

Alto nivel en *Microcontroladores e Internet*

13 de mayo de 2009

Avrak

Presente

Introducción

Motor SCADA

GUI

Futuro

Definición del alto nivel

Los principales del software de alto nivel son:

- ▶ Recolección de datos de las RTUs
 - ▶ Tiempo real

Definición del alto nivel

Los principales del software de alto nivel son:

- ▶ Recolección de datos de las RTUs
 - ▶ Tiempo real
 - ▶ Histórico (almacenamiento)

Definición del alto nivel

Los principales del software de alto nivel son:

- ▶ Recolección de datos de las RTUs
 - ▶ Tiempo real
 - ▶ Histórico (almacenamiento)
- ▶ Representación visual del campo.

Definición del alto nivel

Los principales del software de alto nivel son:

- ▶ Recolección de datos de las RTUs
 - ▶ Tiempo real
 - ▶ Histórico (almacenamiento)
- ▶ Representación visual del campo.
- ▶ Interacción con el campo. (Envío de comandos)

El protocolo

Con ustedes...

El protocolo

Con ustedes...

Qty	Dst	Src	Sec	Com	Dato	BCCH	BCCL
1	1	1	1	1	0-245	1	1

El protocolo

Con ustedes...

Qty	Dst	Src	Sec	Com	Dato	BCCH	BCCL
1	1	1	1	1	0-245	1	1

uCnet

El protocolo

Con ustedes...

Qty	Dst	Src	Sec	Com	Dato	BCCH	BCCL
1	1	1	1	1	0-245	1	1

uCnet

Denominamos SCADA al software encargado de dialoga con las RTUs.

- ▶ Su misión es recolectar datos (a través de polling),
- ▶ almacenarlos,
- ▶ hacerlos disponibles para alguna aplicación (GUI)
- ▶ y ejercer algún control (enviar comandos).

Requisitos secundarios

- ▶ Cross Platform (correr sobre varios SOs)

Requisitos secundarios

- ▶ Cross Platform (correr sobre varios SOs)
- ▶ Debe ser posible su utilización como CLI (posibilidad de correr sin GUI) *complejo desarrollar una GUI genérica*

Requisitos secundarios

- ▶ Cross Platform (correr sobre varios SOs)
- ▶ Debe ser posible su utilización como CLI (posibilidad de correr sin GUI) *complejo desarrollar una GUI genérica*
- ▶ Agnóstico a la base de datos

Requisitos secundarios

- ▶ Cross Platform (correr sobre varios SOs)
- ▶ Debe ser posible su utilización como CLI (posibilidad de correr sin GUI) *complejo desarrollar una GUI genérica*
- ▶ Agnóstico a la base de datos
- ▶ Soportar futuras versiones de uCnet u otros protocolos con *backward compatibility*

Elección de la tecnología en función de los requisitos

En función a los requisitos anteriores,

1. Python es cross platform. Corre en varios sistemas operativos.

Elección de la tecnología en función de los requisitos

En función a los requisitos anteriores,

1. Python es cross platform. Corre en varios sistemas operativos.
2. Python es un lenguaje de scripting, por lo cual aliviana el desarrollo, además puede ser *freezado* como un binario propio de la plataforma

Elección de la tecnología en función de los requisitos

En función a los requisitos anteriores,

1. Python es cross platform. Corre en varios sistemas operativos.
2. Python es un lenguaje de scripting, por lo cual aliviana el desarrollo, además puede ser *freezado* como un binario propio de la plataforma
3. Python se presta para aplicaciones CLI, tiene una consola rica que puede ser explotada para extender aplicaciones.

Elección de la tecnología en función de los requisitos

En función a los requisitos anteriores,

1. Python es cross platform. Corre en varios sistemas operativos.
2. Python es un lenguaje de scripting, por lo cual aliviana el desarrollo, además puede ser *freezado* como un binario propio de la plataforma
3. Python se presta para aplicaciones CLI, tiene una consola rica que puede ser explotada para extender aplicaciones.
4. Python cuenta muchos ORMs y proyectos similares, por lo cual se puede lograr cierta independencia de la base de datos.

Elección de la tecnología en función de los requisitos

En función a los requisitos anteriores,

1. Python es cross platform. Corre en varios sistemas operativos.
2. Python es un lenguaje de scripting, por lo cual aliviana el desarrollo, además puede ser *freezado* como un binario propio de la plataforma
3. Python se presta para aplicaciones CLI, tiene una consola rica que puede ser explotada para extender aplicaciones.
4. Python cuenta muchos ORMs y proyectos similares, por lo cual se puede lograr cierta independencia de la base de datos.

Alunos detalles importantes

- ▶ Python es un lenguaje multiparadigma (Procedural, Objetos, Funcinal) de alto nivel donde *todo es un objeto*.

