

# Introducing the Fair and Logical Trading Project

David M. Evers<sup>†</sup> with Alan S. Abrahams<sup>\*</sup>, Steven O. Kimbrough<sup>\*</sup>,  
Jean M. Bacon<sup>†</sup>, and Andrew J.I. Jones<sup>◇</sup>

<sup>†</sup> University of Cambridge, UK

<sup>\*</sup> University of Pennsylvania, USA

<sup>◇</sup> King's College London, UK

# Talk outline

- Research motivation
  - Logic-based compositional e-commerce.
- Building blocks
  - Free logic.
  - Speech acts.
  - Disquotation.
  - The Formal Language for Business Communication.
- Our prototype.
- Future directions.

# Goals of the F&LT project

- There are two broad F&LT project goals:
  - Design, implement, and field light-weight business application software for inter-organisational transactions.
  - Develop and implement a fundamental and comprehensive theory of business-related communication.
- We encourage the participation of other researchers.
  - There is no shortage of interesting problems!

# Light-weight software (I)

- By 'light-weight' we contrast with existing Electronic Document Interchange (EDI).
  - EDI Messages belong to certain *transaction sets* within standards such as X12 and EDIFACT.
  - The EDI messages are fairly high-level: e.g. a purchase order (X12 transaction set 850).
  - As a consequence there are a large number of potential message types.
- EDI documents exchange information, but do not specify the semantics of interaction.
  - Often the 'trade scenario' expected will be hard-coded into custom application software.

# Light-weight software (II)

- We are keen to increase the accessibility of e-commerce infrastructure.
  - The need to engineer specific types of interaction reduces the ease of conducting open commerce.
  - Addressing access to e-commerce will help achieve what the U.N. term *trade facilitation*.
- Need to tackle the so-called “first trade problem”.
- Reduce the potentially very high cost of the first message interchange by developing a more compositional representation.

# Light-weight software (III)

- The benefits of EDI can (arguably) be split into two main areas:
  - Supporting communications with consistent schemas.
  - Allowing transactions to proceed at machine speeds.
- Naturally EDI messages cannot possibly cover all possible needs, however.
  - Often EDI 'notes' fields are used to convey information beyond the specified structure of a given message.

# Light-weight software (IV)

- Small to medium enterprises (SMEs) will often not require machine speed transactions.
  - We can thus simplify our infrastructure by requiring parts of its operation to be human-driven.
  - The benefit to SMEs of some of the communication being expressed in a formal language remains.
- We propose to embed items of formal dialogue within informal communications.
  - We expect that the larger the organisation, the greater the formal dialogue coverage will be.
- We hope to support the 'fair trade' movement.

# Fair trade aggregators

- Fairtrade. <http://www.fairtrade.org.uk/>
  - The Fairtrade Foundation (UK) aims to protect developing-world farmers from commodity price fluctuations.
- Silverchilli. <http://www.silverchilli.com/>
  - Silverchilli provides an e-commerce front for developing world jewellery producers.
- Global Exchange. <http://www.globalexchange.org/>
  - Global exchange (US) is a human rights organisation which also supports fair trade.

# Formal business comms (I)

- Our other main goal is to work towards developing a theory of business-related communication.
  - Many formalisms exist for purchase orders, invoices, receiving reports, and so on.
- Existing commercial standards do not focus on the underlying logical and semantic structures of their meanings, however.
  - Integration with an organisation's existing information processing systems is generally done in an ad hoc manner.
  - This is known as the *EDI mapping problem*.

# Formal business comms (II)

- A formalisation in itself does not limit what can be inferred.
- Consider a well-formed EDI purchase order message.
  - The deontic state of both parties changes: each now has new obligations and rights.
  - This understanding is independent of the EDI message.
- Our aim is to work towards adding a formal encoding of at least part of the interchange agreement between the parties.

# Formal business comms (III)

- We gain more semantically rich communication through the use of message-building terms and predicates.
  - This contrasts with atomic EDI messages.
- In our initial work we include a trusted third party in negotiations.
  - The trusted third party provides an authoritative interpretation of the formal inferences permissible.

# F&LT building blocks (I)

- Reasoning about speech acts requires resilience against the possible non-existence of terms.
  - Standard first order predicate logic (FOPL) does not provide this resilience.
- In non-free logic for some predicate  $F$  we have that  $F(a) \rightarrow \exists x : F(x)$ .
  - Free Logic requires an explicit assertion  $E!(a)$ .
  - We can thus cope with variables ranging over non-existents.

