auto**N**omous, self-Learning, **OPTI**mal and comp**L**ete **U**nderwater **S**ystems **NOPTILUS**

FP7-ICT-2009.6: Information and Communication Technologies

2nd Project Review

WP6 (Situation Understanding) Progress Report

E. Orfanoudakis, N. Kofinas, and M. G. Lagoudakis
Telecommunication Systems Institute (TSI), Greece

November 28, 2013
FEUP, Portugal

NOPTILUS

SEVENTH FRAMEWORK PROGRAMME

WP6: Situation Understanding

Event Recognition

- *observations*: abstract sensor readings, estimated state
- *events*: patterns in observation sequences

Models

- Probabilistic Context-Free Grammars (PCFGs)

Task 6.1: PCFG Parsing [Orfanoudakis]

- real-time, hierarchical parsing for on-line recognition

Task 6.2: PCFG Learning [Kofinas]

- off-line learning of PCFGs from past AUV mission logs

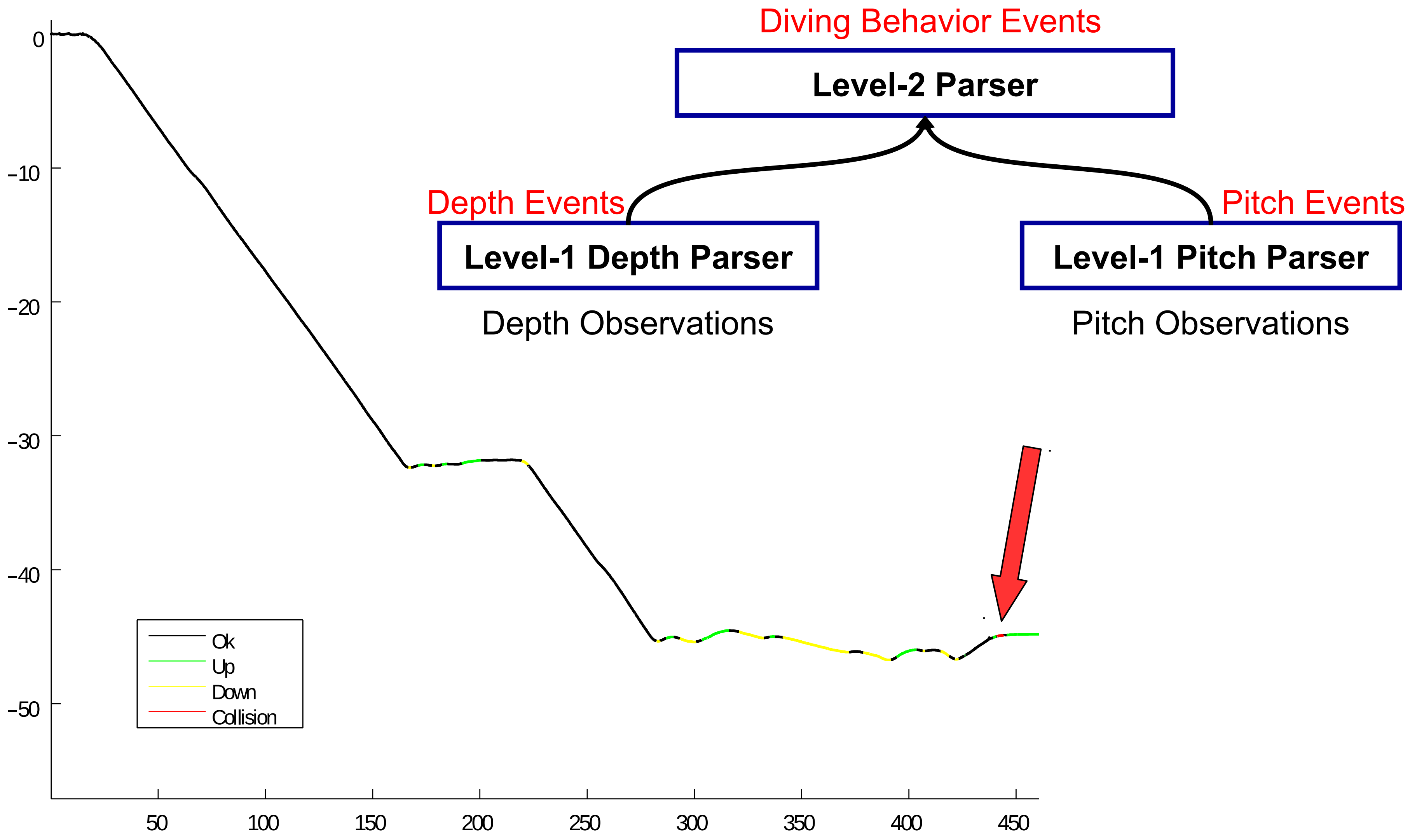
Task 6.3: Integration

- on-line event recognition using learned PCFGs

WP6 Summary from 1st Review

- **Preprocessing**
  - abstraction from raw data to observations
- **Event recognition**
  - *events*: diving behavior
  - *technology*: probabilistic context-free grammars
  - successful recognition on several past mission logs
  - demonstration of feasibility and potential
- **Software**
  - optimized parser for event recognition in Matlab

Diving Behavior Event Recognition



Diving Behavior Events

Level-2 Parser

Depth Events

Pitch Events

Level-1 Depth Parser

Level-1 Pitch Parser

Depth Observations

