The Gesticon:
What it was
What it is
What it should be

Hannes Pirker
OFAI, Vienna
What IS a Gesticon?

- **Repository** of communicative gestures
  - Hand/arm gestures
  - Facial expressions
  - Head & gaze movements
  - Postures
What is a Gesticon good for?

- Application paradigm:
  - ECA systems for automatically generating multimodal behaviour
What is a Gesticon good for?

- Application paradigm:
  - ECA systems for automatically generating multimodal behaviour
  - Systems builds on a set of predefined behaviours/gestures from which it selects appropriate ones

What is a Gesticon good for?

- Application paradigm:
  - ECA systems for automatically generating multimodal behaviour
  - Systems builds on a set of predefined behaviours/ gestures from which it selects appropriate ones
- Appropriate means:
  -
What is a Gesticon good for?

- Application paradigm:
  - ECA systems for automatically generating multimodal behaviour
  - Systems builds on a set of predefined behaviours/gestures from which it selects appropriate ones
- Appropriate means
  - appropriate meaning
What is a Gesticon good for?

- Application paradigm:
  - ECA systems for automatically generating multimodal behaviour
  - Systems builds on a set of predefined behaviours/ gestures from which it selects appropriate ones
- Appropriate means
  - appropriate meaning
  - appropriate temporal + spatial properties
Example: The NECA Architecture

- NECA (Net Environment for Conversational Agents)
- Generating multimodal dialogues between avatars
Example: The NECA Architecture

- NECA (Net Environment for Conversational Agents)
- Generating multimodal dialogues between avatars
- Using
  - Multimodal-Natural Language Generation (M-NLG)
Example: The NECA Architecture

- NECA (Net Environment for Conversational Agents)
- Generating multimodal dialogues between avatars
- Using
  - Multimodal-Natural Language Generation (M-NLG)
  - Speech Synthesis (TTS)
Example: The NECA Architecture

- NECA (Net Environment for Conversational Agents)
- Generating multimodal dialogues between avatars
- Using
  - Multimodal-Natural Language Generation (M-NLG)
  - Speech Synthesis (TTS)
  - Gesture Assignment (GA)
Example: The NECA Architecture

- NECA (Net Environment for Conversational Agents)
- Generating multimodal dialogues between avatars
- Using
  - Multimodal-Natural Language Generation (M-NLG)
  - Speech Synthesis (TTS)
  - Gesture Assignment (GA)
  - Different Player Technologies
Player 1: eShowRoom Scenario
Charamel CharActor
Player 2: Socialite Scenario:
Macromedia Flash
M-NLG: select content related gestures

User Input

Scene Generator

RRL
Scene Description

Affective Reasoner (AR)

modal Natural Language Generator (M-NLG)

RRL
Multi-modal Output

Text/Concept to Speech Synthesis (CTS)

RRL
Phonetic+Prosodic Information

Gesture Assignment Module (GA)

RRL
Animation directives

Player-Specific Rendering

Animation Control Sequence: FLASH / Charamel
M-NLG: select content related gestures

- User Input
  - Scene Generator
  - Scene Description
  - Affective Reasoner (AR)

- Modal Natural Language Generator (M-NLG)
  - Multi-modal Output
  - Text/Concept to Speech Synthesis (CTS)

- Gesture Assignment Module (GA)
  - Phonetic+Prosodic Information
  - Animation directives

- Player-Specific Rendering

- Animation Control Sequence: FLASH / Charamel
M-NLG output

- **Text** annotated with:
  - syntax
  - (information-structure)
  - gesture-info

- **User Input**

- **Text annotated with:**
  - syntax
  - (information-structure)
  - gesture-info

- **Gesture Assignment Module (GA)**

- **Player-Specific Rendering**

- **Animation Control Sequence**: FLASH / Charamel
Speech synthesis output: supplies timing info

User Input

Scene Generator

Scene Description

Affective Reasoner (AR)

Multi-modal Natural Language Generator (M-NLG)

RRL

Scene Description

Text/Concept to Speech Synthesis (CTS)

RRL

Multi-modal Output

Phonetic+Prosodic Information

RRL

Gesture Assignment Module (GA)

RRL

Animation directives

Gesture (candidates) selection based on meaning

Player-Specific Rendering

Animation Control Sequence: FLASH / Charamel
Speech synthesis output: supplies timing info

- **Timing info:**
  - sounds
  - syllables
  - words
  - phrases
  - accent / stress
GestureAssignment
add beats, idle, check constraints...

