Prosody and the Interface Metaphor:

Operational Models

Dafydd Gibbon

Universität Bielefeld

International Seminar on Prosodic Interfaces
Jawaharlal Nehru University, New Delhi
November 2011

Leonhard Euler (1707-1783)

"Ohne Zweifel wäre das eine von den wichtigsten Entdeckungen, wenn man eine Maschine erfünde, die alle Töne unserer Wörter mit allen ihren Artikulationen aussprechen könnte. Die Sache scheint mir nicht unmöglich zu seyn."

"It would be a considerable invention indeed, that of a machine able to mimic our speech with its sounds and articulations. I think it is not impossible."

So?

The point is:

We already know a huge amount about the "tones" and the "sounds and articulations" (though a lot is still to be learned).

But something is missing:

We do not really know how they work, i.e. 'operate'.

Thesis:

To know how "tones" and "sounds and articulations" work, we need to design and build working, operational models. in addition to our rules and representations.

Operational models do not guarantee truth but they add criteria of consistency, soundness, and completeness to our theories.

So?

The point is:

We already know a huge amount about the “tones” and the “sounds and articulations” (though a lot is still to be learned).

But something is missing:

We do not really know how they work, i.e. ‘operate’.

Thesis:

To know how “tones” and “sounds and articulations” work, we need to design and build working, operational models. in addition to our rules and representations.

Operational models do not guarantee truth but they add criteria of consistency, soundness, and completeness to our theories.

So?

The point is:

We already know a huge amount about the “tones” and the “sounds and articulations” (though a lot is still to be learned).

But something is missing:

We do not really know how they work, i.e. ‘operate’.

Thesis:

To know how “tones” and “sounds and articulations” work, we need to design and build working, operational models. in addition to our constraints, rules and representations.

Operational models do not guarantee truth but they add criteria of consistency, soundness, and completeness to our theories.

Static vs. operational models

There are many – relatively distantly - related distinctions:

- Structural models with types of generalisation:
  - representations + (*declarative*) constraint / (*derivational*) rule (cf. OT/GP)
    - in each case with procedural application conventions
- Domain:
  - competence – performance
  - theory – practice

None of those – but rather, two epistemological layers:

- procedural: data structure – algorithm (*space-time properties: static*)
- operational: data type – program (*space-time properties: dynamic*)

Somewhat like the distinction

- decorative model / illustrative model / architect's model
  - a model aeroplane designed to be put on a shelf (*time & space: static*)
- working model
  - a model aeroplane which is designed to fly (*time & space: dynamic*)

Distantly related distinctions

There are many – relatively distantly - related distinctions:

- Structural models with types of generalisation:
  - representations + (*declarative*) constraint / (*derivational*) rule (cf. OT/GP)
    - in each case with procedural application conventions
- Domain:
  - competence – performance
  - theory – practice

None of those – but rather, two epistemological layers:

- procedural: data structure – algorithm (*space-time properties: static)*
- operational: data type – program (*space-time properties: dynamic*)

Somewhat like the distinction

- decorative model / illustrative model / architect's model
  - a model aeroplane designed to be put on a shelf (*time & space: static*)
- working model
  - a model aeroplane which is designed to fly (*time & space: dynamic*)

Distantly related distinctions

There are many – relatively distantly - related distinctions:

- Structural models with types of generalisation:
  - representations + (*declarative*) constraint / (*derivational*) rule (cf. OT/GP)
    - in each case with procedural application conventions
- Domain:
  - competence – performance
  - theory – practice

None of those – but rather, two epistemological layers:

- procedural: data structure – algorithm (*space-time properties: static*)
- operational: data type – program (*space-time properties: dynamic*)

Somewhat like the distinction

- decorative -/ illustrative model (cf. anatomy):
  - a model aeroplane designed to be put on a shelf (*time & space: static*)
- working model (cf. physiology):
  - a model aeroplane which is designed to fly (*time & space: dynamic*)

THEORIES

are interpreted by

MODELS

static models (*timeless*) – operational models (*timed*)

which represent

REALITY

Theories – models - reality

A <u>theory</u>

with a set of <u>premises</u> and <u>derivations</u>/<u>constraints</u>

needs interpretation by a <u>model</u> of <u>reality</u>

with a <u>structure</u> representing a simplified version of reality

A semiotic theory needs interpretation by <u>two</u> models:

- a <u>semantic</u> model (some variety of denotational semantics)
- a <u>media</u> model (some variety of phonetics, writing, gesture)

Each model may be

- an implicit model
  - e.g. our intuitive judgments about speech)
- an explicit model:
  - static: phonetic representations, etc.
  - operational: working model of data structures, algorithms

So: what is an operational model?

A theory

with a set of premises and derivations/constraints

needs interpretation by a model of reality

with a structure representing a simplified version of reality

A semiotic theory needs interpretation by two models:

- a semantic model (some variety of denotational semantics)
- a media model (some variety of phonetics, writing, gesture)

Each model may be

- an implicit model
  - e.g. our intuitive judgments about speech)
- an explicit model:
  - static: phonetic representations, etc.
  - operational: working model of data structures, algorithms

So: what is an operational model?

A theory

with a set of premises and derivations/constraints

needs interpretation by a model of reality

with a structure representing a simplified version of reality

A semiotic theory needs interpretation by two models:

- a semantic model (some variety of denotational semantics)
- a media model (some variety of phonetics, writing, gesture)

Each model may be

- an implicit model
  - e.g. our intuitive judgments about speech)
- an explicit model:
  - static: rules/constraints yielding phonetic representations, etc.
  - operational: real-time working model of data structures, algorithms

So: what is an operational model?

A theory

with a set of premises and derivations/constraints

needs interpretation by a model of reality

with structure representing a simplified version of reality

A semiotic theory needs interpretation by two models:

- a semantic model (some variety of denotational semantics)
- a media model (some variety of phonetics, writing, gesture)

Each model may be

- an implicit model
  - e.g. our intuitive judgments about speech)
- an explicit model:
  - static: rules/constraints yielding phonetic representations, etc.
  - operational: real-time working model of data structures, algorithms

Time

Static approaches

Obviously...

- Mainstream prosody models have much to say about
  - phonological and prosodic <u>forms and structures</u>
    - segments, tones, harmonies, ...
    - <u>relations</u> (interfaces?) between differently motivated structures
    - i.e. data structures
  - <u>mappings</u> (interfaces?) between forms
    - <u>derivations</u> (GP), <u>constraint sequences</u> (OT)
    - i.e. algorithms

But, oddly enough...

- Mainstream phonological prosody models have little to say about
  - real time (and real space, but that's another issue...)
    - real durations, real rhythms, real pauses, ...
    - real procedures, real processes, ...
- A basic strategy seems to be:
  - Leave them to phonetics or put them in a "performance" trash bin...

Static approaches

Obviously...

- Mainstream prosody models have much to say about
  - phonological and prosodic <u>forms and structures</u>
    - segments, tones, harmonies, ...
    - <u>relations</u> (interfaces?) between differently motivated structures
    - i.e. data structures
  - <u>mappings</u> (interfaces?) between forms
    - <u>derivations</u> (GP), <u>constraint sequences</u> (OT)
    - i.e. algorithms

But, oddly enough...

- Mainstream phonological prosody models have little to say about
  - real time (and real space, but that's another issue...)
    - real durations, real rhythms, real pauses, ...
    - real procedures, real processes, ...
- A basic strategy seems to be:
  - Leave them to phonetics or put them in a "performance" trash bin...

Operational approaches

The machine - anatomical models

- Fujimura *passim*: mass+spring theory of the articulators
- Ohala 1994: Frequency code
  - fundamental determinants: static properties of larynx size
    - length of vocal cords
    - (also: thickness, tension of vocal cords, ...)

The machine's workings: operational models

- Time models:
  - Bird & Klein 1989: event structure (Event Phonology)
    - event = < *property* , *interval* >
  - Gibbon 1991, Berndsen 1998: Time Types (categorial, relational, absolute/real)
  - Fujisaki *passim*: declining logarithmic trajectories throughout behaviours
    - cf. Gussenhoven 2002: Production code
- Control models:
  - Lindblom *passim*: hypospeech/hyperspeech (production/perception orientation)
    - economy of speech vs. clarity of speech
    - cf. Gussenhoven 2002: Effort code
  - Levelt *passim*: error correction model
    - self-monitoring and repair (cf. studies in psycholinguistics)
    - other-monitoring and correction (cf. studies in conversation analysis)

Operational approaches

The <u>machine</u> - anatomical models

- Fujimura *passim*: mass+spring theory of the articulators
- Ohala 1994: Frequency code
  - fundamental determinants: static properties of larynx size
    - length of vocal cords
    - (also: thickness, tension of vocal cords, ...)

The machine's <u>workings</u>: operational models

- <u>Time models</u>:
  - Bird & Klein 1989: event structure (Event Phonology)
    - event = < *property* , *interval* >
  - Gibbon 1991, Berndsen 1998: Time Types (categorial, relational, absolute/real)
  - Fujisaki *passim*: declining logarithmic trajectories throughout behaviours
    - cf. Gussenhoven 2002: Production code
- <u>Control models</u>:
  - Lindblom *passim*: hypospeech/hyperspeech (production/perception orientation)
    - <u>economy of speech</u> vs. <u>clarity of speech</u>
    - cf. Gussenhoven 2002: Effort code
  - Levelt *passim*: error correction model
    - self-monitoring and repair (cf. studies in psycholinguistics)
    - other-monitoring and correction (cf. studies in conversation analysis)

Operational approaches

- The machine - anatomical models
  - Fujimura *passim*: mass+spring theory of the articulators
  - Ohala 1992: Frequency code
    - fundamental determinants: static properties of larynx size
      - length of vocal cords
      - (also: thickness, tension of vocal cords, ...)
- The machine's workings: operational models
  - Time models:
    - Bird & Klein 1989: event structure (Event Phonology)
      - event = < *property* , *interval* >
    - Gibbon 1991, Berndsen 1998: Time Types (categorial, relational, absolute/real)
    - Fujisaki *passim*: declining logarithmic trajectories throughout behaviours
      - cf. Gussenhoven 2002: Production code
  - Control models:
    - Lindblom *passim*: hypospeech/hyperspeech (production/perception orientation)
      - economy of speech vs. clarity of speech
      - cf. Gussenhoven 2002: Effort code
    - Levelt *passim*: error correction model
      - self-monitoring and repair (cf. studies in psycholinguistics)
      - other-monitoring and correction (cf. studies in conversation analysis)

Operational approaches

The machine - anatomical models

- Fujimura *passim*: mass+spring theory of the articulators
- Ohala 1994 ‘Frequency code’
    - fundamental determinants: static properties of larynx size
        - length of vocal cords
        - (also: thickness, tension of vocal cords, ...)

The machine’s workings: operational models

- Time models:
    - Bird & Klein 1989: event structure (Event Phonology)
        - event = < *property* , *interval* >
    - Gibbon 1991, Berndsen 1998: Time Types (categorial, relational, absolute/real)
    - Fujisaki *passim*: declining logarithmic trajectories throughout behaviours
        - cf. Gussenhoven 2002 ‘Production code’
- Control models:
    - Lindblom *passim*: hypospeech/hyperspeech (production/perception orientation)
        - economy of speech vs. clarity of speech
        - cf. Gussenhoven 2002: ‘Effort code’
    - Levelt’s error correction model
        - self-monitoring and repair (cf. studies in psycholinguistics)
        - other-monitoring and correction (cf. studies in conversation analysis)

Time models and prosody

Time models:

Essential components of linguistic descriptions:

- not only in semantics (considered obvious)
  - cf. models of tense, aspect, ‘Aktionsart’
- not only in phonetics (also considered obvious)
  - cf. models of VOT, rhythm alternation, tempo
- not only in psycholinguistics (also considered obvious)
  - cf. models of production and reception timing
- not only in computational linguistics (also considered obvious)
  - cf. models of sequential and parallel processing

But also for realistic models of both phonology and prosody:

- notions of inferential ordering
  - rules, constraints – but also how they interact in time
- notions of durations as *time types*
  - *categorial time*, *relational time*, *absolute/real time*

Time models and prosody

Time models:

Essential components of linguistic descriptions:

- not only in semantics (considered obvious)
  - cf. models of tense, aspect, ‘Aktionsart’
- not only in phonetics (also considered obvious)
  - cf. models of VOT, rhythm alternation, tempo
- not only in psycholinguistics (also considered obvious)
  - cf. models of production and reception timing
- not only in computational linguistics (also considered obvious)
  - cf. models of sequential and parallel processing

But also for realistic models of phonology, prosody:

- notions of inferential ordering
  - rules, constraints – but also how they interact <u>in time</u>
- notions of durations as *time types*
  - *<u>categorial</u> time*, *<u>relational</u> time*, *<u>absolute/real</u> time*

Interfaces

Interfaces: metaphor and reality

- Linguistic metaphor:
  - relation/mapping between representations at different levels
    - phonology-phonetics (interpretation)
    - tone-morphology (composition)
    - intonation-semantics/pragmatics (interpretation)

      ... *i.e. basically any pair of levels*
- Computer science entity:

  ... an interface is a tool and concept that refers to a point of interaction between components, and is applicable at the level of both hardware (device interface) and software (parameter interface).