Pitch Observations

WP6 Plans from 1st Review

- **Software** 2 Done!
  - optimized parser for real-time event recognition in Dune
- **Event recognition** rethink!
  - *new events*: getting caught in net, vehicle faults, …
- **Rethinking events**
  - normal vs. abnormal AUV operation 1 Done!
  - *idea*: instead of looking for the abnormal and rare, …
  - … why not look after the normal and frequent?
- **Structured prediction**
  - automatic construction of grammars 3 ongoing!
  - learning rules and probabilities from past missions

WP6: Event Recognition

Rethinking Events

External Events

- **Intention**
  - recognize events which are “external” to the AUV
- **Observations**
  - salinity sensor measurements
  - temperature sensor measurements
  - other external (processed) readings from other WPs
- **Events**
  - crossing borders of areas with different salinity
  - unusual variations in temperature
- **Importance**
  - salinity affects ranging measurements
  - temperature may relate to the presence of a current

Normal/Abnormal Events

Considerations

- what is an interesting event? what do we care about?
- likely interesting events are unusual and unexpected
- in most missions almost nothing abnormal occurs
- *idea*: instead of looking for the abnormal and rare, …
- … why not look after the normal and frequent?
- easier to define normal as opposed to abnormal

Normal Operation

- typical patterns in motion and measurements (PCFG!)