¹<http://docs.python.org/c-api/init.html>

Alunos detalles importantes

- ▶ Python es un lenguaje multiparadigma (Procedural, Objetos, Funcional) de alto nivel donde *todo es un objeto*.
- ▶ El intérprete de Python posee un bloqueo global sobre los el contexto de ejecución, cada hilo debe adquirir este lock para poderse ejecutar. Por defecto este lock se libera cada 100 instrucciones de máquina virtual.¹

¹<http://docs.python.org/c-api/init.html>

Alunos detalles importantes

- ▶ Python es un lenguaje multiparadigma (Procedural, Objetos, Funcional) de alto nivel donde *todo es un objeto*.
- ▶ El intérprete de Python posee un bloqueo global sobre los el contexto de ejecución, cada hilo debe adquirir este lock para poderse ejecutar. Por defecto este lock se libera cada 100 instrucciones de máquina virtual.¹
- ▶ El intérprete no puede aprovechar los hilos del sistema operativo anfitrión, aunque se puede lograr paralelismo utilizando procesos.

¹<http://docs.python.org/c-api/init.html>

Twisted²

Twisted es un framework de red orientad a aventos escrito en Python. Sus ideas fundamentales son

²<http://twistedmatrix.com/>

Twisted²

Twisted es un framework de red orientad a aventos escrito en Python. Sus ideas fundamentales son

Separación de Protocolos y transporte

Twisted fue diseñado de forma tal que existe una separación lógica entre el prtocolo de aplicación (como POP3, HTTP, IRC, etc.) y el transporte subyacente (archivos, sockets o conexiones mediante librerías SSL).

²<http://twistedmatrix.com/>

Twisted²

Twisted es un framework de red orientad a aventos escrito en Python. Sus ideas fundamentales son

Separación de Protocolos y transporte

Twisted fue diseñado de forma tal que existe una separación lógica entre el prtocolo de aplicación (como POP3, HTTP, IRC, etc.) y el transporte subyacente (archivos, sockets o conexiones mediante librerías SSL).

Deferreds

El concepto de Deferred (a veces llamado future o promesa) se refiere a un valor que aún no ha sido computado. Por ejemplo, por que necesita datos de un peer remoto. Un Deferred puede ser pasado entre funciones, pero no se puede preguntar su valor de manera directa.

²<http://twistedmatrix.com/>

Twisted - Deferreds (cont)

A cada Deferred se le puede agergar una cadena de callbacks, siendo la salida de cada callback la entrada del siguiente. A cada callback se puede añadir un errback, que se ejecuta en caso de ocurrir una excepción.

Twised - Deferreds (cont)

A cada Deferred se le puede agergar una cadena de callbacks, siendo la salida de cada callback la entrada del siguiente. A cada callback se puede añadir un errback, que se ejecuta en caso de ocurrir una excepción.

Soporte de Threads

Twisted propone una abstracción sobre los hilos tradicionales, utilizando un thread como fuente para un Deferred. El Deferred recuperará el valor tan pronto el thread termine. Se pueden agregar callbacks a este Deferred que serán ejecutados en el hilo principal.

Twisted - Deferreds (cont)

A cada Deferred se le puede agergar una cadena de callbacks, siendo la salida de cada callback la entrada del siguiente. A cada callback se puede añadir un errback, que se ejecuta en caso de ocurrir una excepción.

Soporte de Threads

Twisted propone una abstracción sobre los hilos tradicionales, utilizando un thread como fuente para un Deferred. El Deferred recuperará el valor tan poronto el thread termine. Se pueden agregar callbacks a este Deferred que serán ejecutados en el hilo principal.

Integración con bucles de ejecución de otras librerías Twisted se puede integrar con loops de eventos de otras librerías como las de GTK+, Qt y Cocoa(utilizando PyObjC).

Twisted - Foreign Loops (cont)

Esto permite utilizar twisted para aplicaciones con GUI utilizando todas sus librerías sin agregar el típico overhead de un hilo por socket. Se puede montar un servidor web completo integrado con una aplicación visual aprovechando esta característica.

³http://en.wikipedia.org/wiki/Reactor_pattern

⁴[http://en.wikipedia.org/wiki/Select_\(Unix\)](http://en.wikipedia.org/wiki/Select_(Unix))

Twisted - Foreign Loops (cont)

Esto permite utilizar twisted para aplicaciones con GUI utilizando todas sus librerías sin agregar el típico overhead de un hilo por socket. Se puede montar un servidor web completo integrado con una aplicación visual aprovechando esta característica.

El reactor³ de twisted es un scheduler que utiliza *select()*⁴. Las cadenas de callbacks de los Deferreds y los timers son manejados por el reactor.

Menos problemas de sincronización.

