentro de la plataforma de automatización global. En la parte final de este proyecto se han descrito los detalles implementativos de tales servicios, teniendo en cuenta la experiencia adquirida en el desarrollo de los prototipos.