  This allows a component, whether a piece of hardware such as a graphics card or a piece of software such as an Internet browser, to function independently while using interfaces to communicate with other components via an input/output system and an associated protocol.

  ... a computing interface may refer to the means of communication between the computer and the user by means of peripheral devices such as a monitor or a keyboard, an interface with the Internet via Internet Protocol, and any other point of communication involving a computer.

  Wikipedia: Interface (computing)

Interfaces: metaphor and reality

- Linguistic metaphor:
  - relation/mapping between representations at different levels
    - phonology-phonetics (interpretation)
    - tone-morphology (composition)
    - intonation-semantics/pragmatics (interpretation)

      ... *i.e. basically any pair of levels*
- Computer science entity:

  ... an interface is a tool and concept that refers to a point of interaction between components, and is applicable at the level of both hardware (<u>device interface</u>) and software (<u>parameter interface</u>).

  > This allows a component, whether a piece of hardware such as a graphics card or a piece of software such as an Internet browser, to <u>function independently</u> while using interfaces to communicate with other components via an input/output system and an associated protocol.

  ... a computing interface may refer to the means of communication between the computer and the user by means of peripheral devices such as a monitor or a keyboard, an interface with the Internet via Internet Protocol, and any other point of communication involving a computer.

  Wikipedia: Interface (computing)

Interfaces: metaphor and reality

- Linguistic metaphor:
  - relation/mapping between representations at different levels
    - phonology-phonetics (interpretation)
    - tone-morphology (composition)
    - intonation-semantics/pragmatics (interpretation)

      ... *i.e. basically any pair of levels*
- Computer science entity:

  ... an interface is a tool and concept that refers to a point of interaction between components, and is applicable at the level of both hardware (device interface) and software (parameter interface).

  This allows a component, whether a piece of hardware such as a graphics card or a piece of software such as an Internet browser, to function independently while using interfaces to communicate with other components via an input/output system and an associated protocol.

  ... a computing interface may refer to the means of communication between the computer and the user by means of peripheral devices such as a monitor or a keyboard, an interface with the Internet via Internet Protocol, and any other point of communication involving a computer.

  Wikipedia: Interface (computing)

Many ‘interfaces’?

So we know

- that interfaces are intended to support modularity
- that there are hardware interfaces (let’s call these ‘phonetic’)
- that there are software interfaces (let’s call these ‘linguistic’)

But:

- Are the interfaces defined ad hoc?
- If not, they presuppose prior known modules.
- So which modules are presupposed?
  - Not many...

However:

- prosody has broad functionality:
  - word level:
    - phonemic, morphemic
  - sentence level:
    - phrasal, sentential
  - utterance level:
    - dialogue acts, turn-taking
- broader than usual concept of language architecture is needed
  - the Rank-Interpretation model (RIM)

Many ‘interfaces’?

So we know

- that interfaces are intended to support modularity
- that there are hardware interfaces (let’s call these ‘phonetic’)
- that there are software interfaces (let’s call these ‘linguistic’)

But:

- Are the interfaces defined ad hoc?
- If not, they presuppose prior known modules.
- So which modules are presupposed?
  - Not many...

However:

- prosody has broad functionality:
  - word level:
    - phonemic, morphemic
  - sentence level:
    - phrasal, sentential
  - utterance level:
    - dialogue acts, turn-taking
- broader than usual concept of language architecture is needed
  - the Rank-Interpretation model (RIM)

Many ‘interfaces’?

So we know

- that interfaces are intended to support modularity
- that there are hardware interfaces (let’s call these ‘phonetic’)
- that there are software interfaces (let’s call these ‘linguistic’)

But:

- Are the interfaces defined ad hoc?
- If not, they presuppose prior known modules.
- So which modules are presupposed?
  - Not many...

However:

- prosody has broad functionality:
  - word level:
    - phonemic, morphemic
  - sentence level:
    - phrasal, sentential
  - utterance level:
    - dialogue acts, turn-taking
- broader than usual concept of language architecture is needed
  - the Rank-Interpretation model (RIM)

Many ‘interfaces’?

So we know

- that interfaces are intended to support modularity
- that there are hardware interfaces (let’s call these ‘phonetic’)
- that there are software interfaces (let’s call these ‘linguistic’)

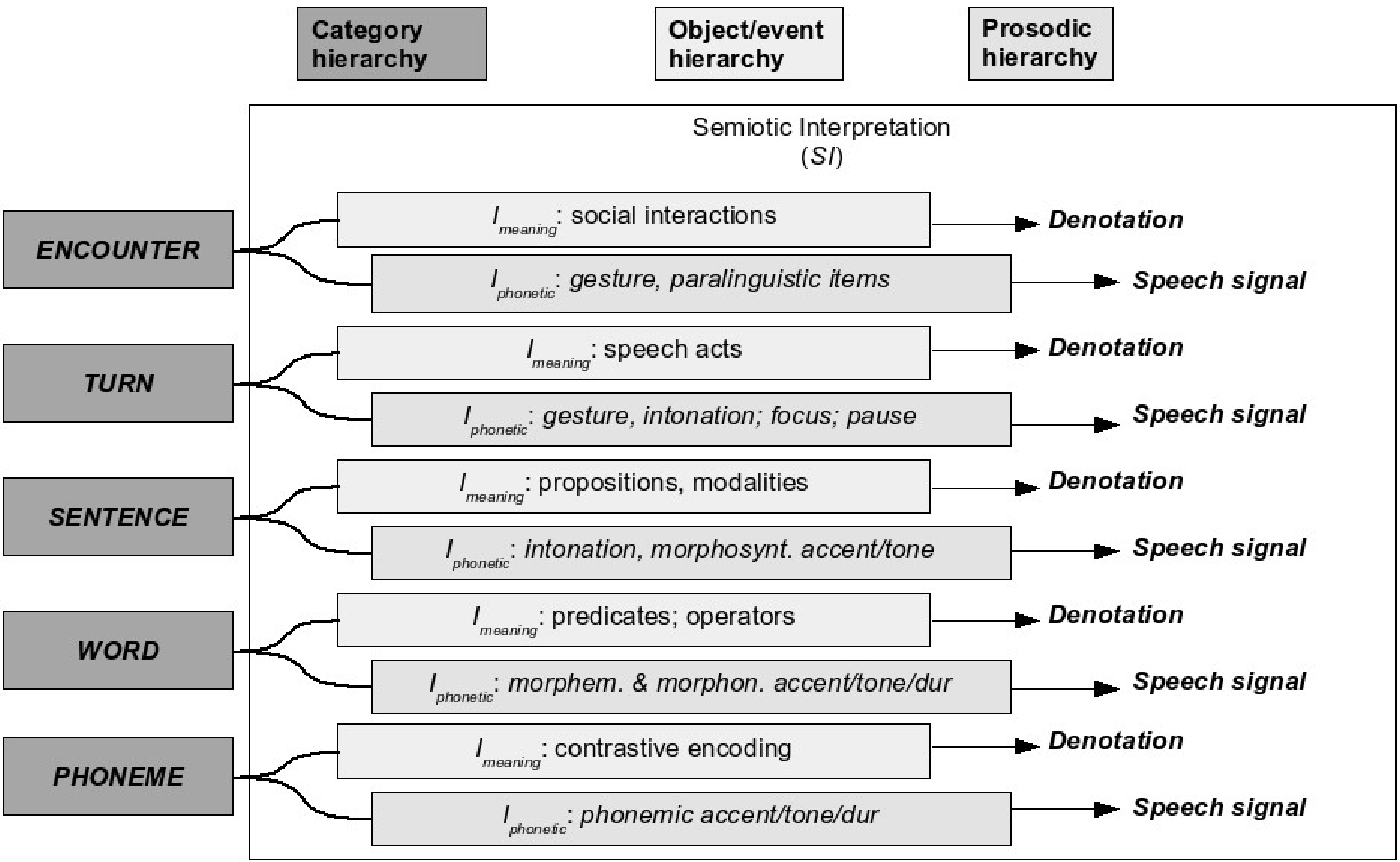
But:

- Are the interfaces defined ad hoc?
- If not, they presuppose prior known modules.
- So which modules are presupposed?
  - Not many...

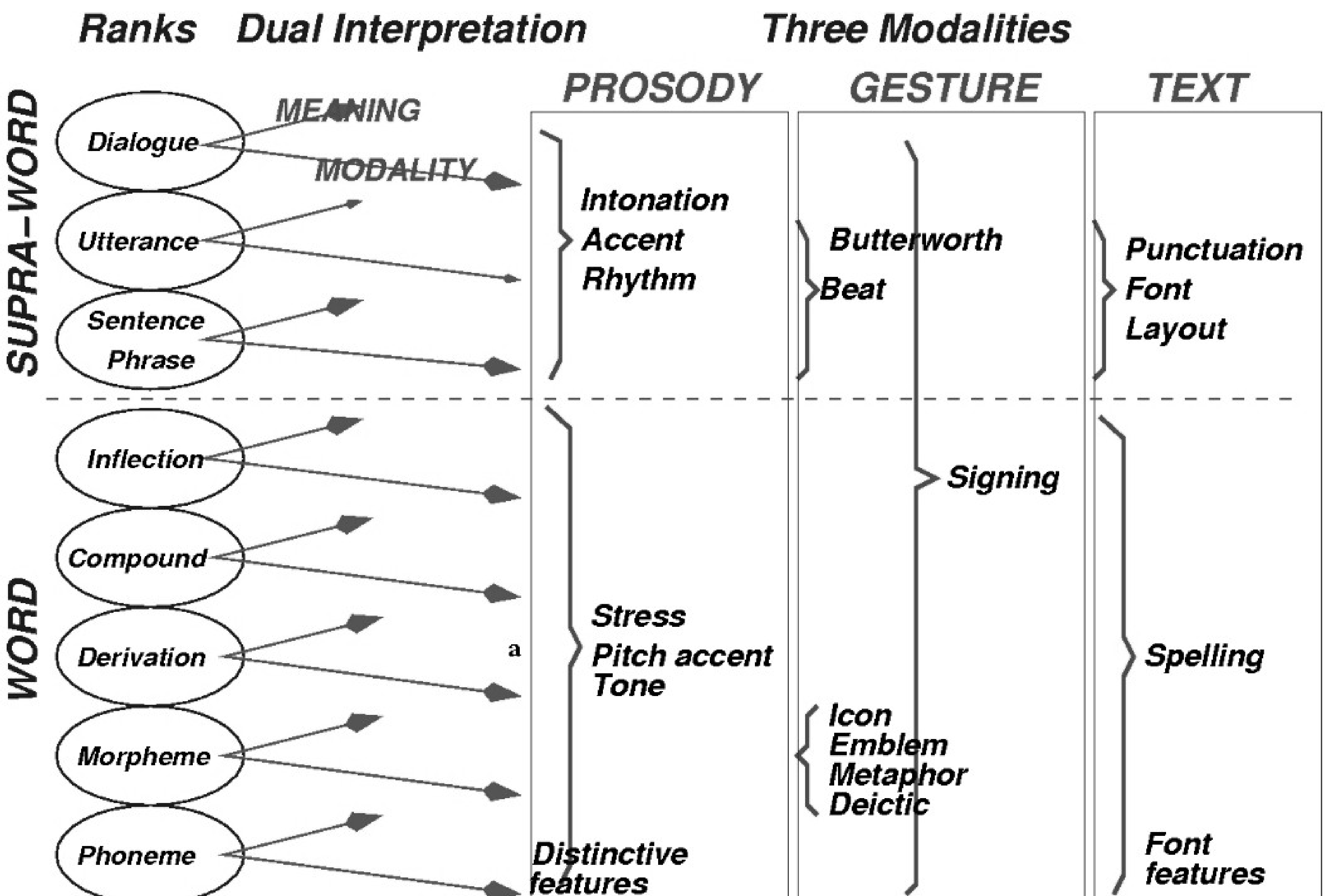
However:

- prosody has broad functionality:
  - word level:
    - phonemic, morphemic
  - sentence level:
    - phrasal, sentential
  - utterance level:
    - dialogue acts, turn-taking
- broader than usual concept of language architecture is needed
  - the Rank-Interpretation Model (RIM)

Rank Interpretation Model - schematic



Rank Interpretation Model - prosody+



Operational models

Domains for operational models

Rhythm:

Linguistic models

as in

- GP cycle, LexPhon, ...
- A-M phon & pros

Phonetic models

as in

- global statistics
  - Roach, &c.:
    - SD, invariance, ...
  - Nolan, &c.:
    - PVI
- quantitative local relations
  - Gibbon
    - tree induction

Melody:

Linguistic models

as in

- functional contour models
- structural level models

Phonetic models

as in

- Fujisaki
- PH & Liberman
- Tilt
- ... (many others)

