

A frame semantics for CHAMELEON

Prof. Paul Mc Kevitt

Faculty of Informatics
University of Ulster
Derry/Londonderry
NORTHERN IRELAND

E-mail: p.mckevitt@ulst.ac.uk

CHAMELEON architecture

- Speech recognizer
- Speech synthesizer
- Gesture recognizer
- Image recognizer
- Laser
- Microphone array
- Domain model
- Dialogue/NLP module
- Topsy
- Blackboard

IntelliMedia TourGuide

- inputs (speech, images, microphone array)
- outputs (speech, laser)
- gives guidance on layout/usage of building
- 2D architectural plans (or 3D model)
- laser points + draws routes
- link to internet/WWW?

Semantic representations

- mapping from/to inputs/outputs
- integration
- communication
- varying synchronization

Sample dialogue

USER: Point to Hanne's office.

CHAMELEON: [points]

This is Hanne's office.

USER: Where is Jorgen's office?

CHAMELEON: [points]

This is Jorgen's office.

USER: Show me Paul's office.

CHAMELEON: [points(twice)]

This is Paul Dalsgaard's office and
this is Paul Mc Kevitt's office.

or/

CHAMELEON: There are two Paul's.

Do you mean Paul Dalsgaard or Paul Mc Kevitt?

USER: Paul Dalsgaard!

CHAMELEON: [points]

This is Paul Dalsgaard's office.

USER: Who is in the office beside him?

CHAMELEON: [points]

Boerge, Jorgen and Hanne's offices
are beside Paul Dalsgaard's office.

USER: [points]

Whose office is this?

CHAMELEON: [points]

This is Ipke's office.

USER: No, that's Tom's office!

CHAMELEON: [points]

I've updated Ipke's office to Tom's office.

or/

CHAMELEON: [points]

Ipke and Tom are in the same office!

USER: Show me the route from Tom's office
to Boerge's office.

CHAMELEON: [draws route]

This is the route from Tom's office
to Boerge's office.

Phenomena

- instructions!
- questions?
- statements/declaratives
- ambiguity
- active dialogue (questions?)
- ellipsis
- reference (endophoric/ pronominal,
exophoric/ demonstrative)
- spatial relationships (beside)
- learning
- routes

Keys

- reference (exophoric)
(“Whose office is this?”)
- spatial relations
(“Who’s in the office beside him?”)

Input frames (F-in)

[MODULE

INPUT: input

INTENTION: intention-type

TIME: timestamp]

[SPEECH-RECOGNIZER

UTTERANCE: (Point to Hanne's office)

INTENTION: instruction!

TIME: timestamp]

[GESTURE

GESTURE: coordinates (3 2)

INTENTION: pointing

TIME: timestamp]

Output frames (F-out)

[MODULE
INTENTION: intention-type
OUTPUT: output
TIME: timestamp]

[SPEECH-SYNTHESIZER
INTENTION: declarative
UTTERANCE: (This is Hanne's office)
TIME: timestamp]

[LASER
INTENTION: description (pointing)
GESTURE: coordinates (3 2)
TIME: timestamp]

Integration frames (F-int)

[MODULE

INTENTION: intention-type

LOCATION: location

OUTPUT: output

TIME: timestamp]

[DIALOGUE/NLP

INTENTION: declarative (pointing)

LOCATION: office (tenant Hanne) (coordinates (5 2))

UTTERANCE: (This is Hanne's office)

TIME: timestamp]

Reference and spatial relationships

[MODULE

INTENTION: intention-type

LOCATION: location

LOCATION: location

LOCATION: location

SPACE-RELATION: beside

REFERENT: person

LOCATION: location

TIME: timestamp]

[DOMAIN-MODEL

INTENTION: query? (who)

LOCATION: office (tenant Hanne) (coordinates (5 2))

LOCATION: office (tenant Jorgen) (coordinates (4 2))

LOCATION: office (tenant Boerge) (coordinates (3 1))

SPACE-RELATION: beside

REFERENT: (person Paul)

LOCATION: office (tenant Paul) (coordinates (4 1))

TIME: timestamp]

Worked example: “Point to Hanne’s office” (giving-an-instruction!)

