EviNS: A Framework for Development of Underwater Acoustic Sensor Networks and Positioning Systems

Oleksiy Kebkal

EvoLogics GmbH, Berlin, Germany

Erlang User Conference, Stockholm

Introduction

Applications

Motivation

Architecture

Example

Summary

Outline

Introduction

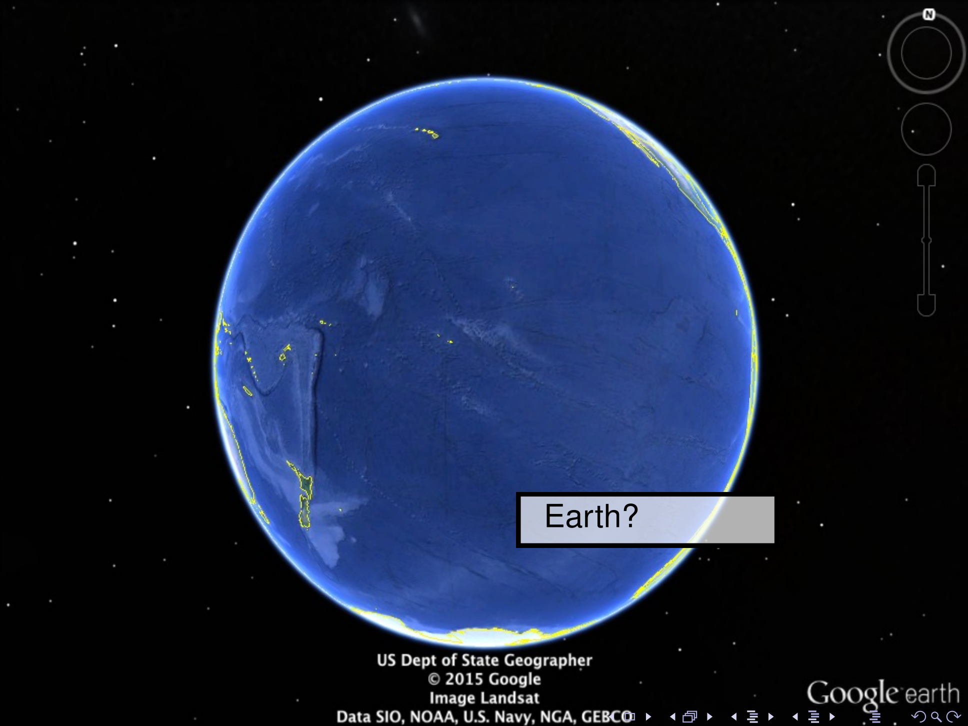
Applications

Motivation

Architecture

Example

Summary


Earth?
US Dept of State Geographer
© 2015 Google
Image Landsat
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google earth

Introduction

Why acoustic waves?

- radio waves: tens of centimeters to a few meters range
- optical waves: affected by scattering, about 100 meters range
- acoustic waves:
  - up to 10 km range depending on frequency range (for frequencies > 10 KHz)
  - can be used both for data transmission and positioning

Introduction
Applications
Motivation
Architecture
Example
Summary

Introduction

Acoustic channel properties

- propagation time: sound speed in water ca. 1500 m/s
- limited frequency range: tens of KHz
- low bitrate: a few kbps, short range a few tens kbps
- half duplex nature of acoustic comms
- strong signal attenuation
- variable multipath in shallow water environment, in movement

Introduction

MAC protocols for wireless radio

most of them assume:

- propagation delay negligible
- frequency range is wide (a few Ghz)

this assumtions cannot be applied in acoustic communication

Introduction

Positioning approaches

- range based: long baseline
- angles + range: ultra-short and short baseline
- cooperative localisation

Introduction

EvoLogics acoustic modem

- transducer with the transmit/receive amplifier
- a digital stack (DSP, FPGA, ARM processors)
- optional USBL antenna