Abnormal Operation

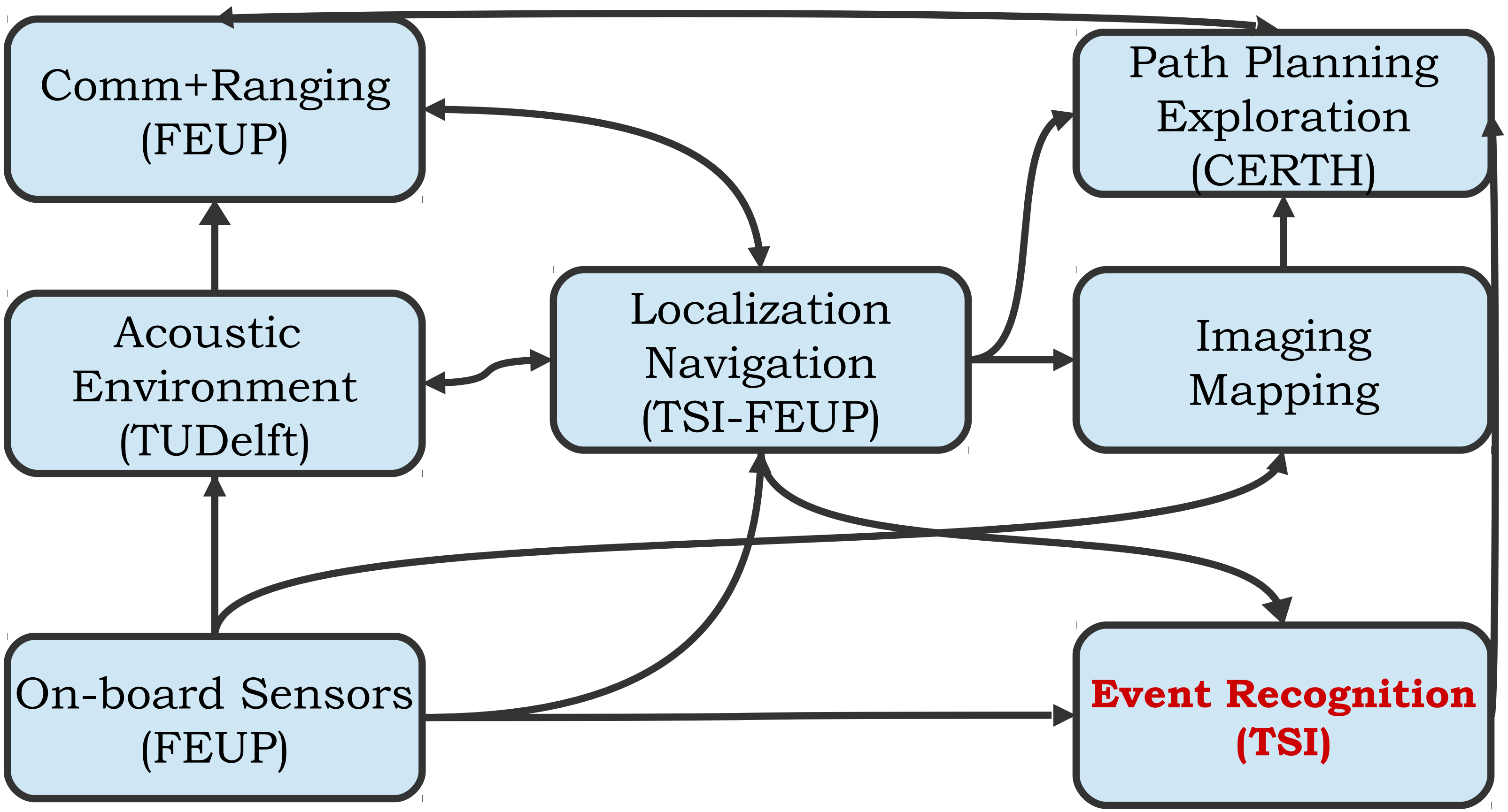
- any pattern that does not occur in normal operation

WP6 Task 6.1: PCFG Parsing

Optimized parser implementation in Dune

Noptilus Dune Code Architecture

NOPTILUS



Comm+Ranging (FEUP)

Path Planning Exploration (CERTH)

Acoustic Environment (TUDelft)

Localization Navigation (TSI-FEUP)

Imaging Mapping

On-board Sensors (FEUP)

Event Recognition (TSI)

TSI Dune Code Repository

- **Github**
  - **`https://github.com/vosk/dune/`**
  - fork of the main LSTS-Dune repository
  - C++ software architecture (and simulator) for AUVs
  - eventually, will be integrated into the main repository
- **Development (ongoing)**
  - WP3: Cooperative Localization
    - partial and fully cooperative localization schemes
    - simulation of acoustic range measurements
  - WP6: Event Recognition
    - real-time CYK parser engine for any given PCFG
    - messaging for communicating observations and events