Domains for operational models

Rhythm:

Linguistic models
as in
- GP cycle
- A-M phonology

Phonetic models
as in
- global statistics
  - SD, invariance, ...
  - PVI
- tree induction

Melody:

Linguistic models
as in
- functional contour models
- structural level models

Phonetic models
as in
- Fujisaki
- PH & Liberman
- Tilt
- ... (many others)

Developing operational tone models

1. Tone computing scenarios
   - Tone computing: theory formation: generate-and-test
   - Tone computing: empirical heuristics: statistical analysis
   - Tone computing: applications: speech synthesis, recognition, diagnosis
2. Architecture and Workflow
   - General TTS synthesis architecture
   - MBROLA synthesis workflow
   - Annotation
3. Practical heuristic synthesis
   - Geeky stuff
   - Sample I/O of pitch interpreter
   - 12 Nigerian languages
   - So how is the synthetic voice created?
   - Implementation demo

Developing operational tone models

1. Tone computing scenarios
   - Tone computing: theory formation: generate-and-test
   - Tone computing: empirical heuristics: statistical analysis
   - Tone computing: applications: speech synthesis, recognition, diagnosis
2. Architecture and Workflow
   - General TTS synthesis architecture
   - MBROLA synthesis workflow
   - Annotation
3. Practical heuristic synthesis
   - Geeky stuff
   - Sample I/O of pitch interpreter
   - 12 Nigerian languages
   - So how is the synthetic voice created?
   - Implementation demo

Developing operational tone models

1. Tone computing scenarios

- Tone computing: theory formation: generate-and-test
- Tone computing: empirical heuristics: statistical analysis
- Tone computing: applications: speech synthesis, recognition, diagnosis

2. Architecture and Workflow

- General TTS synthesis architecture
- MBROLA synthesis workflow
- Annotation

3. Practical heuristic synthesis

- Geeky stuff
- Sample I/O of pitch interpreter
- 12 Nigerian languages
- So how is the synthetic voice created?
- Implementation demo

Tone computing scenarios

Contexts for tone computing

1) Tone computing in theory formation
   - Inputs → [grammars] → outputs

     + parsers and generators

     modelled asFS tone automata
2) Tone computing as empirical heuristics
   - Scripted and unscripted ‘natural’ speech
   - Tone sequencing
     - Ibibio low tone sequencing; tone correlation
   - Tone modelling
     - Thadou tone verification
3) Tone computing as application
   - Verifying transcriptions and annotations
   - Teaching phonetics, phonology, prosody
   - Interdisciplinary communication
     - e.g. between linguists, phoneticians and engineers

Contexts for tone computing

1) Tone computing in theory formation
   - Inputs → [grammars] → outputs

     + parsers and generators

     modelled asFS tone automata
2) Tone computing as empirical heuristics
   - Scripted and unscripted 'natural' speech
   - Tone sequencing
     - Ibibio low tone sequencing; tone correlation
   - Tone modelling
     - Thadou tone verification
3) Tone computing as application
   - Verifying transcriptions and annotations
   - Teaching phonetics, phonology, prosody
   - Interdisciplinary communication
     - e.g. between linguists, phoneticians and engineers

Contexts for tone computing

1) Tone computing in theory formation
   - Inputs → [grammars] → outputs
     + parsers and generators
     modelled asFS tone automata
2) Tone computing as empirical heuristics
   - Scripted and unscripted ‘natural’ speech
   - Tone sequencing
     - Ibibio low tone sequencing; tone correlation
   - Tone modelling
     - Thadou tone verification
3) Tone computing as application
   - Verifying transcriptions and annotations
   - Teaching phonetics, phonology, prosody
   - Interdisciplinary communication
     - e.g. between linguists, phoneticians and engineers

Scenario 1

Operational models in theory formation

Specifics

Which models?

- Automata models for prosodic theory:
  - tone terrace automata (for intonation, cf. Fujisaki, IPO, JPH, DG)
    - Baule, Tem, Ega
  - multitier automata for grammatical tone
    - Ibibio

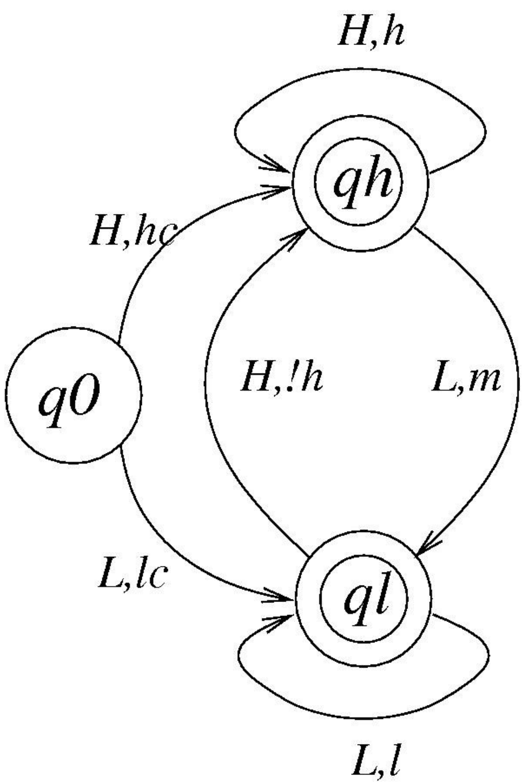
- Operational models for speech synthesis of Niger-Congo tone languages:
  - Abuja hands-on workshop 2010:
    - Adegbola, Barnard, Ekpenyong, Gibbon, Odejobi, Salffner, Urua
    - microvoices for 12 Nigerian languages (approx. 3%!)

      Anaang, Efik, Esan, Ibibio, Igala, Igbo, Itu Mbonuso, Leggbo, Nembe, Oron, Yagba, Yoruba
    - post-workshop outcome - synthetic voice for Igbo marketplace domain:

      Ugonna V. Duruide (2010), "A Preliminary Igbo text-to-speech application". BA thesis, U Ibadan.