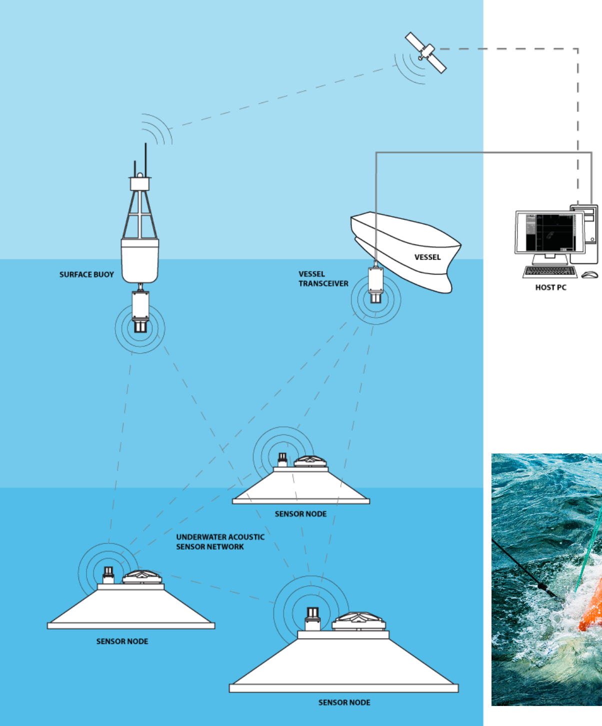
S2C-Phy physical layer protocol

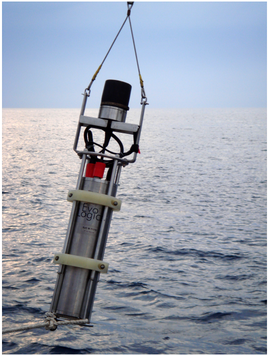
- modulation / packet detection / demodulation
- estimation of the underwater acoustic channel parameters
- positioning: time differences of arrival on the elements of the USBL grid

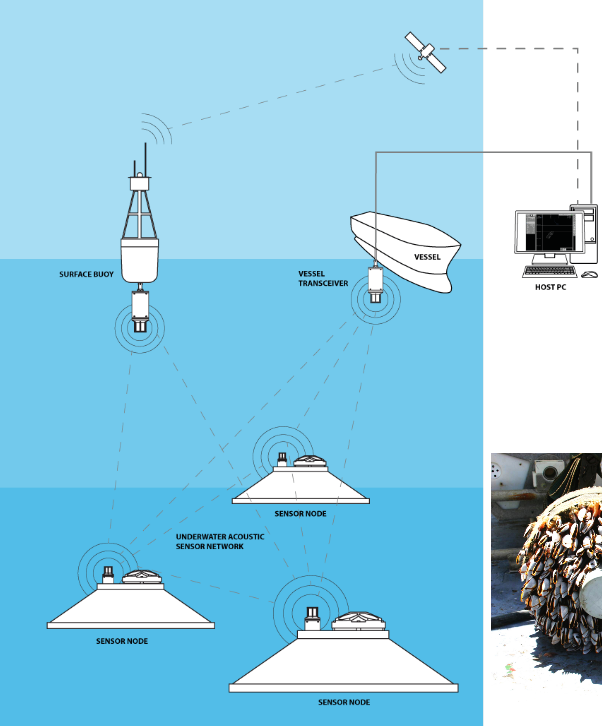
D-MAC data-link layer protocol

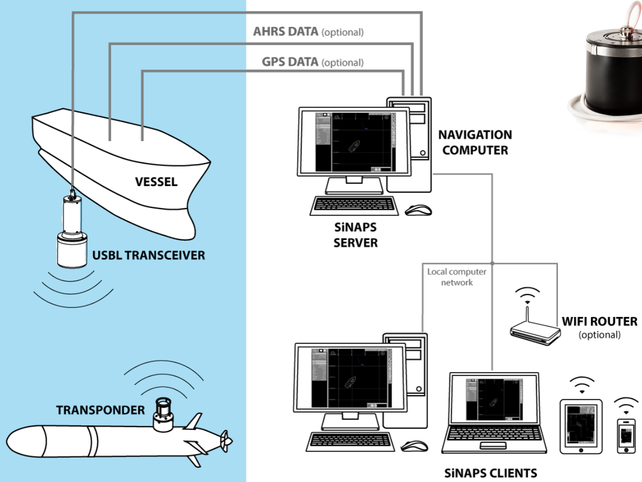
- burst media access algorithm
- short-term media access algorithm