Implementation Challenges

Complexity

- thousands of lines of code
- additions of new features
- debugging new functionality
- compatibility with updates/upgrades

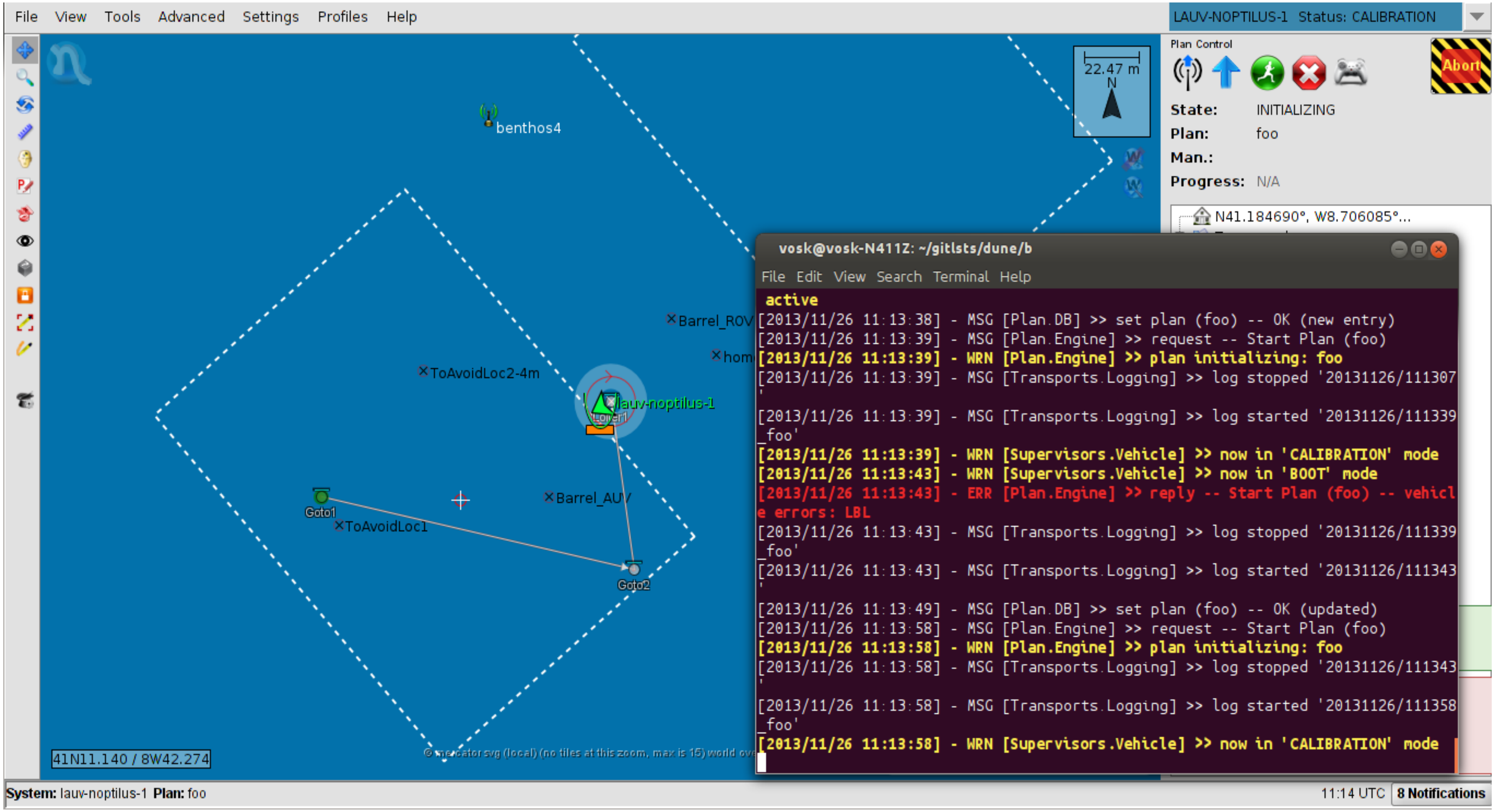
Optimization

- execution on a limited embedded platform
- real-time synchronization of observations
- hard to assess real-time performance