Tone computing: theory formation

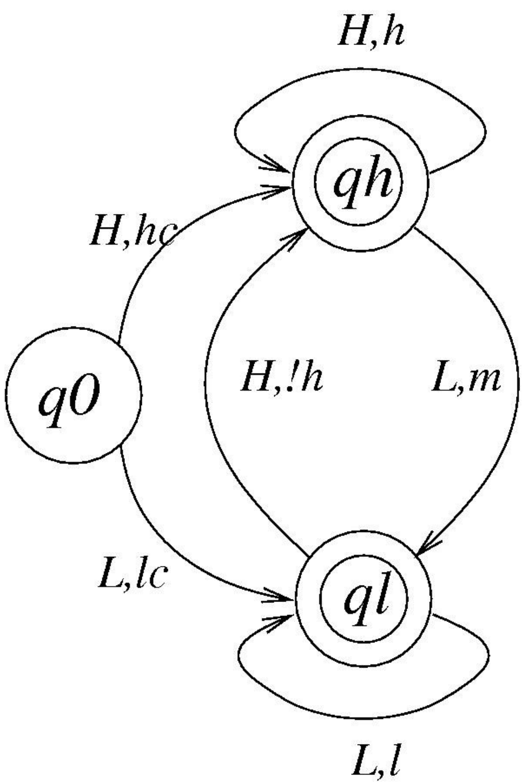
Finite State tone automata

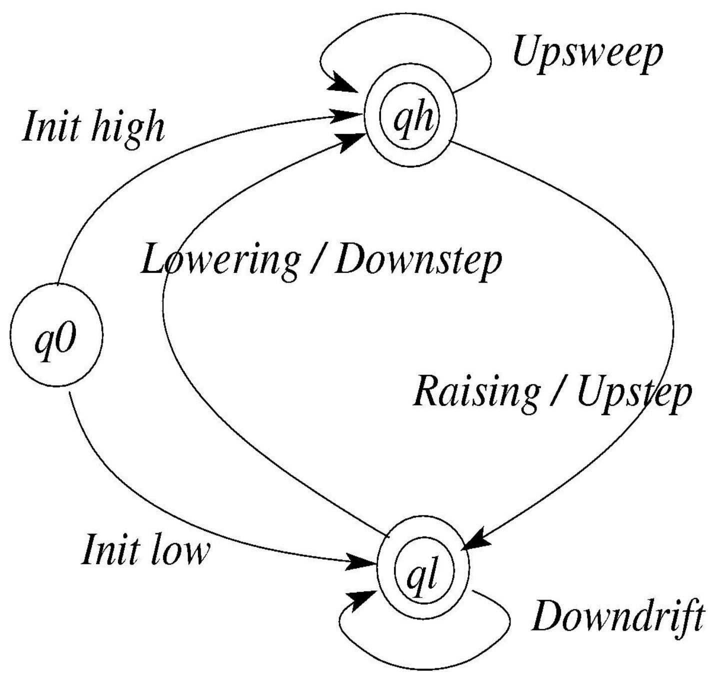


System with
2 register tones
& terracing

Tone computing: theory formation

Finite State tone automata

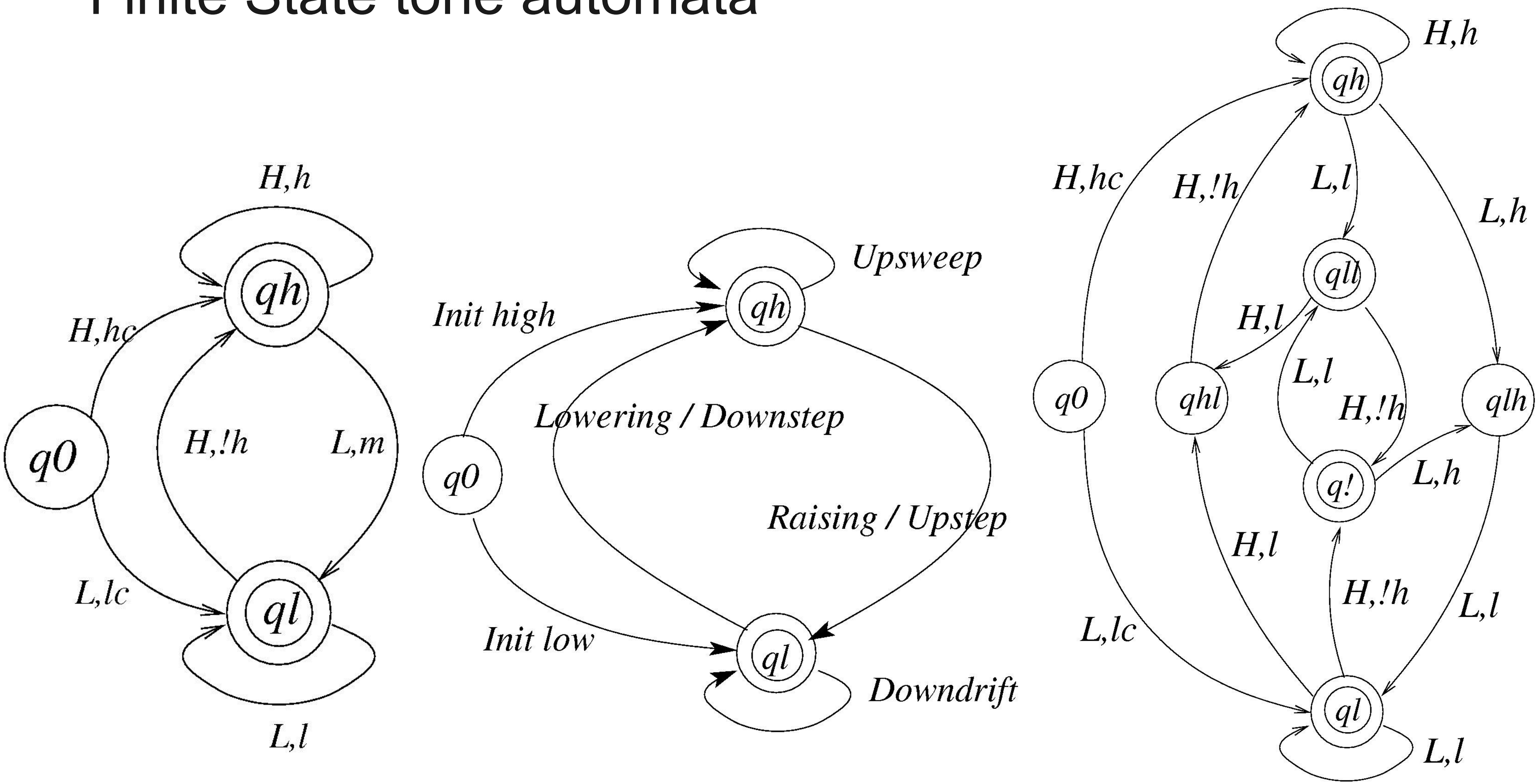




Operational model
of system with
2 register tones
& terracing

Tone computing: theory formation

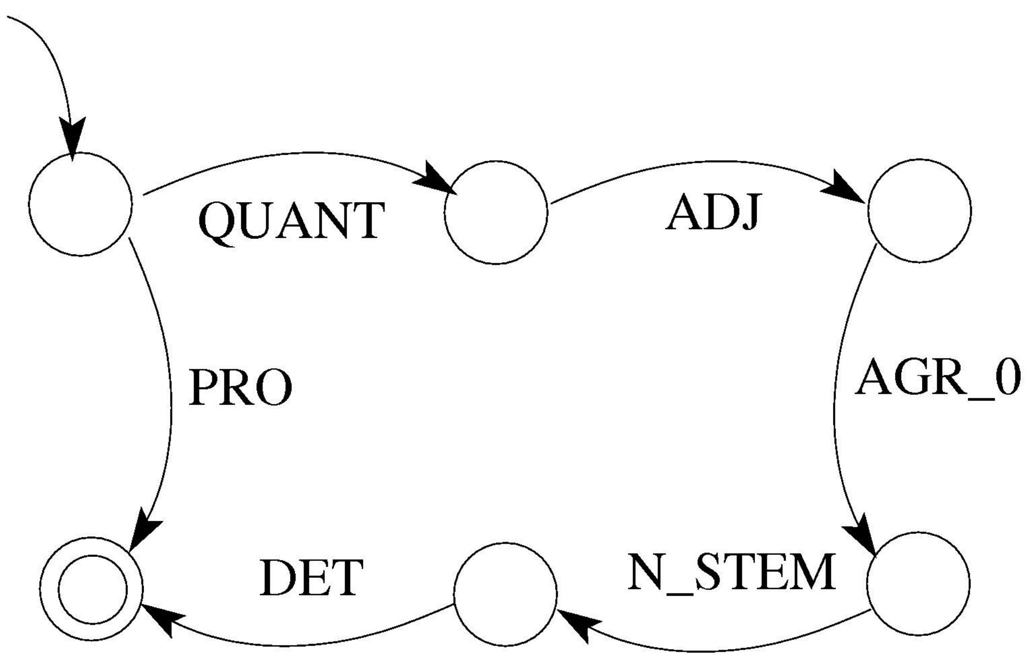
Finite State tone automata



System with
2 register tones
& terracing with
'lookahead'

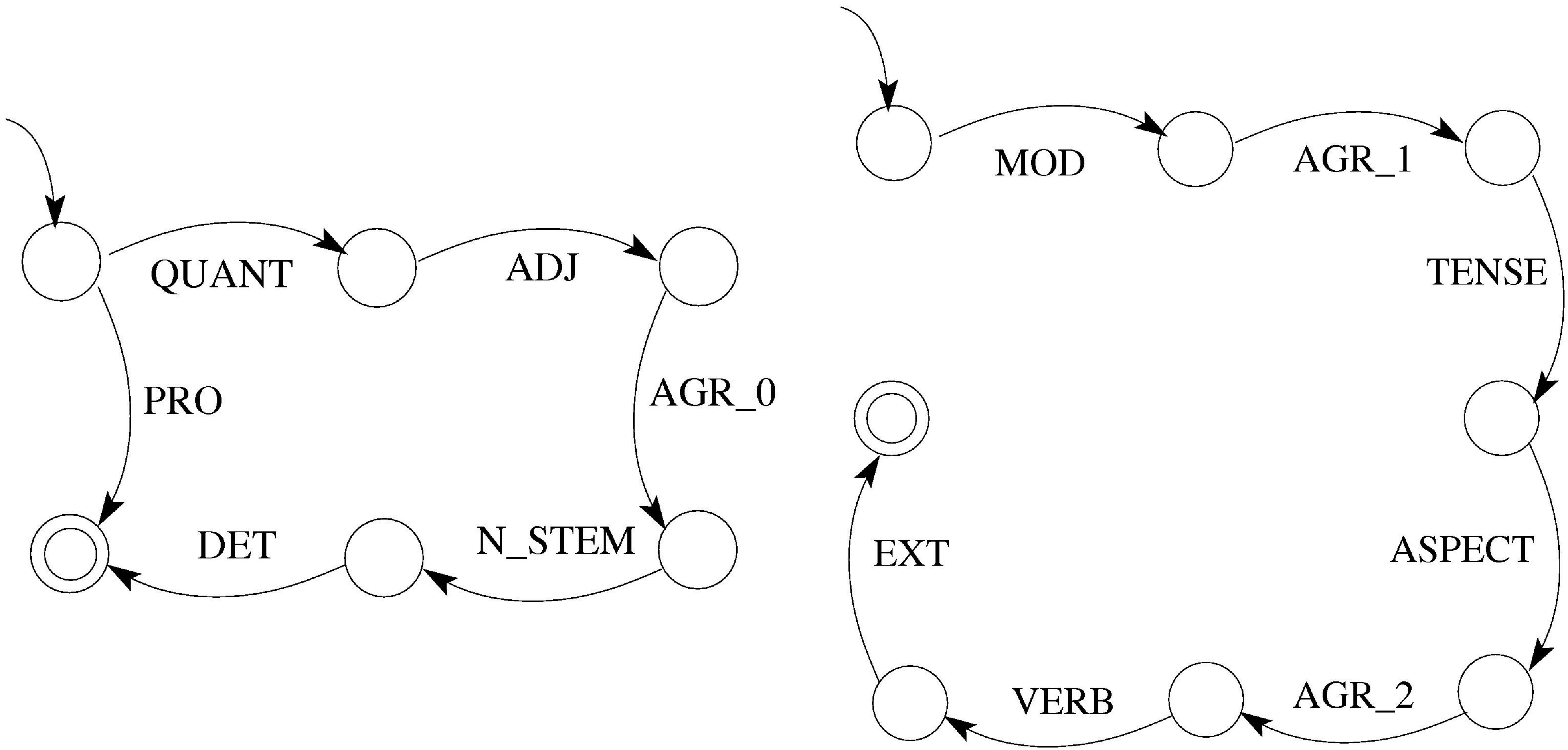
Tone computing: theory formation

Ibibio grammar (for morpho-tonotactics):



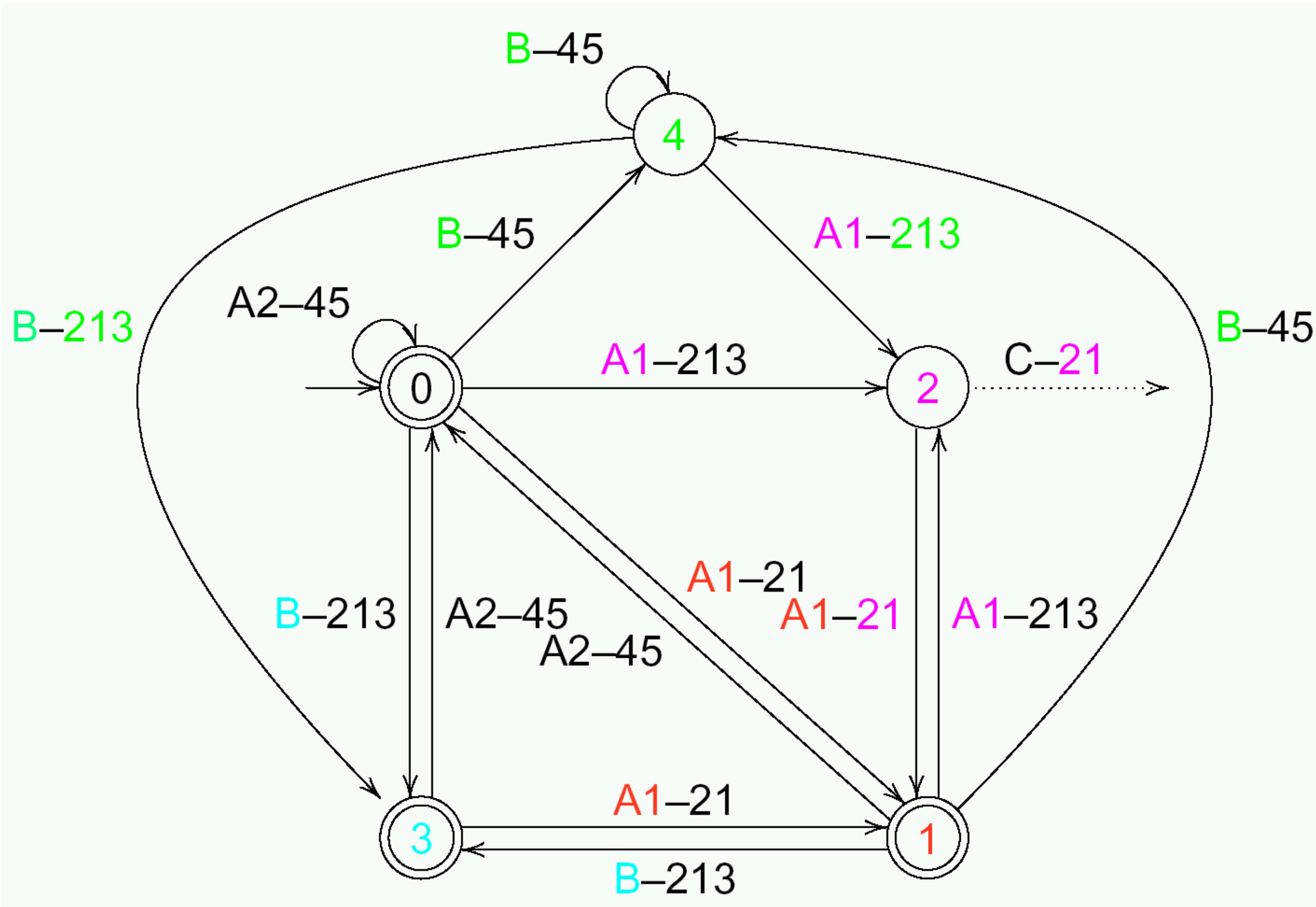
Tone computing: theory formation

Main parts of Ibibio grammar which are relevant for morpho-tonotactics:



Productivity of the approach

Jansche's automaton model of Tian-jin Mandarin tone:



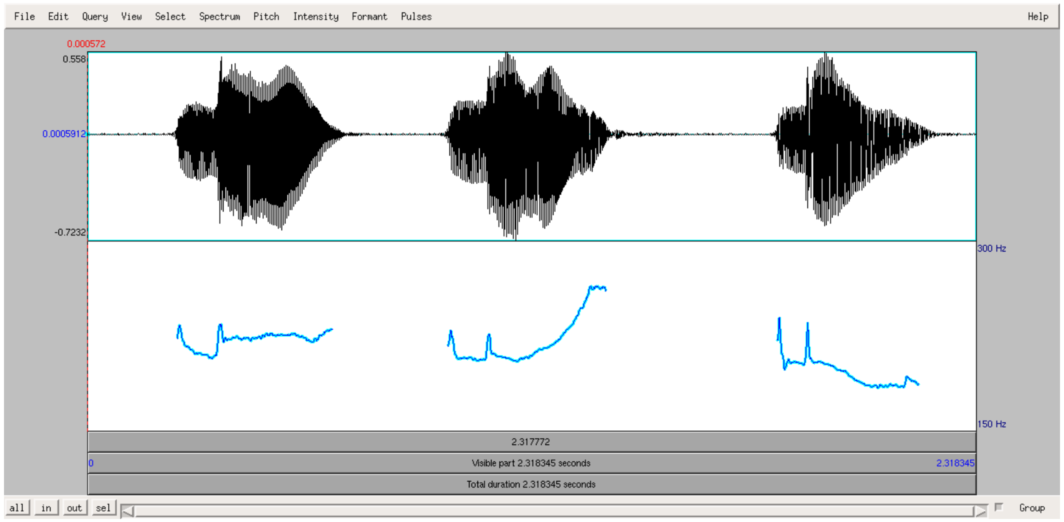
Scenario 2

Empirical heuristics

for operational models

Tone computing: empirical heuristics

Thadou tone verification

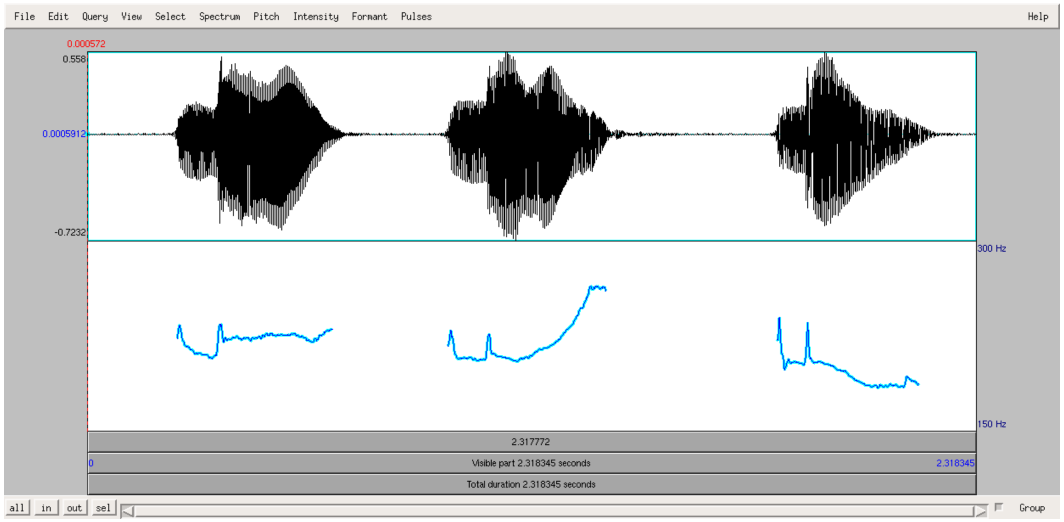


Thadou tones:

- *lów* (H) ‘field’,
- *lǒw* (LH) ‘medicine’,
- *lòw* (L) ‘negative marker’.