³http://en.wikipedia.org/wiki/Reactor_pattern

⁴[http://en.wikipedia.org/wiki/Select_\(Unix\)](http://en.wikipedia.org/wiki/Select_(Unix))

Twisted

Ejemplo simple de echo

```
from twisted.internet import protocol, reactor
class MiEchoProtocol(protocol.Protocol):
    def dataReceived(self, data):
        self.transport.write( data )
class ClientFactory( protocol.ClientFactory )
    protocol = MiEchoProtocol
reactor.connectTCP( '127.0.0.1', 5432, ClientFactory())
```

Almacenamiento

SQLAlchemy es un ORM para Python. Basicamente asociar registros a objetos y encapsular el DML⁵. Abstrae en Python el DDL⁶.

A través del módulo dbapi de la librería estándar de Python soporta un buen número de bases de datos.

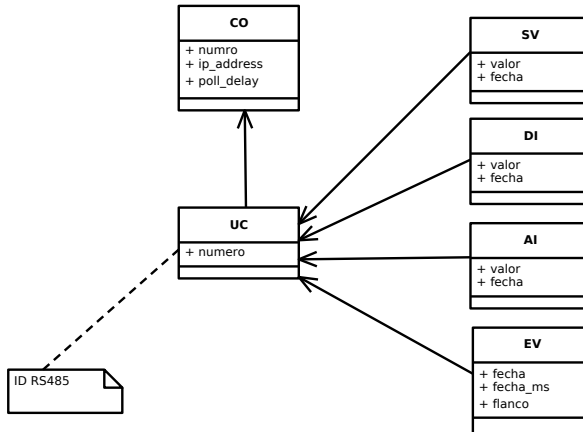
Cambiar la base de datos es cuestión de cambiar la URL de conexión. *tipo://user:pass@host/base/*

⁵<http://es.wikipedia.org/wiki/DML>

⁶http://es.wikipedia.org/wiki/Lenguaje_de_definici%C3%B3n_de_datos

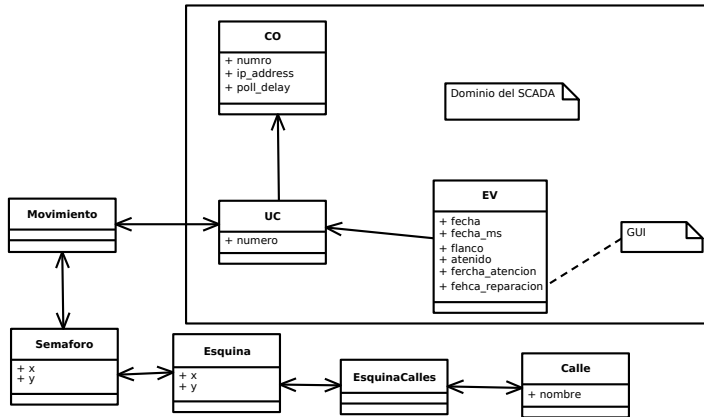
Almacenamiento

En la base de datos se cuentan con las siguientes tablas relacionadas con la adquisición de datos



Almacenamiento

Esquema final de la base de datos para el sistema de semaforización:



Cada entrada se define como punto y tiene una dirección dada por

Almacenamiento - Punto

Punto Cada entrada se define como punto y tiene una dirección dada por el CO, la UC, el tipo y número de puerto y número de bit si se aplica.

Puerto En el software de la UC se creo un mapeo secuencia de puertos de “alto nivel” a los puertos físicos. De esta forma se presentan como un vector uniforme ante la aplicación de alto nivel.

Movimiento En el sistema de semáforos asociaba 2 bits a una conjunto de lámparas. Esta entidad es la abstracción de alto nivel de una “entidad” del hardware que puede estar trabajando sobre uno o más bits o bytes. Arreglo de puntos.

GUI

Qt es un toolkit de desarrollo de aplicaciones para C++. PyQt4 es un *binding* para Python de este toolkit. Qt presenta algunas características interesantes:

- ▶ Módulo **QtSql**, interacción con bases de datos ala RAD (Access, VB, Delphi, etc.)
- ▶ Módulo **QtGui**, conjunto de widgets con un subframework de gráficos en el cual existe una escena, un visor y un conjunto de items.
 - ▶ Tanto la escena, el visor o los items son “vectoriales”, se les pueden aplicar matrices de transformación.
 - ▶ Movimiento con el mouse sencillo
 - ▶ Capas

Qt partes no exploradas

- ▶ Módulo **QtScript** permite hacer scripting de la aplicación mediante ECMAScript (Javascript).
- ▶ Módulo **QtSvg**, renderización de gráficos vectoriales.
- ▶ Módulo **QtWebKit**, motor de renderización de HTML WebKit (el motor de Safari (browser de MacOS, iPhone). Posibilidad de integración con WMS⁷
- ▶ Módulo **QtOpenGL**, 3D.
- ▶ Extensiones **PyQwt**, gráficos estadísticos.