Simulation

- accuracy of acoustic model and measurements
- accuracy of acoustic environment model

Dune+Neptus Environment



File View Tools Advanced Settings Profiles Help
LAUV-NOPTILUS-1 Status: CALIBRATION
Plan Control
Abort
22.47 m
N
benthos4
Barrel_ROV
hom
ToAvoidLoc2-4m
lauv-noptilus-1
Loiter1
Barrel_AUV
Goto1
ToAvoidLoc1
Goto2
41N11.140 / 8W42.274
N41.184690°, W8.706085°...
vosk@vosk-N411Z: ~/gitlsts/dune/b
File Edit View Search Terminal Help
11:14 UTC
8 Notifications

State: INITIALIZING
Plan: foo
Man.:
Progress: N/A

System: lauv-noptilus-1 Plan: foo

```
active
[2013/11/26 11:13:38] - MSG [Plan.DB] >> set plan (foo) -- OK (new entry)
[2013/11/26 11:13:39] - MSG [Plan.Engine] >> request -- Start Plan (foo)
[2013/11/26 11:13:39] - WRN [Plan.Engine] >> plan initializing: foo
[2013/11/26 11:13:39] - MSG [Transports.Logging] >> log stopped '20131126/111307
'

[2013/11/26 11:13:39] - MSG [Transports.Logging] >> log started '20131126/111339
_foo'
[2013/11/26 11:13:39] - WRN [Supervisors.Vehicle] >> now in 'CALIBRATION' mode
[2013/11/26 11:13:43] - WRN [Supervisors.Vehicle] >> now in 'BOOT' mode
[2013/11/26 11:13:43] - ERR [Plan.Engine] >> reply -- Start Plan (foo) -- vehicl
e errors: LBL
[2013/11/26 11:13:43] - MSG [Transports.Logging] >> log stopped '20131126/111339
_foo'
[2013/11/26 11:13:43] - MSG [Transports.Logging] >> log started '20131126/111343
'

[2013/11/26 11:13:49] - MSG [Plan.DB] >> set plan (foo) -- OK (updated)
[2013/11/26 11:13:58] - MSG [Plan.Engine] >> request -- Start Plan (foo)
[2013/11/26 11:13:58] - WRN [Plan.Engine] >> plan initializing: foo
[2013/11/26 11:13:58] - MSG [Transports.Logging] >> log stopped '20131126/111343
'

[2013/11/26 11:13:58] - MSG [Transports.Logging] >> log started '20131126/111358
_foo'
[2013/11/26 11:13:58] - WRN [Supervisors.Vehicle] >> now in 'CALIBRATION' mode
```

WP6 Task 6.2: PCFG Learning

Learning the structure of a grammar

Learning Approaches

- existing work mostly in natural language processing
- no single algorithm can learn any arbitrary grammar!

➢Using annotated examples

- annotated strings and/or syntactic trees are required
- *problem*: infeasible to annotate lots of data!

➢Using positive and negative examples

- near-miss negative data are required
- *problem*: near-misses can hardly be defined!