Tone computing: empirical heuristics

Thadou tone verification



Thadou tones:

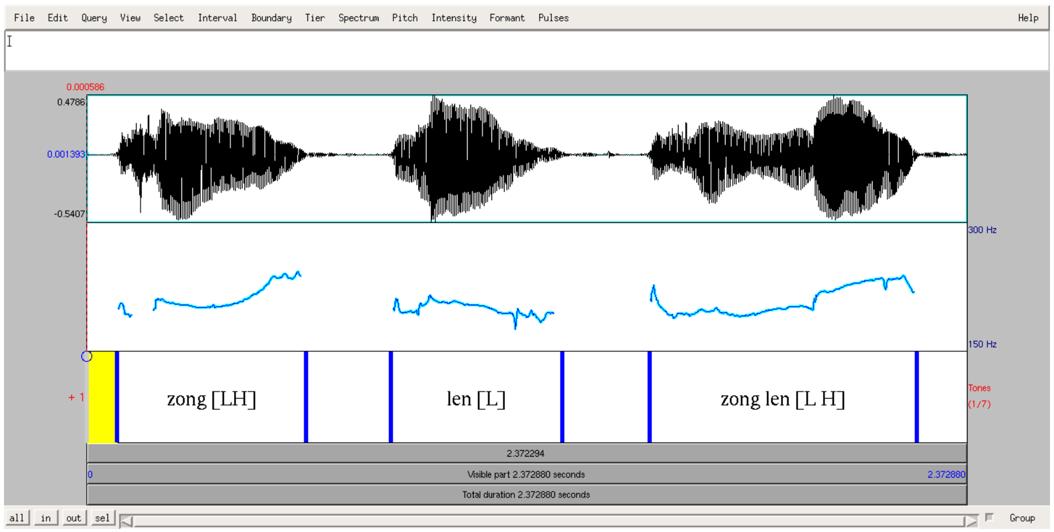
lów (H) ‘field’,
lǒw (LH) ‘medicine’,
lòw (L) ‘negative marker’.

The descriptive statistics are over averages of 16 pitch samples for each of 3 occurrences of each vowel with which each tone is associated (e.g. 864=18x16x43).
The values over all measurement sets per tone are in parentheses.

Tone	N	min	max	mean	sd	offset	slope
H	18 (864)	200 (220)	244 (222)	221	0.29	221	-0.03
LH	17 (816)	215 (198)	237 (268)	220	7.07	209	1.3
L	18 (864)	192 (178)	213 (227)	203	6.3	215	-1.31

Tone computing: empirical heuristics

Thadou tone verification



LH *zǒng* 'monkey'
L *lèn* 'big' tones in isolation
L+H *zòng lén* 'bit monkey' tone sequence
Note H tone shift and L deletion.

Operational model (close copy synthesis): úy tsôm ('dog' 'short') 'short dog'

(Hyman's tone marks)

All of the preceding feeds into...

Scenario 3

Operational models

in applications

Tone computing: application

- Speech technology:
  - speech recognition
  - speech synthesis
  - language identification
  - speaker identification
- Interdisciplinary cooperation:
  - Communication between linguists, phoneticians and engineers
  - Evaluation of data:
    - automatic formal verification of transcriptions and annotations
- Teaching
    - Classes in Bielefeld and various other places
    - 2010 Abuja seminar: voicelets/microvoices for 11 Nigerian languages
    - BA thesis Ugonna Duruibe, Ibadan: Igbo voice (MBROLA)
    - Technology: esp. speech synthesis

Operational models: N-C tone

Operational models for speech synthesis of Nigerian Niger-Congo tone languages:

Abuja hands-on workshop 2010:

- Tutors:
  Adegbola, Barnard, Ekpenyong, Gibbon, Odejobi,Salffner, Urua, Wagner

- Students:
  - linguistics and computer science majors
  - various African universities

- Outcome:
  Microvoices for 12 N-C tone languages (approx. 3% of Nigerian languages)
  Anaang, Efik, Esan, Ibibio, Igala, Igbo, Itu Mbonuso, Leggbo, Nembe, Oron, Yagba, Yoruba

Post-workshop outcome - synthetic voice for Igbo marketplace domain:
Ugonna V. Duruide (2010), “A Preliminary Igbo text-to-speech application”. BA thesis, U Ibadan.

Operational model: linguistic specs

- Linguistic specification:
  - tiers:
    - intervals (segments)
    - phoneme labels
    - tone labels
- Linguistic-phonetic interface:
  - theory-based linguistic model: prosodic matrix/tiers
  - operationalised as a set of [phoneme, duration, pitch] tuples.
- Operational synthesis:
  - phonemes are assigned acoustic properties using a database of recorded speech
  - intervals are associated with explicit durations
  - tones are implemented as F0 (pitch) trajectories in time

Operational model: linguistic specs

- Linguistic specification:
  - tiers:
    - intervals (segments)
    - phoneme labels
    - tone labels
- Linguistic-phonetic interface:
  - theory-based linguistic model: prosodic matrix/tiers
  - operationalised as a set of [phoneme, duration, pitch] tuples.
- Operational synthesis:
  - phonemes are assigned acoustic properties using a database of recorded speech
  - intervals are associated with explicit durations
  - tones are implemented as F0 (pitch) trajectories in time

Operational model: linguistic specs

- Linguistic specification:
  - tiers:
    - intervals (segments)
    - phoneme labels
    - tone labels
- Linguistic-phonetic interface:
  - theory-based linguistic model: prosodic matrix/tiers
  - operationalised as a set of [phoneme, duration, pitch] tuples.
- Operational synthesis:
  - phonemes are assigned acoustic properties using a database of recorded speech
  - intervals are associated with explicit durations
  - tones are implemented as F0 (pitch) trajectories in time

Operational model: technical specs

- Task-specific requirements specification:
  - Users:
    - academic learning environment
      - rather than engineering workplace
    - usability by linguists and phoneticians → minimum math :)
    - usability in West African environment → not new or online
  - Intended output:
    - *perceptual similarity* rather than exact acoustic modelling
  - Development strategy:
    - basic script programming & testing
      - flexible hands-on command-line modelling
      - rather than fixed consumer GUI (Graphical User Interface)
        - GUI comes later
  - Software decision:
    - *clear linguistic interface* for tone / frequency → MBROLA
    - widely available, free, simple, tested → MBROLA
    - interoperable on Linux, Mac, Win → MBROLA

Operational model: technical specs

- Task-specific requirements specification:
  - Users:
    - academic learning environment
      - rather than engineering workplace
    - usability by linguists and phoneticians → minimum math :)
    - usability in West African environment → not new or online
  - Intended output:
    - *perceptual similarity* rather than exact acoustic modelling
  - Development strategy:
    - basic script programming & testing
      - flexible hands-on command-line modelling
      - rather than fixed consumer GUI (Graphical User Interface)
        - GUI comes later
  - Software decision:
    - *clear linguistic interface* for tone / frequency → MBROLA
    - widely available, free, simple, tested → MBROLA
    - interoperable on Linux, Mac, Win → MBROLA

Operational model: technical specs

- Task-specific requirements specification:
  - Users:
    - academic learning environment
      - rather than engineering workplace
    - usability by linguists and phoneticians → minimum math :)
    - usability in West African environment → not new or online
  - Intended output:
    - *perceptual similarity* rather than exact acoustic modelling
  - Development strategy:
    - basic script programming & testing
      - flexible hands-on command-line modelling
      - rather than fixed consumer GUI (Graphical User Interface)
        - GUI comes later
  - Software decision:
    - *clear linguistic interface* for tone / frequency → MBROLA
    - widely available, free, simple, tested → MBROLA
    - interoperable on Linux, Mac, Win → MBROLA

Operational model: technical specs

- Task-specific requirements specification:
  - Users:
    - academic learning environment
      - rather than engineering workplace
    - usability by linguists and phoneticians → minimum math :)
    - usability in West African environment → not new or online
  - Intended output:
    - *perceptual similarity* rather than exact acoustic modelling
  - Development strategy:
    - basic script programming & testing
      - flexible hands-on command-line modelling
      - rather than fixed consumer GUI (Graphical User Interface)
        - GUI comes later
  - Software decision:
    - *clear linguistic interface* for tone / frequency → MBROLA
    - widely available, free, simple, tested → MBROLA
    - interoperable on Linux, Mac, Win → MBROLA

Operational model: technical specs

- Task-specific requirements specification:
  - Users:
    - academic learning environment
      - rather than engineering workplace
    - usability by linguists and phoneticians → minimum math :)
    - usability in West African environment → not new or online
  - Intended output:
    - *perceptual similarity* rather than exact acoustic modelling
  - Development strategy:
    - basic script programming & testing
      - flexible hands-on command-line modelling
      - rather than fixed consumer GUI (Graphical User Interface)
        - GUI comes later
  - Software decision:
    - *clear linguistic interface* for tone / frequency → MBROLA
    - widely available, free, simple, tested → MBROLA
    - interoperable on Linux, Mac, Win → MBROLA