Introduction
Applications
Motivation
Architecture
Example
Summary

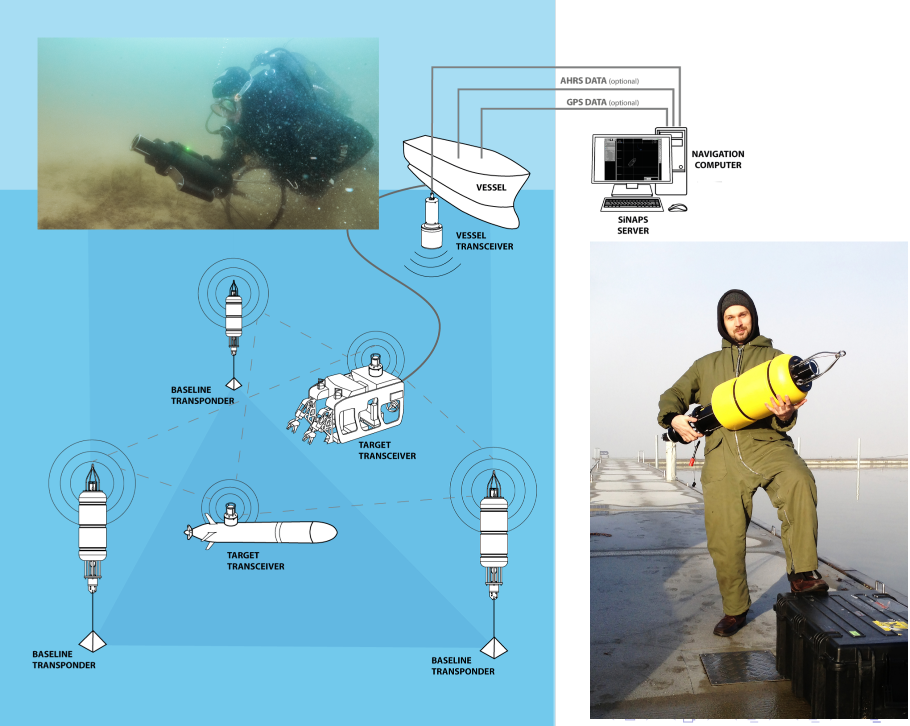

SURFACE BUOY
VESSEL
VESSEL TRANSCEIVER
HOST PC
SENSOR NODE
UNDERWATER ACOUSTIC SENSOR NETWORK
SENSOR NODE
SENSOR NODE




SURFACE BUOY
VESSEL
VESSEL
TRANSCEIVER
HOST PC
SENSOR NODE
UNDERWATER ACOUSTIC
SENSOR NETWORK
SENSOR NODE
SENSOR NODE

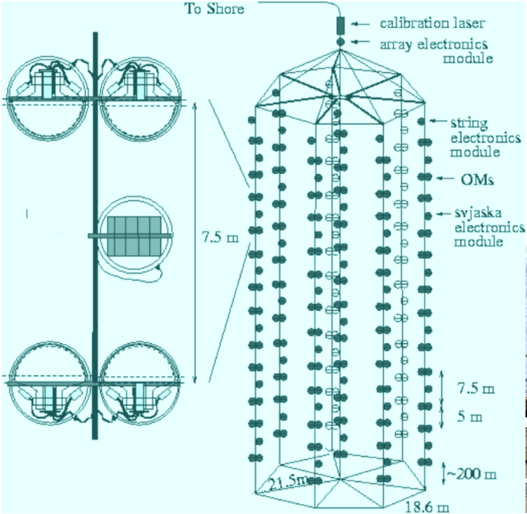

AHRS DATA (optional)
GPS DATA (optional)
NAVIGATION COMPUTER
VESSEL
SiNAPS SERVER
USBL TRANSCEIVER
Local computer network
WIFI ROUTER (optional)
TRANSPONDER
SiNAPS CLIENTS


EvoLogics


AHRS DATA (optional)
GPS DATA (optional)
NAVIGATION COMPUTER
VESSEL
SiNAPS SERVER
VESSEL TRANSCEIVER
BASELINE TRANSPONDER
TARGET TRANSCEIVER
TARGET TRANSCEIVER
BASELINE TRANSPONDER
BASELINE TRANSPONDER

Baikal Neutrino Telescope

1360 m depth in the Baikal lake, Russia



UASN frameworks

- UANT[1]
- SUNSET[2]
- DESERT[3]
- UNetStack[4]

---

[1] Torrese, D *et al* “Software-defined Underwater Acoustic Networking Platform”, WUWNet’09

[2] Petrioli, C *et al* “The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks”, Ad Hoc Networks

[3] Masiero, R. *et al* “DESERT Underwater: An NS-Miracle-based framework to design, simulate, emulate and realize test-beds for underwater network protocols”, Oceans’12, Yeosu

[4] Chitre, M. *et al* “UnetStack: An agent-based software stack and simulator for underwater networks”, Oceans’14, St. John’s

EviNS framework

based on Erlang language

why?

Challenges

- system testing under realistic conditions frequently cannot be afforded
- practical use of the systems is very expensive: high deployment, recovery and maintenance costs
- limited channel bandwidth: remote diagnostics and update nearly impossible
- no common standards for interaction between sensors

Introduction
Applications
Motivation
Architecture
Example
Summary

Challenges

In the end...

developer has to solve

- a very specific task
- during a limited period of time
- in the beginning the hardware is not yet available
- in the end not enough time to test the system

Realistically...

