

MARINET, a service integrated network



Carsten Zerbst

MARINET

Many sensors and other services like radar could be found on board.

MARINET

Many sensors and other services like radar could be found on board.

- nautical devices with NMEA interface

MARINET

Many sensors and other services like radar could be found on board.

- nautical devices with NMEA interface
- stream from radio, radar, video

MARINET

Many sensors and other services like radar could be found on board.

- nautical devices with NMEA interface
- stream from radio, radar, video
- calculation programmes for load, ships stability

MARINET

Many sensors and other services like radar could be found on board.

- nautical devices with NMEA interface
- stream from radio, radar, video
- calculation programmes for load, ships stability
- electronic logbook

MARINET

Many sensors and other services like radar could be found on board.

- nautical devices with NMEA interface
- stream from radio, radar, video
- calculation programmes for load, ships stability
- electronic logbook
- ...

There is no common standard for service integration.

There is no common standard for service integration.

The lack of this standard makes things

expensive more work to integrate things

unsure no automatical survey possible

difficult to build services upon other services

There is no common standard for service integration.

The lack of this standard makes things

expensive more work to integrate things

unsure no automatical survey possible

difficult to build services upon other services

MARINET is an attempt to create a service integrated network

Design Principles

CORBA [1] is used as underlying middleware. It is therefore OS, programming language and hardware independent.

Design Principles

CORBA [1] is used as underlying middleware. It is therefore OS, programming language and hardware independent.

Classes

Sensor operations to get data

Design Principles

CORBA [1] is used as underlying middleware. It is therefore OS, programming language and hardware independent.

Classes

Sensor operations to get data

Calculation operations to send and receive data

Design Principles

CORBA [1] is used as underlying middleware. It is therefore OS, programming language and hardware independent.

Classes

Sensor operations to get data

Calculation operations to send and receive data

Log operations to send and retrieve data

Design Principles

CORBA [1] is used as underlying middleware. It is therefore OS, programming language and hardware independent.

Classes

Sensor operations to get data

Calculation operations to send and receive data

Log operations to send and retrieve data

Data Transport

MARINETs aim is to transport small data, e.g. NMEA sequences. Completion for data streams in the near future is sensible.

This data is encapsulated in a sequence of unions.

```
enum DataTypeT {DTLONG, DTDOUBLE, DTBOOLEAN, DTSTRING, DTTIME};  
union DataT switch (DataTypeT) {  
    case DTBOOLEAN:    boolean booleanval;  
    case DTDOUBLE:     double doubleval;  
    case DTLONG:       long longval;  
    case DTSTRING:     string stringval;  
    case DTTIME:       TimeBase::TimeT timeval;  
};  
  
typedef sequence<DataT> DataListT;
```

The unions provide many advantages over the any type.

There is a second list describing the contents of the `DataListT`

```
enum ReferenceT {
    RTAbsolute,
    RTRelative,
    RTUnknown
};

struct DataInfoT {
    DataTypeT  typecode;
    string    datatype;
    string    unit;
    ReferenceT referencetype;
    string    referencelocation;
};

typedef sequence<DataInfoT> DataInfoListT;
```

This gives us a generic way to map arbitrarily data into IDL (except of streams), on the other hand on could easily gather information about the transported data.

Property, Configuration and Location

A subset of the Property Service [4] is used to get information about the objects and configure them.

Name	Type	Modus
Location	string	Fixed_Readonly / _Normal
DeviceType	string	Fixed_Readonly
Vendor	string	Fixed_Readonly
...		

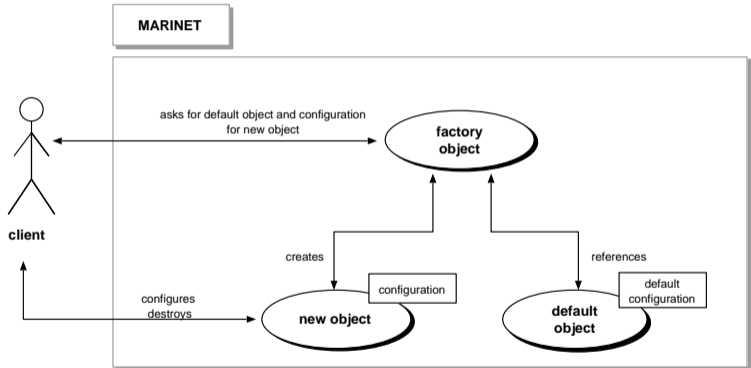
Property, Configuration and Location

A subset of the Property Service [4] is used to get information about the objects and configure them.

Name	Type	Modus
Location	string	Fixed_ReadOnly / _Normal
DeviceType	string	Fixed_ReadOnly
Vendor	string	Fixed_ReadOnly
...		