Architecture and workflow

of operational model

Architecture and workflow

of operational model

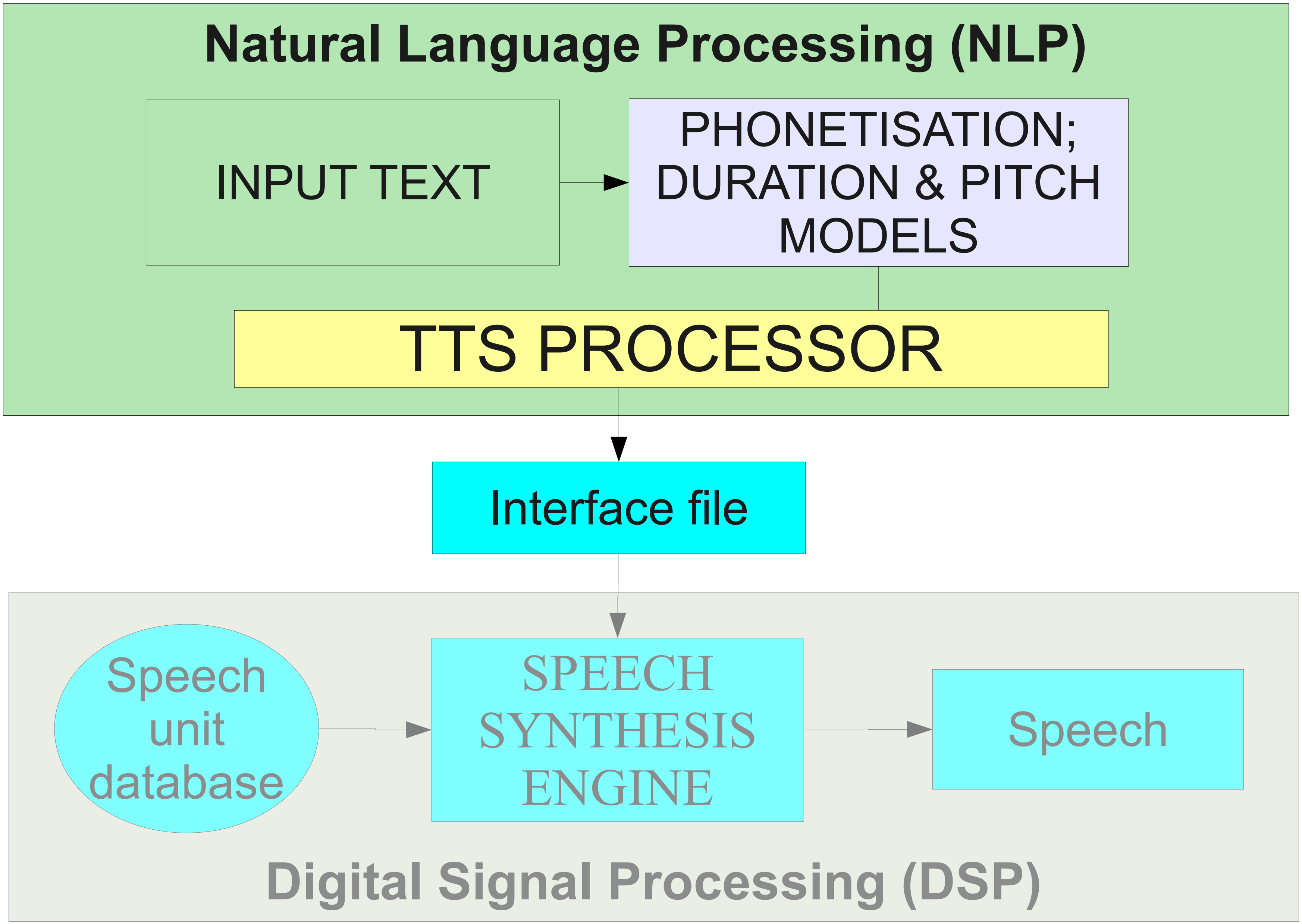
... with <u>real</u> interfaces

Architecture and data flow

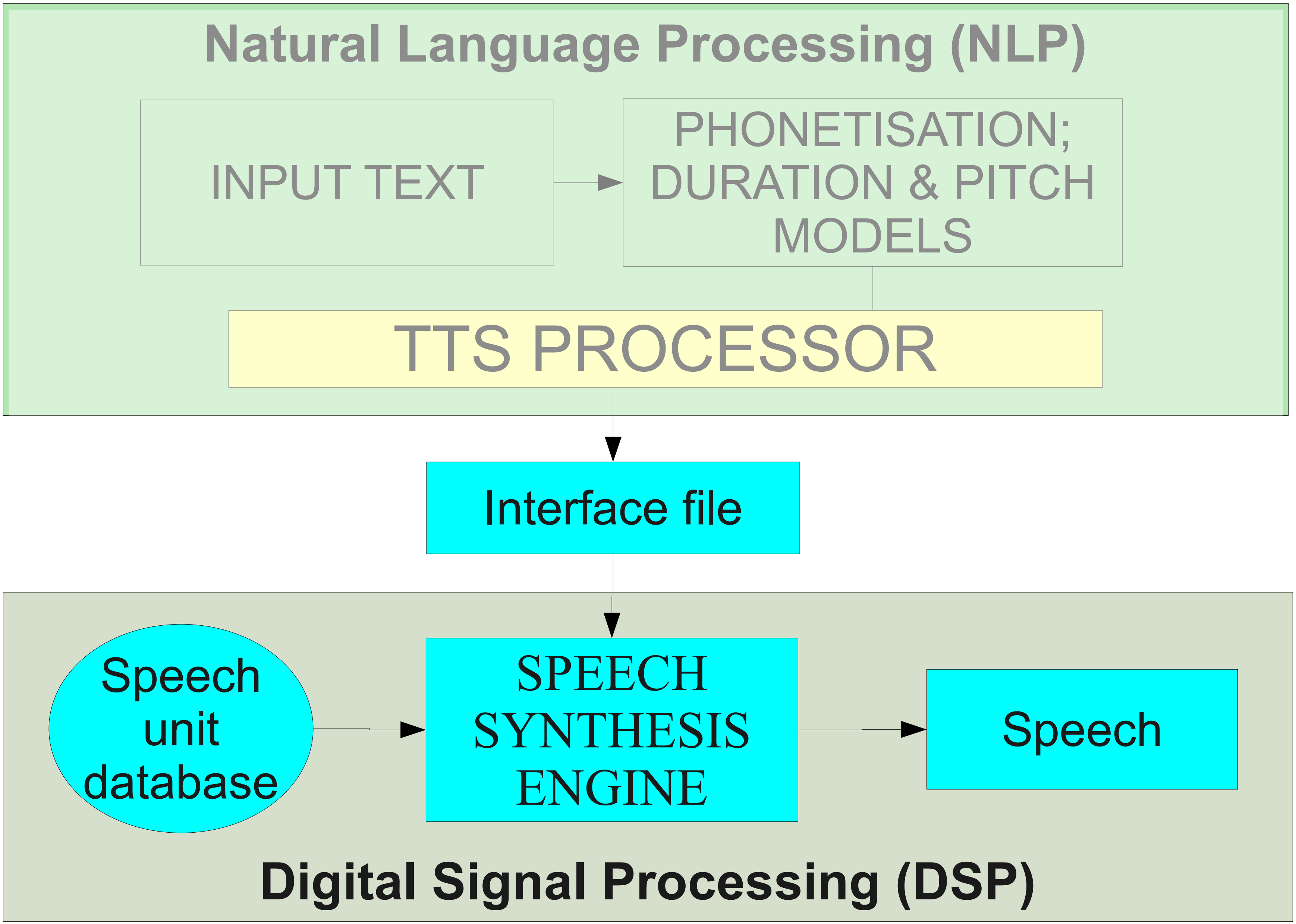
of operational model

... with <u>real</u> interfaces ☺

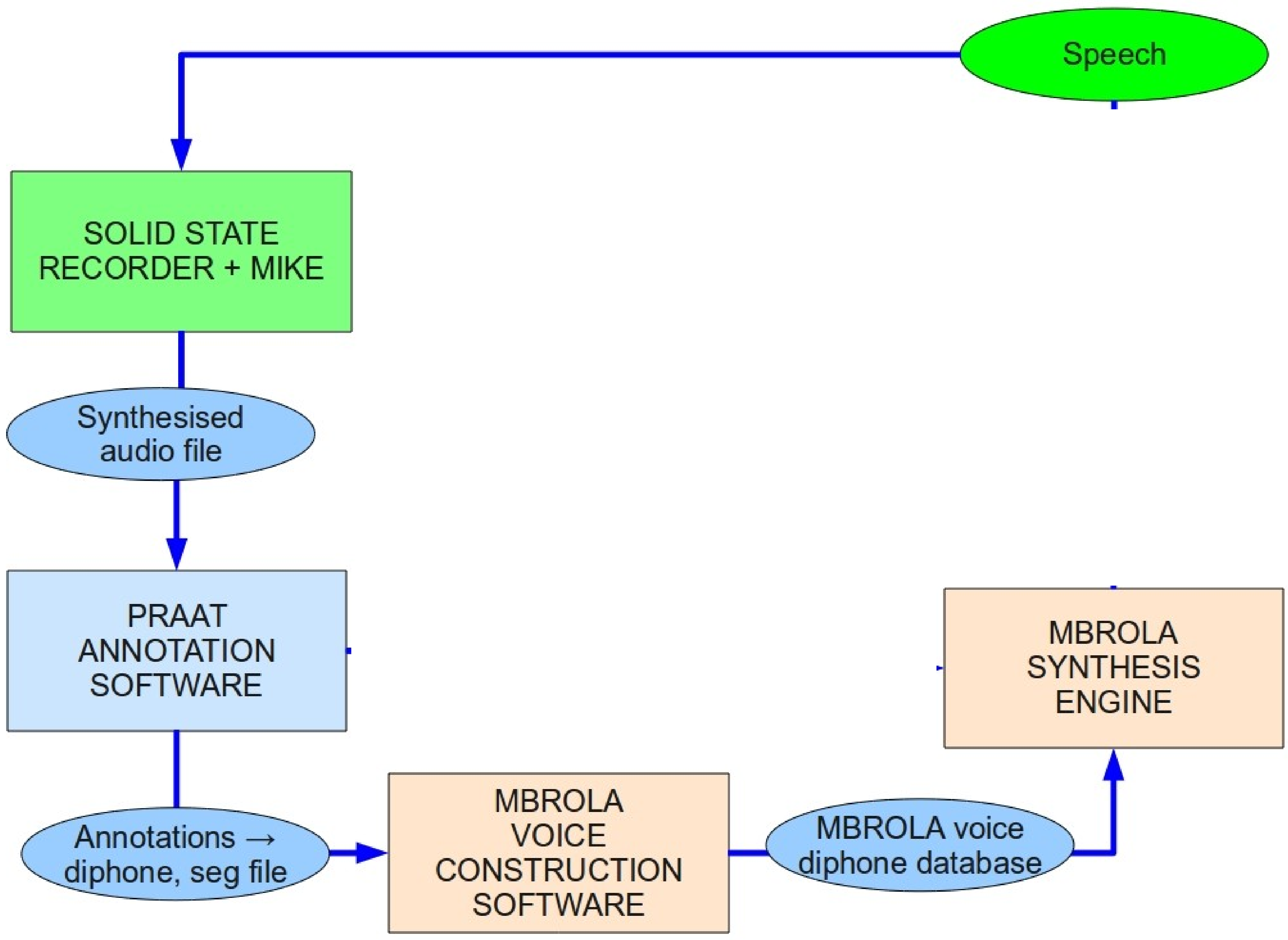
General TTS synthesis architecture



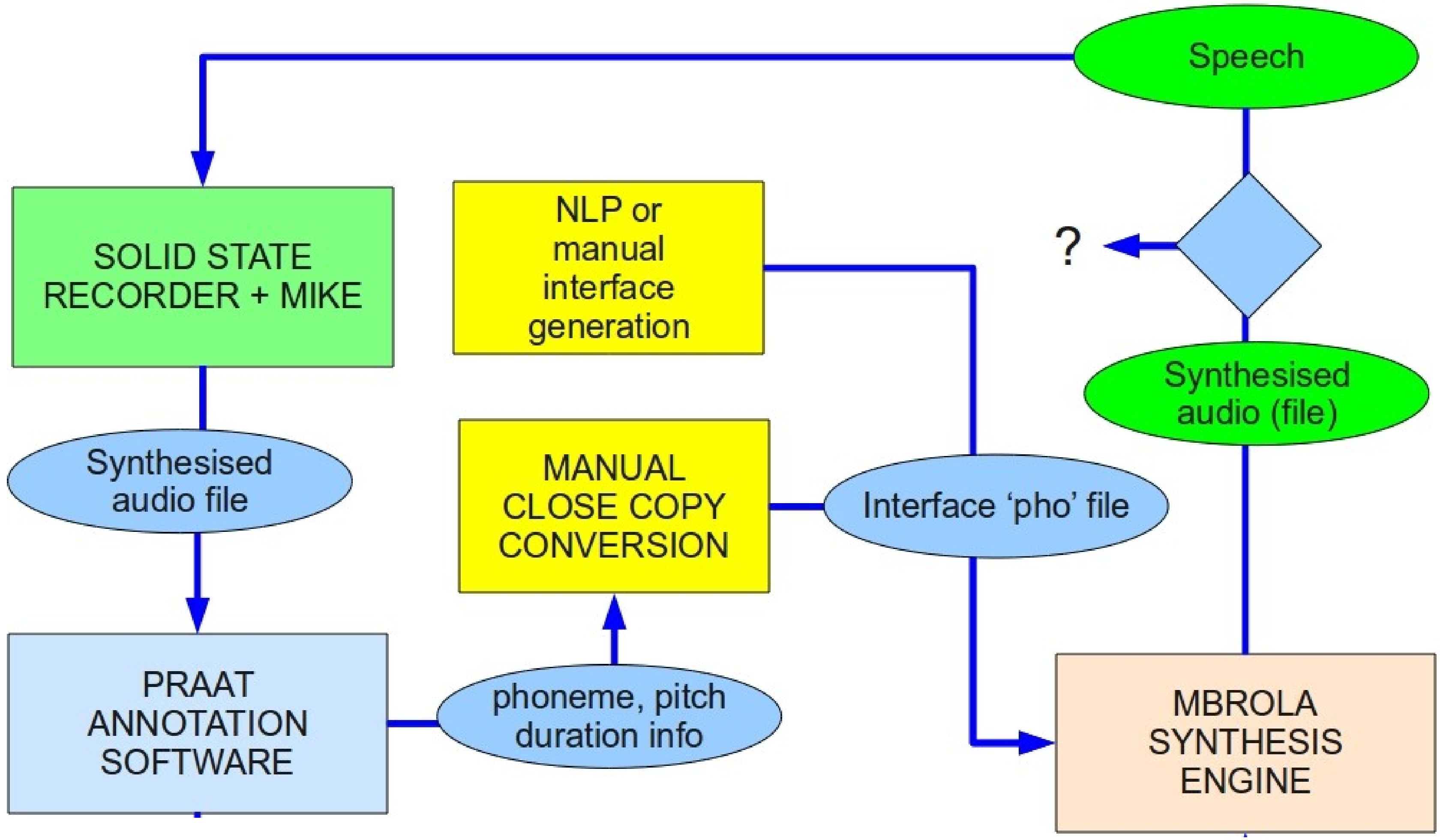
General TTS synthesis architecture



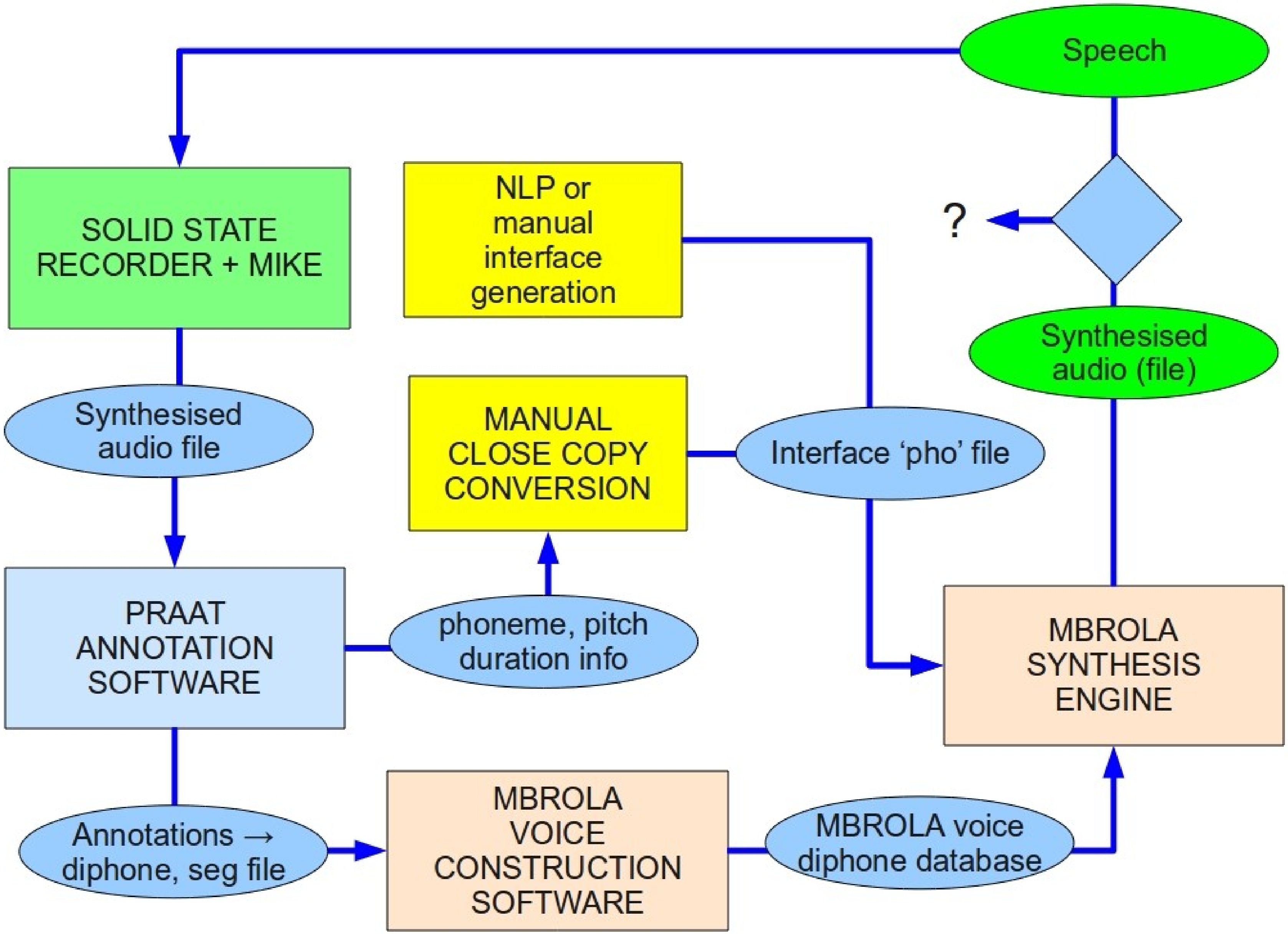
MBROLA voice-creation workflow



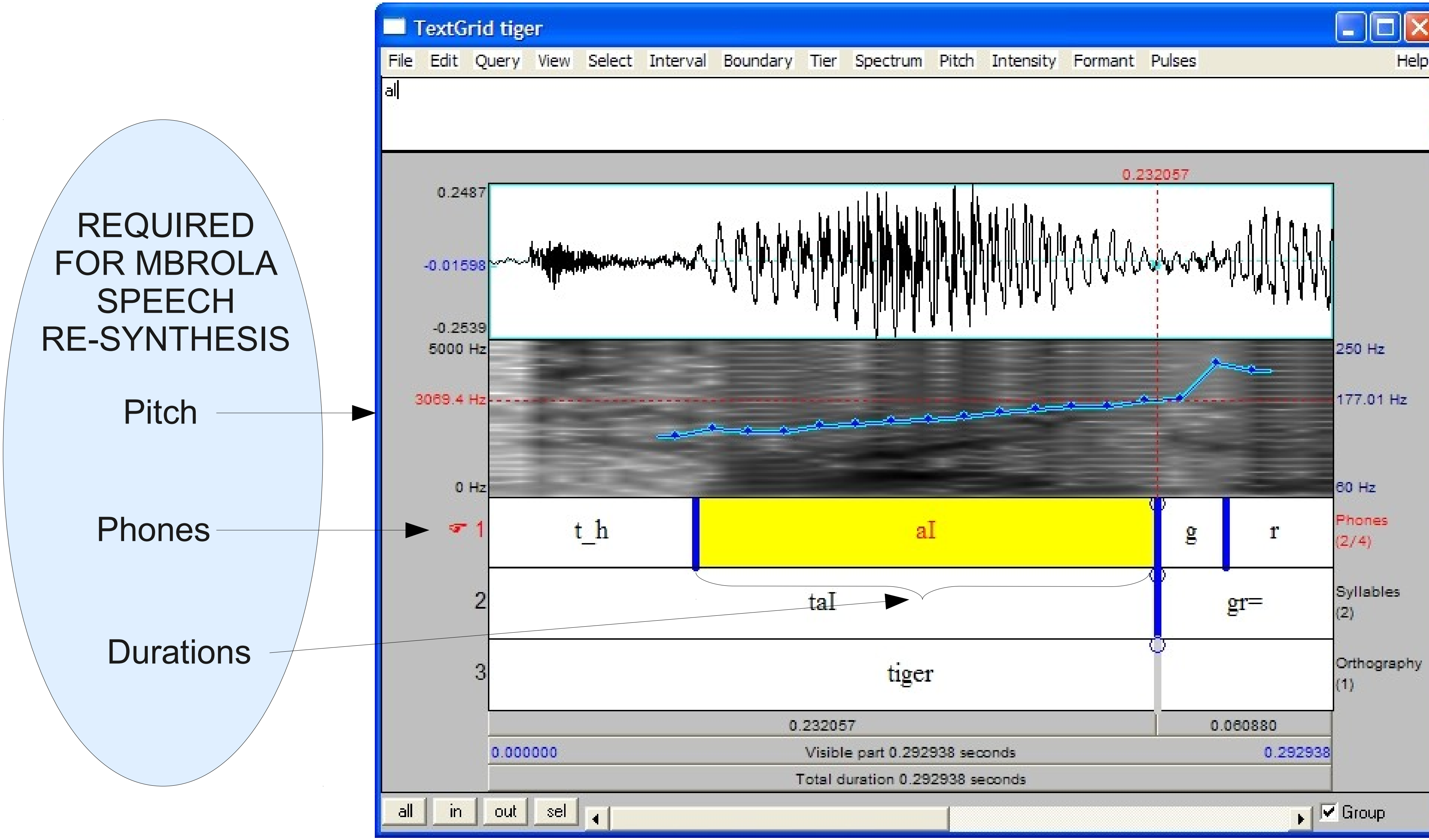
MBROLA synthesis data flow



MBROLA overall workflow



Annotation


REQUIRED FOR MBROLA SPEECH RE-SYNTHESIS
Pitch
Phones
Durations
TextGrid tiger
File Edit Query View Select Interval Boundary Tier Spectrum Pitch Intensity Formant Pulses Help
aI
0.232057
0.2487
-0.01598
-0.2539
5000 Hz
3069.4 Hz
0 Hz
250 Hz
177.01 Hz
60 Hz
1
t_h
aI
g
r
Phones (2/4)
2
taI
gr=
Syllables (2)
3
tiger
Orthography (1)
0.232057
0.060880
0.000000
Visible part 0.292938 seconds
0.292938
Total duration 0.292938 seconds
