

middleware for incremental processing in conversational agents

David Schlangen*, Timo Baumann*, Hendrik Buschmeier**, Okko Buß*, Stefan Kopp**, Gabriel Skantze***, Ramin Yaghoubzadeh**

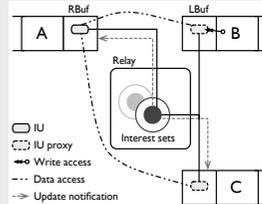
* University of Potsdam, Germany | ** Bielefeld University, Germany | *** KTH, Stockholm, Sweden

Abstract We describe three implementations of ‘middleware’ layers for incremental processing in dialogue systems, which take care of passing around and maintaining incremental information between the system modules. The implementations are all based on the model proposed by (Schlangen & Skantze 2009), but differ in certain important aspects.

IPAACA Implementation based on the ‘D-Bus’ message bus system, which is used for remote procedure calls and bidirectional synchronisation of IUs.

- IUs reside as objects on the D-Bus and are seen as proxies by other modules
- Published IUs can be modified from either side; all interested components are informed automatically
- Update notification and module administration are handled by a dedicated ‘Relay’, where modules register
- Modules provide a list of IU categories and module names they are interested in, thus creating loose functional links or fixed links, respectively
- Due to the wide availability of D-Bus libraries, versions of IPAACA are provided for C++, Python, and Java

Example Data access on the IU proxies is transparently delegated over the D-Bus; module A has published an IU. B and C are registered in the corresponding interest set, thus receiving a proxy of this IU in their left buffer. Whenever B changes the IU, A and C receive update notifications.



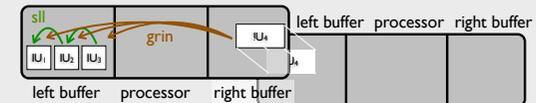
Conclusions The approaches differ along certain dimensions.

- Strength of module coupling: IPAACA couples modules loosely, via brokering system; jindigo enforces tight coupling via share IU-network. InproTK has provisions for both styles.

The IU-Model (S&S’09) Basic notions:

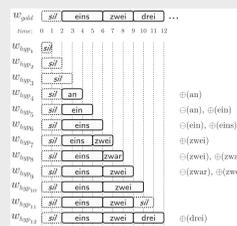
- IU: Incremental Unit, minimal unit of information to be passed around between modules of processing system.
- Module: Consists of left buffer, processor, and right buffer.
- Operations:
 - add: new IUs are posted by one module for next one.

- purge / revoke: IUs that were wrongly hypothesized are ‘taken back’. (E.g., ‘four’ becomes ‘fourty’.)
- commit: module signals that it will not revoke IU.



InproTK Java-based implementation, but with provisions for allowing modules programmed in other languages (via OAA-bindings).

- event-driven left-to-right (bottom-up) processing, via even listeners
- communicates both edit messages (updates) as well as full buffers
- support for concurrent as well as sequential modules
- implemented in Java, integrated with Sphinx-ASR
- no information replication, access via grounded-in links, ‘intelligent’ IUs
- comes with selection of modules, including ASR smoothing to avoid excessive hypothesis editing



Jindigo Java-based Incremental Dialogue Framework www.jindigo.net

- Single Java process, shared memory
- Asynchronous modules
- Standard modules and tools implemented
- Graph-based update model
 - Efficient update messages
 - Supports immutable IUs
 - supports parallel hypotheses

String	Right buffer	Update message
t ₁ : one	(w1) ← one → (w2)	[w1, w2]
t ₂ : one five	(w1) ← one → (w2) ← five → (w3)	[w1, w3]
t ₃ : one	(w1) ← one → (w2) ← five → (w3)	[w1, w2]
t ₄ : one four five	(w1) ← one → (w2) ← five → (w3) ← four → (w4) ← five → (w5)	[w1, w5]
t ₅ : [commit]	(w1) ← one → (w2) ← five → (w3) ← four → (w4) ← five → (w5)	[w5, w5]

References & Acknowledgements

You can find the packages here: <http://purl.org/net/Middlewares-SIGdial2010>
Please download and try them out!

BF acknowledges DFG (via CITEC cluster of excellence).
UP also DFG (via Emmy Noether Programme)