A naming scheme is defined on base off some properties, the naming service may be used to find an object [3]. They are advertised to the trading service [5], so one may search for an object by its properties.

Most classes use the factory pattern.



Event Management

All data sources may connect to an event service [2]. This pushes data to all connected clients.

The advantages of the using the event service

- + the data source are encapsulated from clients
- + no polling necessary
- + works as broadcast

Event Management

All data sources may connect to an event service [2]. This pushes data to all connected clients.

The advantages of the using the event service

- + the data source are encapsulated from clients
- + no polling necessary
- + works as broadcast
- external service needed

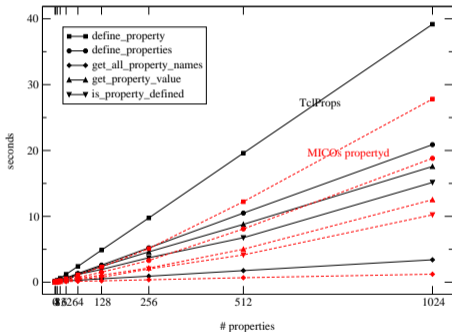
Implementation

Combat (former tclmico) [6] was used for a rapid prototype implementation.

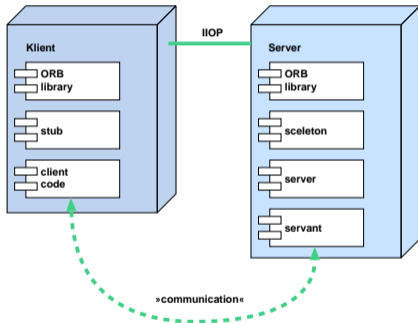
The fast development allowed to test different design ideas.

Tcl string facilities are eg. usefull to couple with devices with NMEA interface. Theses require to map string representation into MARINET.

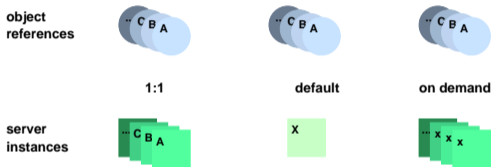
The performance of Tcl is good compared to pure C++



Anatomy of a CORBA application



There are many possibilities to map the external object reference onto a real class instance.



This makes CORBA the ideal glue between physical devices and the network.

SYSTEM_ID, UNIQUE_ID

Server code

```
set sensorserv [  
  ::MARINET::Base Base \  
  $rootpoa $propertyDefs  
]  
  
set ref [$sensorserv _this]
```

USER_ID, MULTIPLE_ID

```
set mypoa [$rootpoa create_POA mypoa $mgr \  
  [list USER_ID MULTIPLE_ID]]  
set basedserv [::MARINET::BaseD BaseD $mypoa]  
set oid "TheFirstObjectID"  
$mypoa activate_object_with_id $oid $basedserv  
set ref [$mypoa id_to_reference $oid]
```

SYSTEM_ID, UNIQUE_ID

Server code

```

set sensorserv [
  ::MARINET::Base Base \
  $rootpoa $propertyDefs
]

set ref [$sensorserv _this]

```

USER_ID, MULTIPLE_ID

```

set mypoa [$rootpoa create_POA mypoa $mgr \
  [list USER_ID MULTIPLE_ID]]
set basedserv [::MARINET::BaseD BaseD $mypoa]
set oid "TheFirstObjectID"
$mypoa activate_object_with_id $oid $basedserv
set ref [$mypoa id_to_reference $oid]

```

Servant Code

```

public method get_props {} {
  set props {}
  set id [namespace tail $this]
  . . .
  return $props
}

```

```

public method get_props {} {
  set props {}
  set id [$poa reference_to_id [$this _this]]
  . . .
  return $props
}

```

References

- [1] Object Management Group. *CORBA for Beginners Page*. <http://www.omg.org/corba/beginners.html>.
- [2] Object Management Group. *Event Service Specification*, März 1995. <http://www.omg.org/cgi-bin/doc?formal/97-12-11.pdf>.
- [3] Object Management Group. *Naming Service Specification*, März 1995. <ftp://ftp.omg.org/pub/docs/formal/97-12-10.pdf>.
- [4] Object Management Group. *Property Service*, Juli 1996. <ftp://ftp.omg.org/pub/docs/formal/97-12-20.pdf>.
- [5] Object Management Group. *Trading Object Service Specification*, März 1997. <ftp://ftp.omg.org/pub/docs/formal/97-12-23.pdf>.
- [6] Frank Pillhofer. *TclMico*. Goethe Universität Frankfurt, Informatik, 1999. <http://www.informatik.uni-frankfurt.de/~fp/Tcl>.