User Input

Scene Generator
Scene Description

Affective Reasoner (AR)

Multi-modal Natural Language Generator (M-NLG)

RRL
Scene Description

Multi-modal Output

Text/Concept to Speech Synthesis (CTS)

RRL
Multi-modal Output

Phonetic+Prosodic Information

RRL
Animation directives

Gesture Assignment Module (GA)

Player-Specific Rendering

Animation Control Sequence: FLASH / Caramel

gesticon

Gestural candidates) selection based on meaning
GestureAssignment
add beats, idle, check constraints...

User Input

Scene Generator

Scene Description

RRL

Affective Reasoner (AR)

Gestures (candidates) selection based on meaning

Multi-modal Natural Language Generator (M-NLG)

Text/Concept to Speech Synthesis (CTS)

RRL

Multi-modal Output

Phonetic + Prosodic Information

RRL

Animation directives

Constraint checking
Scheduling

Gesture Assignment Module (GA)

Player-Specific Rendering

Animation Control Sequence: FLASH / Charamel

© ÖFAI, Wien
GA output: SMIL-structures

User Input

Scene Generator

Affective Reasoner (AR)

RRL
Scene Description

Multi-modal Natural Language Generator (M-NLG)

RRL
Multi-modal Output

Text/Concept to Speech Synthesis (CTS)

RRL
Phonetic+Prosodic Information

Gesture Assignment Module (GA)

RRL
Animation directives

Player-Specific Rendering

Gesticon

Constraint checking
Scheduling

Gesture (candidates) selection based on meaning
Player-specific rendering

User Input

Scene Generator

Scene Description

RRL

Multi-modal Natural Language Generator (M-NLG)

Multi-modal Output

RRL

Text/Concept to Speech Synthesis (CTS)

Phonetic+Prosodic Information

RRL

Gesture Assignment Module (GA)

Animation directives

Constraint checking

Animation Control Sequence: FLASH / Charamel

Player-Specific Rendering

Gesture (candidates) selection based on meaning

Gesticon
Gesticon?

User Input

Scene Generator

Affective Reasoner (AR)

RRL

Scene Description

Animation Control Sequence: FLASH / Caramel

?

gesticon

?
The Gesticon

- Ideally the **central** inventory for “all” knowledge on gestures
- Encode mapping between **form and meaning**
The Gesticon

• Ideally the central inventory for “all” knowledge on gestures
• Encode mapping between form and meaning
The Gesticon

- Ideally the central inventory for “all” knowledge on gestures
- Encode mapping between form and meaning

NAME

Main Entry: **lexicon**

Pronunciation: ˈleks-ən
Function: noun

Inflected Form(s): plural **lexica** /ˈleksəkə/; or **lexicons**

Etymology: Late Greek *lexikon*, from neuter of *lexikos* of words, from Greek *lexis* word, speech, from *legein* to say -- more at LEGEND

1: a book containing an alphabetical arrangement of the words in a language and their definitions: **dictionary**

2a: the vocabulary of a language, an individual speaker or group of speakers, or a subject b: the total stock of morphemes in a language

3: **repertoire, inventory**
The Gesticon

- Ideally the central inventory for “all” knowledge on gestures
- Encode mapping between form and meaning