all
in
out
sel
Group

Praat annotation language

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 0.2929375
tiers? <exists>
size = 1
item []:
    item [1]:
        class = "IntervalTier"
        name = "Phones"
        xmin = 0
        xmax = 0.2929375
        intervals: size = 4
```

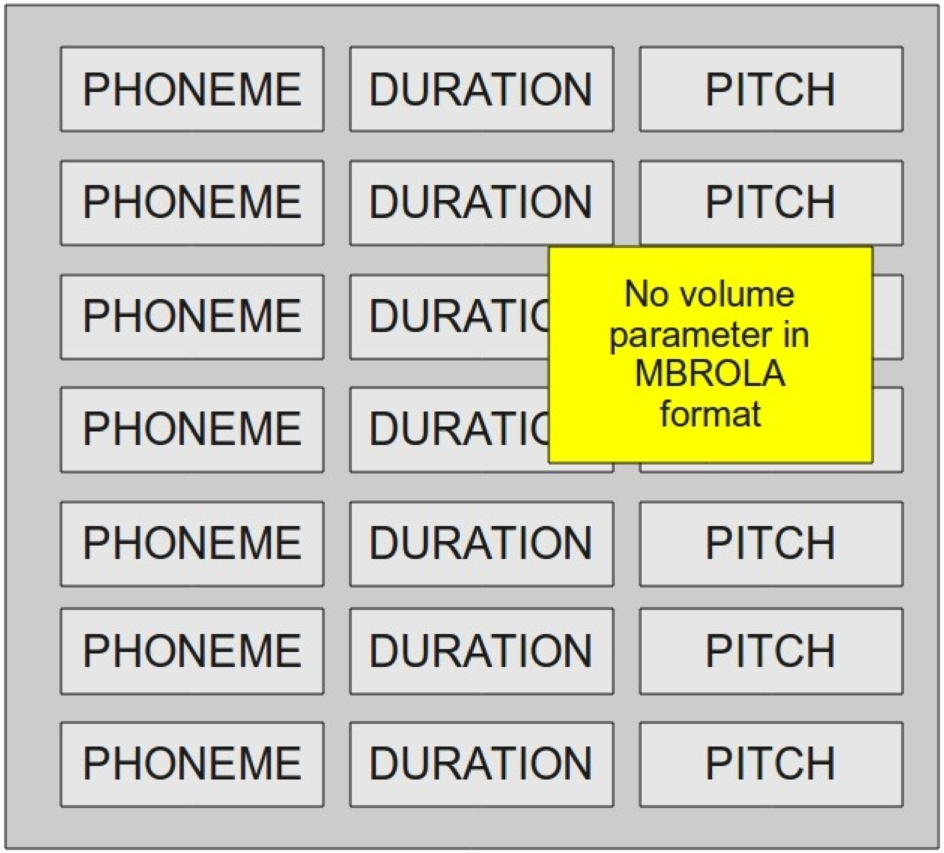
```
        intervals [1]:
            xmin = 0
            xmax = 0.0718
            text = "t_h"
        intervals [2]:
            xmin = 0.0718
            xmax = 0.2323
            text = "aI"
        intervals [3]:
            xmin = 0.2323
            xmax = 0.25579
            text = "g"
        intervals [4]:
            xmin = 0.2557
            xmax = 0.2929
            text = "r\`="
```

Praat annotation language

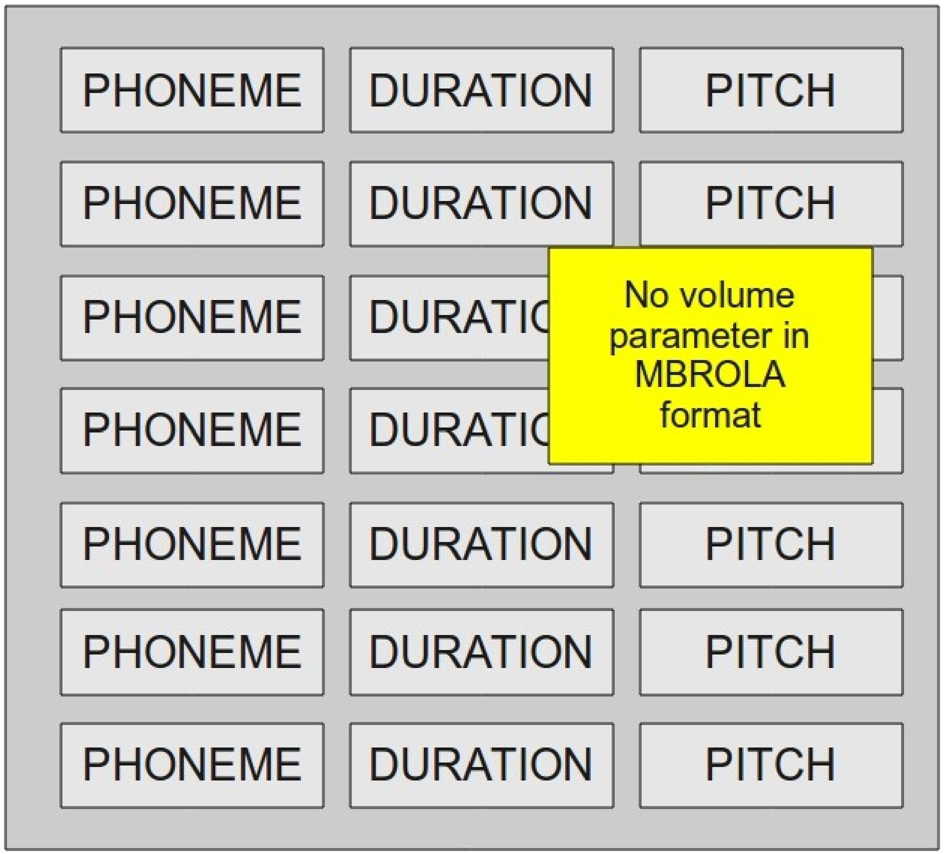
```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 0.2929375
tiers? <exists>
size = 1
item []:
    item [1]:
        class = "IntervalTier"
        name = "Phones"
        xmin = 0
        xmax = 0.2929375
        intervals: size = 4
```