# F&LT building blocks (II)

- Monotonic reasoning will not explain the state of affairs within business interactions.
  - E.g. a party discovers new information which implies a breach of a contractual clause they had previously believed was satisfied.
- Statements made by any party at any time are relevant merely through their existence.
  - Capture this by recording speech acts.
  - What we can infer from them is a separate matter.
  - They are intensional, thus assertions may be false, promises may be broken, etc.

# F&LT building blocks (III)

- The speech acts will often contain embedded propositional content.
  - E.g. “I assert that I have met all the conditions of our contractual obligations”
- Clearly a great deal of further state is required to determine aspects such as:
  - Who are the parties that “I” and “our” refer to?
  - What were the contractual obligations?
  - What evidence is linked to this statement?

# F&LT building blocks (IV)

- Kimbrough's Disquotation Theory allows us to pack and unpack embedded propositions.
- Kimbrough has shown disquotation to be sound and complete.
  - ... and much better him than me!
- Say  $\phi$  represents the predicate "I have met all the conditions of our contractual obligations".
- $\phi$  is the object of "I assert that ...".
  - Packaged form is  $[\phi]$ .

# F&LT building blocks (V)

- Our example statement becomes:  
 $\exists e(\text{assert}(e) \wedge \text{Subject}(e, I) \wedge \text{Object}(e, [\phi]))$
- Expanding the meaning of  $[\phi]$  is done through an axiom schema such as:

$$\forall e((\text{assert}(e) \wedge \text{Object}(e, [\phi])) \rightarrow (\text{Veridical}(e) \leftrightarrow \phi))$$

- Note that it is this axiom schema which is actually effecting disquotation.

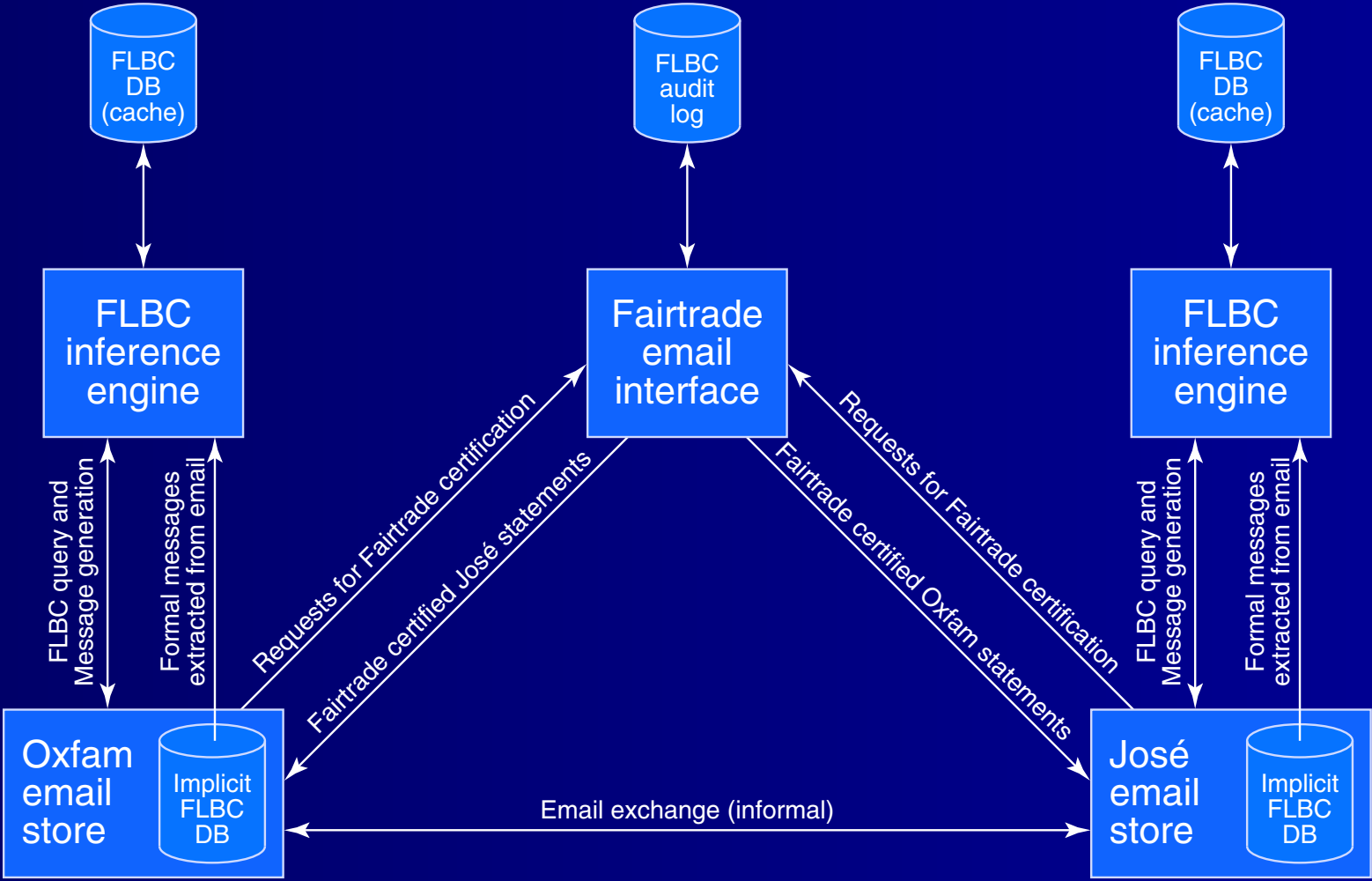
# F&LT building blocks (VI)