NAME

FORM

Main Entry: *lex-i-con* ~
Pronunciation: 'lek-i-kan also -k&n
Function: noun

Inflected Form(s): plural *lex-i-ca* ~ /-k&/; or *lexicons*
Etymology: Late Greek *lexikon*, from neuter of *lexikos* of words, from Greek *lexis* word, speech, from *legein* to say -- more at LEGEND

1: a book containing an alphabetical arrangement of the words in a language and their definitions: DICTIONARY

2 a: the vocabulary of a language, an individual speaker or group of speakers, or a subject b: the total stock of morphemes in a language

3: REPERTOIRE, INVENTORY
The Gesticon

- Ideally the central inventory for “all” knowledge on gestures
- Encode mapping between form and meaning

**NAME**

**FORM**

**MEANING**

- Main Entry: *lexicon*
- Pronunciation: 'lek-sə-"kān also -kən
- Function: noun
- Inflected Form(s): plural *lexica* /-kə/; or *lexicons*
- Etymology: Late Greek *lexikon*, from neuter of *lexikos* of words, from Greek *lexis* word, speech, from *legein* to say -- More at LEGEND

1. a book containing an alphabetical arrangement of the words in a language and their definitions: **dictionary**
2. a: the vocabulary of a language, an individual speaker or group of speakers, or a subject b: the total stock of morphemes in a language

**REPERTOIRE, INVENTORY**
What’s in a name?

- 3 of 3.800 pictures called “hands up” in Google
<gesticonEntry>
  <verbatim/>
  <function/>
  <restrictions/>
  <form/>
  <playercode/>
</gesticonEntry>
<verbatim>
Thinking: Tina: adaptor: moves right hand to chin but in addition left hand moves to shoulder-height + palm up
</verbatim>
<function type="adaptor"
    alignto="sentence"
    aligntype="par"
    meaning="think"/>
Gesticon Entry

<function>

<gesticonEntry>
  <verbatim/>
  <function/>
  <restrictions/>
  <form/>
  <playerCode/>
</gesticonEntry>

<function type=" adaptor" alignto=" sentence" aligntype=" par" meaning=" think"/>
Gesticon Entry

<restriction>

<gesticonEntry>
  <verbatim/>
  <function/>
  <restrictions/>
  <form/>
  <playerCode/>
</gesticonEntry>

Collection of constraints that control applicability of a gesture apart from meaning-attribute

<restriction>
  <and>
    <constraint name="gender" val="female"/>
    <constraint name="speaker" val="tina"/>
    <constraint name="occ_emotion" val="anger"/>
  </and>
</restriction>
• does not provide information for the renderer/player but stores information “from” the renderer
• does not provide information for the renderer/player but stores information “from” the renderer

• Purpose: provide GestureAlignment at least a rough clue on the temporal and spacial properties of a otherwise completely opaque gesture
<form>

<gesticonEntry>
<verbatim/>
<function/>
<restrictions/>
<form/>
<playercode/>
</gesticonEntry>

• does not provide information for the renderer/player but stores information “from” the renderer

• Purpose: provide GestureAlignment at least a rough clue on the temporal and spacial properties of a otherwise completely opaque gesture

• In current terminology this is called METADATA
Gesticon Entry

<form>
  <gesticonEntry>
    <verbatim/>
    <function/>
    <restrictions/>
    <form/>
    <playerCode/>
  </gesticonEntry>
</form>

<form>
  <position>
    <components/>
  </position>
</form>
Gesticon Entry

```xml
<gesticonEntry>
  <verbatim/>
  <function/>
  <restrictions/>
  <form/>
  <player code/>
</gesticonEntry>

<form>
  <position>
    <start left="D(own)O(ut)" right="DO"/>
    <end left="T(op)O(out)" right="TC(enter)"/>
  </position>
</form>
```
<form>
  <components>
    <stroke>
      <dur min="1000" default="1300" max="2000"/>
    </stroke>
    <hold>
      <dur min="500" default="1000" max="50000"/>
    </hold>
  </components>
</form>
<playercode>