```
        intervals [1]:
            xmin = 0
            xmax = 0.0718
            text = "t_h"
        intervals [2]:
            xmin = 0.0718
            xmax = 0.2323
            text = "aI"
        intervals [3]:
            xmin = 0.2323
            xmax = 0.25579
            text = "g"
        intervals [4]:
            xmin = 0.2557
            xmax = 0.2929
            text = "r\`="
```

MBROLA interface language

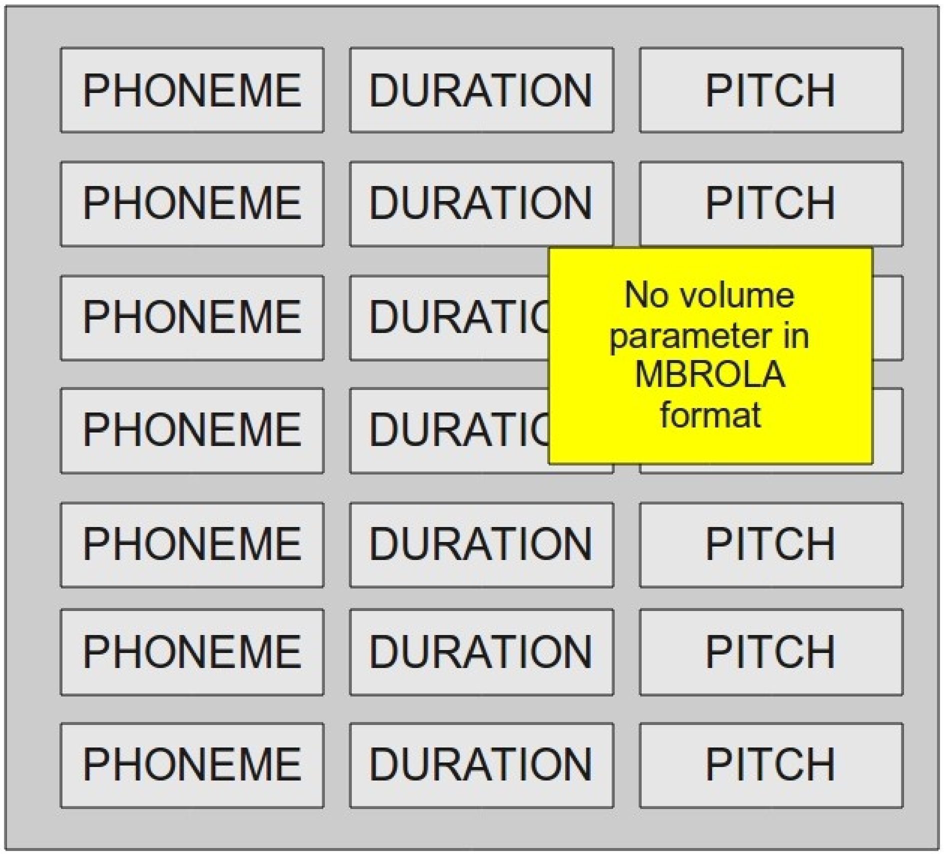


MBROLA interface language



Phoneme	Duration	Pitch pos-val pairs	
(SAMPA)	(ms)	%pos Hz	%pos Hz
I	60	75 109	
n	80		
v	40		
e	90	0 109	50 153
n	30	25 195	
S	80		
@	70	50 20	
n	30		

MBROLA interface language



Phoneme	Duration	Pitch pos-val pairs	
(SAMPA)	(ms)	%pos Hz	%pos Hz
I	60	75 109	
n	80		
v	40		
e	90	0 109	50 153
n	30	25 195	
S	80		
@	70	50 20	
n	30		

Close copy synthesis:

convert annotation format directly to interface format

→ manual

→ Python tool (also on the web)

Simple pitch assignment algorithm

For each row in the file:

Input:

- [phoneme duration tone]
  - (where tone is either unmarked or marked HIGH)

Method:

- if tone mark is HIGH
  - assign current pitch line with tone increment
    - (where tone increment is a fraction of the current pitch line)
- else
  - assign current pitch line
- calculate asymptotic declination for next row
  - where declination factor <0 (falling), 0 (monotone), >0 (rising)

Output:

- [phoneme duration [placement pitch]]

Geeky stuff

Read command parameters
Set variables

for each line in file:

```
# Reset pitchline at pause:

    if (phoneme == pause)
        pitchline = onsetpitch

# Assign pitch to vowels:

    if (tone >= 0 && vowel ~ phoneme)
        if (row ~ "HIGH")
            outputpitch = pitchline + (pitchline * tone)
        else
            outputpitch = pitchline
        freqstring = position outputpitch

# Use original frequency:

    if (tone < 0 && frequency != 0)
        outputpitch = frequency
        freqstring = position outputpitch

# Calculate declination line for next row:

    pitchline = pitchline * declination

# Output synthesis values to interface file:

    print phoneme duration freqstring
```

Geeky stuff

Read command parameters
Set variables

for each line in file:

```
# Reset pitchline at pause:

    if (phoneme == pause)
        pitchline = onsetpitch

# Assign pitch to vowels:

    if (tone >= 0 && vowel ~ phoneme)
        if (row ~ "HIGH")
            outputpitch = pitchline + (pitchline * tone)
        else
            outputpitch = pitchline
        freqstring = position outputpitch
```

Note:

This demo model is kept as simple as possible, consistent with reasonable results.

e.g.: no baseline, no final fall

```
# Use original frequency:

    if (tone < 0 && frequency != 0)
        outputpitch = frequency
        freqstring = position outputpitch

# Calculate declination line for next row:

    pitchline = pitchline * declination

# Output synthesis values to interface file:

    print phoneme duration freqstring
```

Sample I/O of pitch interpreter

INPUT: `pho-pitch-assign-04.sh UgoIgbo01 200 0.92 0.3 test02-short.pho`

Sample I/O of pitch interpreter

INPUT: `pho-pitch-assign-04.sh UgoIgbo01 200 0.92 0.3 test02-short.pho`

voice onset declination tone interface

Sample I/O of pitch interpreter

INPUT:

```
pho-pitch-assign-04.sh UgoIgbo01 200 0.92 0.3 test02-short.pho
```

voice onset declination tone interface

OUTPUT:

```
_   200
a   301 50  184
gw  223
a   169 50  155
_   1445
O   331 50  239
k   162
a   231 50  155
_   1296
dZ  296
i   216 50  220
```

```
_   1250
o   110 50  184
s   123
i   90  50  155
k   128
a   156 50  171
p   106
a   160 50  145
```

```
_   1095
E   282 50  239
d   174
E   145 50  155
_   1237
dZ  131
i   71  50  169
a   238 50  202
pY  129
U   162 50  171
_   1170
```

Sample I/O of pitch interpreter

INPUT:

```
pho-pitch-assign-04.sh UgoIgbo01 200 0.92 0.3 test02-short.pho
```

voice onset declination tone interface

OUTPUT:

```
_   200
a   301 50  184
gw  223
a   169 50  155
_   1445
O   331 50  239
k   162
a   231 50  155
_   1296
dZ  296
i   216 50  220
```

```
_   1250
o   110 50  184
s   123
i   90  50  155
k   128
a   156 50  171
p   106
a   160 50  145
```

```
_   1095
E   282 50  239
d   174
E   145 50  155
_   1237
dZ  131
i   71  50  169
a   238 50  202
pY  129
U   162 50  171
_   1170
```

phoneme duration %-placement frequency

Test...

Abuja WB Step B teaching plan

- Goals
  - enhancing skills in phonetics
  - understanding of basic speech synthesis
- Prerequisites
  - knowledge of phonetics
  - experience with computer use
- Synthetic ‘tone-deaf’ microvoice creation:
  - Data: Speech recordings
  - Data: Annotations of phone boundaries
  - Data: Identification of phone centres (‘quasi-steady states’)
  - Processing: Cut out diphones with required extra time at beginning and end
  - Processing: Make a .seg file with time-stamp metadata for the diphones
  - Processing: Use the (licensable) Mbrolator software.
- Schedule:
  - Wednesday morning: recordings, annotations
  - Wednesday afternoon: voice-making (DG)
  - Friday morning: tests with voices
  - Saturday morning: tests with voices

How well does this work?

- Benchmark:
  - Igbo original recording (evaluation standard)
- Standard synthesis procedures:
  - Igbo close copy (implementation gold standard)
    - input from original data set as used for voice production
      - same phoneme combinations
      - same durations
      - same frequencies
  - Igbo generalisation (in progress)
    - input from different data
- Experimental tone synthesis:
  - Igbo partial close copy (computational tone model)
    - phonemes and durations from original data
    - tone assignment algorithm
      - declination line only
      - high tone increment only
      - neither :)

12 Nigerian languages

‘Tone deaf’ microvoices:

Anaang

Efik

Esan

Ibibio

Igala

Igbo

Itu Mbonuso

Leggbo

Nembe

Oron

Yagba

Yoruba

Approx. 3% of Nigerian languages...

ProsTest, an operational model

Interactive web tool with MBROLA

Thanks!

Data sources

with data mainly from
Eno-Abasi Urua
participants in the Abuja WB Step B Spring School 2010
Ugonna V. Duruide (2010):
“A Preliminary Igbo text-to-speech application”,
BA thesis, U Ibadan

Also Gibbon, Haokip, Pandey, Bachan

Relevant publications

Gibbon, Dafydd (1987). Finite state processing of tone languages. In: Proceedings of European ACL, Copenhagen.

Gibbon, Dafydd, Komdedzi Kofi Folikpo, Shu-Chuan Tseng (1996). Prosodic inheritance and phonetic interpretation: lexical tones in Ewegbe. Distributed, unpublished.

Gibbon, Dafydd (2001). Finite state prosodic analysis of African corpus resources, Proceedings of Eurospeech 2001, Aalborg, Denmark, I: pp. 83-86.

Gibbon, Dafydd (2002). Computational phonology and the typology of West African tone systems. In: Gut, U. & Gibbon, D., eds. (2002), Typology of African Prosodic Systems, Bielefeld: Bielefeld Occasional Papers in Typology 1.

Gibbon, Dafydd, Eno-Abasi Urua, Ulrike Gut (2003). A computational model of low tones in Ibibio. In: Proceedings of the International Congress of Phonetic Sciences, Barcelona, 2003, I: 623-626.

Gibbon, Dafydd (2004). Tone and timing: two problems and two methods for prosodic typology. In: Proceedings of the International Conference on Tonal Aspects of Languages, 28-30 March 2004, Beijing.

Gibbon, Dafydd, Eno-Abasi Urua & Moses Ekpenyong (2006). Problems and solutions in African tone language Text-To-Speech. In: Justus Roux, ed., Proceedings of the Multiling 2006 Conference, Stellenbosch, South Africa.

Gibbon, Dafydd and Eno-Abasi Urua (2006). Morphotonology for TTS in Niger-Congo languages. In: Proceedings of 2rd International Conference on Speech Prosody. Dresden: TUD Press.

Gibbon, Dafydd (In progress). Analysis and synthesis of lexical tone in discourse: Bete narrative prosody. WOCAL 2009, Köln. (Accepted presentation.)

Urua, Eno-Abasi & Dafydd Gibbon (In progress). Preserving and understanding the Medefaidrin language: a new contribution to Documentary Linguistics. WOCAL 2009, Köln, Germany. (Accepted presentation.)

Firmin Ahoua, Adjépole Kouamé & Dafydd Gibbon (In progress). Prosodic domains and tones in speech and songs in Anyi Sanvi. WOCAL 2009, Köln, Germany. (Accepted presentation.)

Gibbon, Dafydd, Firmin Ahoua, Francois Kipré Blé & Sascha Griffiths (In progress). Discrete level narrative, terraced music: insights from underdocumented Ivorian languages. Language Documentation and Language Theory 2 Conference, London, UK. (Accepted presentation.)

Gibbon, Dafydd, Pramod Pandey, Mary Kim Haokip & Jolanta Bachan (2009). Prosodic issues in synthesising Thadou, a Tibeto-Burman tone language. InterSpeech 2009, Brighton, UK.

Gibbon, Dafydd. 2009. Why should linguists compute? Reflections on language documentation and linguistic theory. USEM Journal of Linguistics, Languages and Literature. Volume 2.