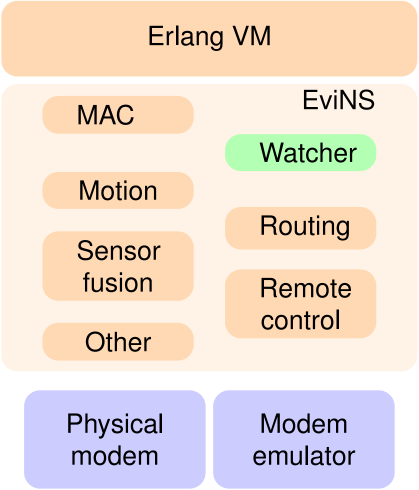
developers are usually aware about existence of errors in their software

Why Erlang?

UASN related features

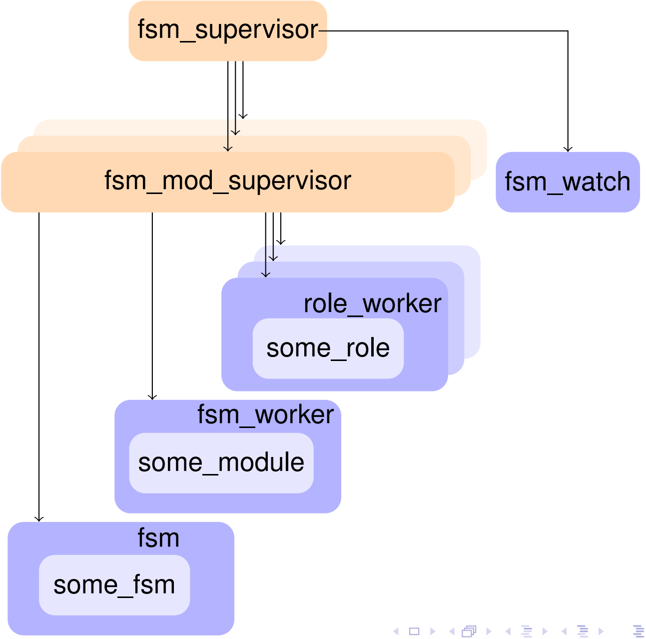
- Lightweight concurrency
- Process isolation
- Fault detection primitives
- Soft real-time
- Hardware interaction
- Large software systems
- Complex functionality
- Continuous operation

EviNS framework architecture



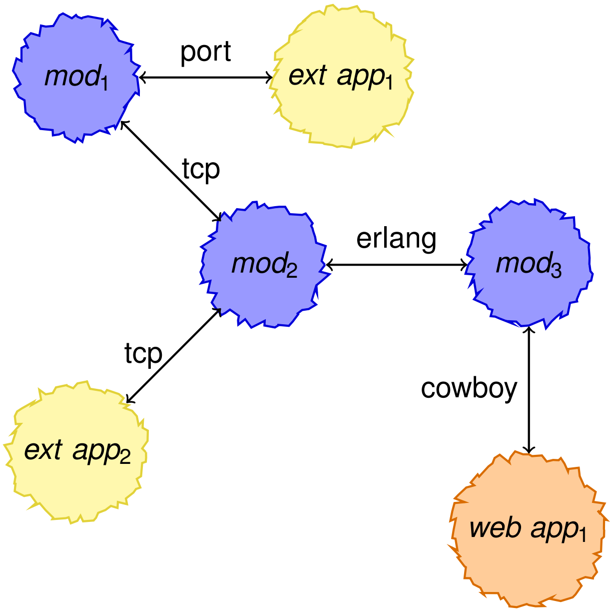
Introduction
Applications
Motivation
Architecture
Example
Summary