Links to external specifications

<gesticonEntry>
  <verbatim/>
  <function/>
  <restrictions/>
  <form/>
  <playercode/>
</gesticonEntry>

<playercode type="flash" id="g_39_2"/>

<playercode type="character"
             length="2600"
             id="tina/character/motions/gs_thinking"/>
<playercode>

Links to external specifications

<gesticonEntry>
    <verbatim/>
    <function/>
    <restrictions/>
    <form/>
    <playercode type="flash" id="g_39_2"/>
</gesticonEntry>

key to an entry in a flash-based repository

<playercode type="character"
    length="2600"
    id="tina/character/motions/gs_thinking"/>

filename for Character key-frame sequence
Step 1: NECA Gesticon

```xml
<gesticonEntry>
  <verbatim/>
  <function/>
  <restrictions/>
  <form/>
  <playercode/>
</gesticonEntry>
```
Step 2: The Universal Gesticon

- Changing the bearing
  - NECA: making existing opaque animations USABLE within one architecture
  - NOW: making representations RE-USABLE for different architectures
Step 2: The Universal Gesticon

- Changing the bearing
  - NECA: making existing opaque animations USABLE within one architecture
  - NOW: making representations RE-USABLE for different architectures
Step 2: The Universal Gesticon

- Changing the bearing
  - Working on generalized/application independent `<function>` soon proved unsatisfactory
Step 2: The Universal Gesticon

- Changing the bearing
  - Working on generalized/application independent <function> soon proved unsatisfactory
  - Turning focus to elaboration of <form> element
Step 2: The Universal Gesticon

- Changing the bearing
  - Working on generalized/application independent <function> soon proved unsatisfactory
  - Turning focus to elaboration of <form> element
  - Aim: provide a format for player independent, re-usable specification of physical appearance of gestures, facial expressions, postures
Step 2: The Universal Gesticon

- Content of `<form>` element changes
  - from METADATA (i.e. minimal tempo-spatial information necessary for e.g. scheduling)
Step 2: The Universal Gesticon

- Content of `<form>` element changes
  - from METADATA (i.e. minimal tempo-spatial information necessary for e.g. scheduling)
  - to a fine grained description of body movements that are ultimately to be used for animation rendering
Current Gesticon Document (D6e)

• Current description covers this later sense of <form> element only

• For arms & hands:
  • Heavily building on MURML (Kranstedt et al. 2004)

• Adding
  • Facial expression
  • Head, gaze
  • Upper-body, hips, legs...
Where do we go from here?

- `<form>` still needs a lot of elaboration
- Evaluation Criteria
  - Expressive Power
  - Conciseness
  - Ease of implementation for players
Where do we go from here?

- Elaborated form feature only make sense iff it is usable -- and actually used by players
- Evaluate chances for this...
Where do we go from here?

• Open topics:
  • What do we really miss from the original Gesticon
    • <function>
Where do we go from here?

- **Open topics:**
  - What do we really miss from the original Gesticon
  - `<function>` --> FML
Where do we go from here?

- Open topics:
  - What do we really miss from the original Gesticon
    - <function>
    - <constraints> --> ???
Where do we go from here?

• Open topics:
  • What do we really miss from the original Gesticon
    • \(<\text{function}\>\)
    • \(<\text{constraints}\>\)
    • \(<\text{form\_as\_METADATA}\> + <\text{playercode}\>\)
Where do we go from here?

• Open topics:
  • What do we really miss from the original Gesticon
    • <function>
    • <constraints>
    • <form_as_METADATEA> + <playercode> --> ???
Where do we go from here?

• Open topics:
  • What do we really miss from the original Gesticon
    • <function>
    • <constraints>
    • <form_as_METADATA> + <playercode> --> ???
  • --> minimal interface for applications where players with limited flexibility are used ???
Where do we go from here?

Open topics:
- Is it useful to keep all the information on a gesture in one place?
- Logically there is no problem to distribute to `<form>` and meaning-to-form mapping in different files?
Where do we go from here?

• Keep swinging ...