USER(U-in)(1): “Point to Hanne’s office”

PHENOMENON(1): GIVING AN INSTRUCTION

PROCESSING(1):

SPEECH-RECOGNIZER:

- (1) wakes up when it detects registering of U-in
- (2) maps U-in into F-in
- (3) places and registers F-in on blackboard:

FRAME(F-in)(1):

[SPEECH-RECOGNIZER

UTTERANCE: (Point to Hanne’s office)

INTENTION: instruction!

TIME: timestamp]

PROCESSING(2):

DIALOGUE/NLP:

- (1) wakes up when it detects registering of F-in
- (2) maps F-in into F-int
- (3) places and registers F-int on blackboard:

FRAME(F-int)(1):

[DIALOGUE/NLP

INTENTION: instruction! (pointing)

LOCATION: office (tenant Hanne) (coordinates (X Y))

TIME: timestamp]

PROCESSING(3):

DOMAIN-MODEL:

(1) wakes up when it detects registering of F-int

(2) reads F-int and sees its from DIALOGUE/NLP

(3) produces updated F-int (coordinates)

(4) places and registers updated F-int on blackboard:

FRAME(F-int)(2):

[DOMAIN-MODEL

INTENTION: instruction! (pointing)

LOCATION: office (tenant Hanne) (coordinates (5 2))

TIME: timestamp]

PROCESSING(4):

DIALOGUE/NLP:

(1) wakes up when it detects registering of F-int

(2) reads F-int and sees it's from DOMAIN-MODEL

(3) produces updated F-int (intention + utterance)

(4) places and registers updated F-int on blackboard:

FRAME(F-int)(3):

[DIALOGUE/NLP

INTENTION: declarative (pointing)

LOCATION: office (tenant Hanne) (coordinates (5 2))

UTTERANCE: (This is Hanne's office)

TIME: timestamp]

PROCESSING(5):

LASER:

(1) wakes up when it detects registering of F-int

(2) reads F-int and sees it's from DOMAIN-MODEL

(3) produces F-out (pruning + registering)

(4) places and registers F-out on blackboard:

FRAME(F-out)(1):

[LASER

INTENTION: description (pointing)

GESTURE: coordinates (5 2)

TIME: timestamp]

PROCESSING(6):

SPEECH-SYNTHESIZER:

(1) wakes up when it detects registering of F-int

(2) reads F-int and sees it's from DIALOGUE/NLP

(3) produces F-out (pruning + registering)

(4) places and registers F-out on blackboard:

FRAME(F-out)(2):
[SPEECH-SYNTHESIZER
INTENTION: declarative
UTTERANCE: (This is Hanne's office)
TIME: timestamp]

PROCESSING(7):

TOPSY:

- (1) wakes up when it detects registering of F-out(1) and F-out(2)
- (2) reads F-out(1) and F-out(2) and sees they are from
LASER and SPEECH-SYNTHESIZER
- (3) dials and fires LASER and SPEECH-SYNTHESIZER
in a rhythmic way (synchronized)
 - (1) LASER (reads its own F-out(1) and) fires G-out(1)
 - (2) SPEECH-SYNTHESIZER (reads its own F-out(2) and)
fires U-out(1)

CHAMELEON(G-out)(1): [points]

CHAMELEON(U-out)(1): This is Hanne's office.

Worked example:
“Whose office is this?”
(Exophoric reference)

USER(G-in)(1): [points]

USER(U-in)(6): Whose office is this?

PHENOMENON(2): REFERENCE (EXOPHORIC; demonstrative (this))

PROCESSING(35):

SPEECH-RECOGNIZER:

- (1) wakes up when it detects registering of U-in
- (2) maps U-in into F-in
- (3) places and registers F-in on blackboard:

FRAME(F-in)(6):

[SPEECH-RECOGNIZER

 UTTERANCE: (Whose office is this ?)

 INTENTION: query?