⁷http://es.wikipedia.org/wiki/Web_Map_Service

Nueva arquitectura

La interfase web todavía no ha sido explorada. La web
sestá sufriendo importantes cambios que apuntan a una mejor
interactividad.

Nueva arquitectura

La interfase web todavía no ha sido explorada. La web se está sufriendo importantes cambios que apuntan a una mejor interactividad.

Aprovechando la ubicuidad de HTTP y planteando un esquema de RPC (en general en HTTP se lo suele llamar “web service”) se puede ganar generalidad en el SCADA.

Nueva arquitectura

La interfase web todavía no ha sido explorada. La web se está sufriendo importantes cambios que apuntan a una mejor interactividad.

Aprovechando la ubicuidad de HTTP y planteando un esquema de RPC (en general en HTTP se lo suele llamar “web service”) se puede ganar generalidad en el SCADA.

Los mecanismos de codificación de información más populares sobre HTTP son XML y JSON. Existe un mecanismo de RPC que se basa en cada codificación

XMLRPC

Invocación desde el cliente

```
<?xml version="1.0"?>
<methodCall>
  <methodName>org.wikipedia.intercambioDatos</methodName>
  <params>
    <param>
      <value><i4>360</i4></value>
    </param>
    <param>
      <value><i4>221</i4></value>
    </param>
  </params>
</methodCall>
```

XMLRPC - Respuesta

Respuesta desde el servidor

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>Intercambio datos nro.
        360 por 221</string></value>
    </param>
  </params>
</methodResponse>
```

JSONRPC 1.0

Petición del cliente

```
{ "method": "echo", "params": ["Hello JSON-RPC"], "id": 1 }
```

Respuesta del servidor

```
{ "result": "Hello JSON-RPC", "error": null, "id": 1 }
```

Existe una notable diferencia entre la cantidad de datos enviados por llamadas RPC similares

Scada Cliente Servidor

HTTP/HTTPS como capa de “transporte” para RPC:

- ▶ Es estándar
- ▶ HTTPS contempla seguridad
- ▶ HTTP PUSH⁸ o streaming (socket abierto).

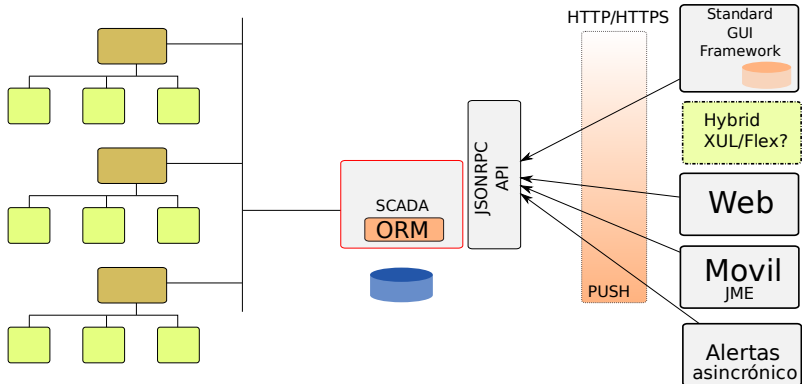
JSON como codificación de datos

- ▶ Es leible por el humano
- ▶ Los browsers están adquiriendo capacidades de paresos nativo.
- ▶ Contempla comunicación semi-P2P.

La comunicación entre los clientes (no solo GUIs) y el servidor SCADA se realizan mediante JSONRPC. Se define una API.

⁸http://en.wikipedia.org/wiki/Push_technology

Arquitectura



JSONRPC API

La API JSONRPC permitiría

- ▶ Configuración
- ▶ Consulta de puntos
- ▶ Envío de comandos
- ▶ Comunicación asincrónica P2P (escencial para el módulo alertas)
- ▶ Inicio de sesión (HTTPS?)
- ▶ Control de acceso

El cliente del SCADA se puede escribir en cualquier lenguaje bajo cualquier plataforma.

GUI Framework

Configuración visual DnD.

WEB

Llamadas asincrónicas a la API. Quizás mashups con WMS de Mapserver o Mapnik. En focalizado en función del potencial cliente.

Móviles

Interfase propia de la plataforma, por tamaño del display. Ejeción de javascript inexistente o costosa (*Ej: blackberry*)

Alertas

Ejecución de procesos en función de ciertos estados, tabla de cuando vs que. Envío de mails, envío de SMS, accionamiento de algun PBX, etc.

Otros clientes

Híbridos con Flex, XUL o alguna otra tecnología para GUIs.

Algunos detalles

- ▶ El esquema cliente servidor se presenta inadecuado para estadísticas. ¿RPC sobre un DSL como R? *Se escuchan propuestas!*
- ▶ Obtención de datos de GPS