EviNS supervision tree



Introduction
Applications
Motivation
Architecture
Example
Summary

EviNS inter-process communiction

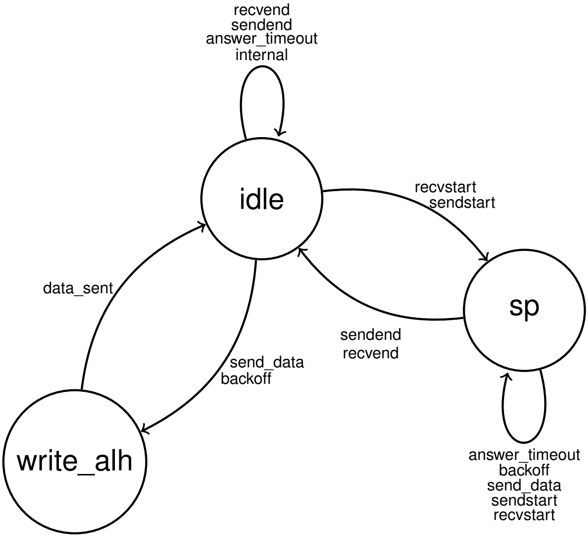


EviNS module

Module implementation steps

- define fsm_worker and fsm behaviours
- explicitly define finite state machine or push down automata
- event preprocessor: external and internal events + timers
- create module configuration

CSMA-Aloha FSM



Introduction
Applications
Motivation
Architecture
Example
Summary

CSMA-Aloha FSM: implementation

```
-define(TRANS, [
    {idle, [
        {internal,       idle},
        {answer_timeout, idle},
        {send_data,      write_alh},
        {backoff,        write_alh},
        {sendend,        idle},
        {recvend,        idle},
        {sendstart,      sp},
        {recvstart,      sp}
    ]},
    {write_alh, [
        {data_sent,      idle}
    ]},
    {sp, [
        {answer_timeout, sp},
        {backoff,        sp},
        {send_data,      sp},
        {sendstart,      sp},
        {recvstart,      sp},
        {sendend,        idle},
        {recvend,        idle}
    ]},
    {alarm,[]}
]).
```

CSMA-Aloha FSM: configuration

```
{module, a1, [
      {role,at,iface,
          {socket,"192.168.6.1",9200,client}},
      {role,triv_alh,iface,
          {socket,"0.0.0.0",9210,server}},
      {mfa, mod_triv_alh, run, []}]}.
```

CSMA-Aloha FSM: event preprocessor

```
handle_event(MM, SM, Term) ->
  Got_sync = readETS(SM, got_sync),
  case Term of
    {timeout, Event} ->
      fsm:run_event(MM, SM#sm{event=Event}, {});
    {connected} ->
      SM;
    T={send_data, _P} when Got_sync ->
      fsm:run_event(MM, SM#sm{event=send_data}, T);
    {send_data, _P} ->
      fsm:cast(SM, triv_alh, {send, {string, "ERROR SYNC\n"} });
    {async, _PID, {recvim,_,_,_,_,_,_,_,_, Payl}} ->
      fsm:cast(SM, triv_alh, {send, {binary, list_to_binary([Payl,"\n"])} });
    {async, Tuple} ->
      case Tuple of
        {sendstart,_,_,_,_} ->
          fsm:run_event(MM, SM#sm{event=sendstart},{});
        {sendend,_,_,_,_} ->
          fsm:run_event(MM, SM#sm{event=sendend},{});
        {recvstart} ->
          fsm:run_event(MM, SM#sm{event=recvstart},{});
        {recvend,_,_,_,_} ->
          fsm:run_event(MM, SM#sm{event=recvend},{});
        _ ->
          SM
      end;
    {sync, _Req, _Asw} ->
      insertETS(SM, got_sync, true),
      SM;
    UUg ->
      ?ERROR(?ID, "~s: unhandled event:~p~n", [?MODULE, UUg]),
      SM
  end.
```

CSMA-Aloha FSM: state handler example

```
handle_sp(_, #sm{event = send_data} = SM, {send_data, P}) ->
    insertETS(SM, current_msg, P),
    case fsm:check_timeout(SM, backoff) of
        false ->
            Backoff_tmp = change_backoff(SM, increment),
            fsm:set_timeout(SM#sm{event=eps}, {s, Backoff_tmp}, backoff);
        true  ->
            fsm:cast(SM, triv_alh, {send, {string, "OK\n"} }),
            SM#sm{event = eps}
    end;
handle_sp(_, #sm{event = backoff} = SM, _) ->
    Backoff_tmp = change_backoff(SM, increment),
    fsm:set_timeout(SM#sm{event=eps}, {s, Backoff_tmp}, backoff);
handle_sp(_, SM, _) ->
    SM#sm{event = eps}.
```

Summary

Final notes

- project page:
  https://github.com/okebkal/evins
- “release early, release often”
- acoustic modem emulator per request
- any integration efforts with other acoustic modems are welcome

Future work

- modem interface generalization to support other acoustic modems
- networking protocol interface unification

Introduction
Applications
Motivation
Architecture
Example
Summary

Thanks for a Patient Hearing

Questions?

Introduction
Applications
Motivation
Architecture
Example
Summary