 TIME: timestamp]

PROCESSING(36):

DIALOGUE/NLP:

- (1) wakes up when it detects registering of F-in
- (2) maps F-in into F-int
- (3) places and registers F-int on blackboard:

FRAME(F-int)(16):

[DIALOGUE/NLP

INTENTION: query? (who)

LOCATION: office (tenant Person) (coordinates (X Y))

REFERENT: this

TIME: timestamp]

PROCESSING(37):

DOMAIN-MODEL:

- (1) wakes up when it detects registering of F-int
- (2) reads F-int and sees its from DIALOGUE/NLP
- (3) cannot update F-int as doesn't have a name or coordinates
- (4) goes back to sleep

PROCESSING(38):

GESTURE:

- (1) wakes up when it detects registering of G-in
- (2) maps G-in into F-in
- (3) places and registers F-in on blackboard:

FRAME(F-in)(7):

[GESTURE

GESTURE: coordinates (3 2)

INTENTION: pointing

TIME: timestamp]

PROCESSING(39):

TOPSY:

- (1) wakes up when it detects registering of F-in(6) and F-in(7)
- (2) reads F-in(6) and F-in(7) and
sees they are from SPEECH-RECOGNIZER and GESTURE
that they have same/close timestamp!
there is a query? (with referent) and pointing
in a rhythmic way (synchronized)
- (3) dials and fires DIALOGUE/NLP to read GESTURE

PROCESSING(40):

DIALOGUE/NLP:

- (1) woken up by TOPSY and told to read F-in
- (2) sees F-in is from GESTURE
- (3) determines referent of “this” to be (coordinates (3 2))
- (4) produces updated F-int (coordinates)
- (5) places and registers updated F-int on blackboard:

FRAME(F-int)(17):

[DIALOGUE/NLP

INTENTION: query? (who)

LOCATION: office (tenant Person) (coordinates (3 2))

REFERENT: this (coordinates (3 2))

TIME: timestamp]

PROCESSING(41):

DOMAIN-MODEL:

- (1) wakes up when it detects registering of F-int
- (2) reads F-int and sees its from DIALOGUE/NLP
- (3) produces updated F-int (tenant)
(the domain model finds two tenants for coordinates (3 2)
and chooses the most recent as default)
- (4) places and registers updated F-int on blackboard:

FRAME(F-int)(18):

[DOMAIN-MODEL

INTENTION: query? (who)

LOCATION: office (tenant Ipke) (coordinates (3 2))

REFERENT: this (coordinates (3 2))

TIME: timestamp]

PROCESSING(42):

DIALOGUE/NLP:

- (1) wakes up when it detects registering of F-int
- (2) reads F-int and sees it's from DOMAIN-MODEL
- (3) produces updated F-int (intention + utterance)
- (4) places and registers updated F-int on blackboard:

FRAME(F-int)(19):

[DIALOGUE/NLP

INTENTION: declarative (who)

LOCATION: office (tenant Ipke) (coordinates (3 2))

REFERENT: this (coordinates (3 2))

UTTERANCE: (This is Ipke's office)

TIME: timestamp]

PROCESSING(43):

LASER:

- (1) wakes up when it detects registering of F-int
- (2) reads F-int and sees it's from DOMAIN-MODEL
- (3) produces F-out (pruning + registering)
- (4) places and registers F-out on blackboard:

FRAME(F-out)(11):

[LASER

INTENTION: description (pointing)

GESTURE: coordinates (3 2)

TIME: timestamp]

PROCESSING(44):

SPEECH-SYNTHESIZER:

- (1) wakes up when it detects registering of F-int
- (2) reads F-int and sees it's from DIALOGUE/NLP
- (3) produces F-out (pruning + registering)
- (4) places and registers F-out on blackboard:

FRAME(F-out)(12):
[SPEECH-SYNTHESIZER
INTENTION: declarative
UTTERANCE: (This is Ipke's office)
TIME: timestamp]

PROCESSING(45):

TOPSY:

- (1) wakes up when it detects registering of F-out(11) and F-out(12)
- (2) reads F-out(11) and F-out(12) and sees they are from
LASER and SPEECH-SYNTHESIZER
- (3) dials and fires LASER and SPEECH-SYNTHESIZER
in a rhythmic way (synchronized)
 - (1) LASER (reads its own F-out(11) and) fires G-out(6)
 - (2) SPEECH-SYNTHESIZER (reads its own F-out(12) and)
fires U-out(6)

CHAMELEON(G-out)(6): [points]

CHAMELEON(U-out)(6): This is Ipke's office.

Conclusion and future work

- adequate expressiveness of semantics
- augmentation of semantics
- comparing/testing
various methods of synchronization
- mobile computing
(head mounted display/ wearable computer)
- multiple speakers/VideoConferencing