➢Using positive examples

- algorithms specialized for certain grammar classes
- typically assume few symbols and few examples

Our Approach

- **Ongoing investigation**
  - review of existing approaches and their assumptions
  - implementation of promising learning algorithms
  - adaptation to the characteristics of our problem
  - experiments on examples from real AUV data
  - focus on handling large sets of positive training strings
- **Current promising learning algorithm**
  - learn with basic chunk and merge operations
  - use iterative bi-clustering to decide the best chunk
  - at each iteration, check for possible merge operations
  - repeat until the training corpus is full reduced
  - rule probabilities can be extracted during the process

Grammar Learning Example (1)

	a	b	c
a	6	0	4
b	0	6	0
c	0	4	0

Corpus:

$a^n c b^n$

- acb
- aacbb
- aaacbbb
- aaaacbbbb

Rules:

- N0 → ac

	a	b	N0
a	3	0	3
b	0	6	0
N0	0	4	0

Corpus:

- N0 b
- a N0 bb
- aa N0 bbb
- aaa N0 bbbb

Rules:

- N0 → ac
- N1 → N0 b

Grammar Learning Example (2)

	a	b	N1
a	3	0	3
b	0	3	0
N1	0	3	0

Corpus:

- N1
- a N1 b
- aa N1 bb
- aaa N1 bbb

Rules:

- N0 → ac
- N1 → N0 b
- N2 → a N1

	a	b	N2
a	1	0	2
b	0	3	0
N2	0	3	0

Corpus:

- N1
- N2 b
- a N2 bb
- aa N2 bbb

Rules:

- N0 → ac
- N1 → N0 b
- N2 → a N1
- N3 → N2 b

Grammar Learning Example (3)

	a	b	N3
a	1	0	2
b	0	1	0
N3	0	2	0

Corpus:

- N1
- N3
- a N3 b
- aa N3 bb

Rules:

- N0 → ac
- N1 → N0 b
- N2 → a N1
- N3 → N2 b
- N4 → a N3

	a	b	N4
a	0	0	1
b	0	1	0
N4	0	2	0

Corpus:

- N1
- N3
- N4 b
- a N4 bb

Rules:

- N0 → ac
- N1 → N0 b
- N2 → a N1
- N3 → N2 b
- N4 → a N3
- N5 → N4 b

Grammar Learning Example (4)

	a	b	N4
a	0	0	1
b	0	1	0
N4	0	2	0

Corpus:

- N1
- N3
- N4 b
- a N4 bb

Rules:

- N0 → ac
- N1 → N0 b
- N2 → a N1
- N3 → N2 b
- N4 → a N3
- N5 → N4 b

Corpus:

- N1
- N3
- N3
- N3

Rules:

$a^n c b^n$

- S → N1
- S → N3
- N0 → ac
- N1 → N0 b
- N2 → a N1
- N2 → a N3
- N3 → N2 b

WP6 Progress and Plans

What's there? What's next?

WP6 Incremental Progress

- **Events**
  - normal vs. abnormal events in AUV operation
- **Task 6.1: PCFG Parsing**
  - implementation/development on cloned Dune repository
  - optimized parser for real-time event recognition in Dune
  - required messaging modifications in IMC
- **Task 6.2: PCFG Learning**
  - learning (structured prediction) of grammars
  - learning using positive data only from past missions

WP6 Plans

- **Task 6.1: Software**
  - integration of parser in the main Dune repository
  - on-board testing and debugging
- **Task 6.2: Structured prediction**
  - efficient implementation of grammar learning algorithm
  - learning experiments on data from past missions
  - testing on similar mission logs
  - testing on synthetic abnormalities
- **Events**
  - identification of normal/abnormal external events
  - focus on specific measurements for parsing
  - (hierarchical) grammar learning on selected inputs

WP6 Integration/Contribution

- **Off-line (before)**
  - identify type of event
  - identify related data
  - collect normal data
  - learn grammar(s)
- **On-line (during)**
  - execute parser onboard
  - recognize abnormal
  - signal detection(s)
- **Off-line (after)**
  - parse past mission logs
  - feature detection

- **Event**
  - propeller free-winding
  - current, control, gyro
  - normal mission data
  - rules and probabilities
- **Mission**
  - use learned grammar
  - detect abnormality
  - reset controller
- **Investigation**
  - detect past occurrences
  - extract event statistics

Thank you!