- Kimbrough's Formal Language for Business Communication will provide the terms over which speech acts will be encoded:
  - Assert
  - Command
  - Request
  - Commit
  - Declare
- Jones' work on signalling conventions fits well with FLBC expansions of speech acts.

# F&LT building blocks (VII)

- It is also necessary that a governing convention is defined to establish that parties will understand particular messages mean what they say.
  - Avoiding relying on a notion of 'intention' is crucial when involving electronic agents in such FLBC communications.
- We do not plan to investigate complete protocol systems or attempt to encode complete systems of law in our initial prototype, however!

# F&LT architecture



# F&LT scenario (I)

- Our aim is to shift towards more direct producer / retailer communication, with the facilitator acting as a trusted third party in negotiations.
  - Incrementally formalise the producers' communications.
- Say we have a producer José, a retailer Oxfam and Fairtrade as facilitator.

# F&LT scenario (II)

- We assume that Fairtrade aggregates the types of products its producers can make into a catalogue to avoid the difficulty of product specification.
- Oxfam can indicate to Fairtrade the hypothetical messages it wants to receive.
  - E.g. “I, X, will produce for you, Oxfam, 100 units of catalogue item 324 for \$10 per unit by date Z”.

# F&LT scenario (III)

- Fairtrade can then determine how to contact its producers to effect this hypothetical state of affairs.
- Producers will make their own statements, e.g. promises of production, but importantly will assert their belief in the coverage of Oxfam's hypothetical statement by their specific ones.
- Our envisaged interaction follows.

# F&LT scenario (IV)

1. Oxfam indicates that it is intending to do business on the terms and conditions published by the Fairtrade facilitator.
2. Oxfam indicates the goal it wishes to achieve with respect to the delivery of a particular shipment of produce.
3. Fairtrade deduces the set of producers it needs to contact in order to achieve Oxfam's objectives with reasonable certainty.

# F&LT scenario (V)

4. The selected producers make an offer to Oxfam using Fairtrade as a trusted third party.
5. Oxfam enters into an agreement with a suitable producer.
  - Fairtrade keeps a log of the speech acts registered by both parties.
    - Through speech act logs, Oxfam and José should be able to determine their outstanding obligations at any time.

# F&LT prototype (I)

- We have developed an initial proof of concept prototype F&LT software service.
  - It provides only basic information interchange.
  - So far only simple validity checks have been coded.
- Implemented in SWI-Prolog.
  - Our representation is very RDF-like.
  - We do not strictly use 3-tuples however.
  - Prolog allows us to include computational semantics directly into our representation.

# Initiating a purchase order

```
1 mdb(b:e1, nType, po) .
2 mdb(b:e1, speaker, buyer) .
3 mdb(b:e1, addressee, seller) .
4 mdb(b:e1, cul, b:t1) .
5 mdb(b:t1, value, Now) .
6 mdb(b:e1, content, s, b:e2) .
7 mdb(b:e1, content, b, b:e3) .
```

- Note the items of embedded propositional content  $b:e3$  and  $b:e2$ .
- FLBC terms above are coloured red to emphasise they will be within the interchange agreement.

# Embedded propositional content

```
8 mdb(b:e2, nType, deliver).
9 mdb(b:e2, agent, seller).
10 mdb(b:e2, benefactive, buyer).
11 mdb(b:e2, cul, b:t2).
12 mdb(b:e2, require, b:e2-r1) :-
13     mdb(b:t1, value, T1),
14     mdb(b:t2, value, T2),
15     T2 =< T1 + 30.
```

- Note the two types of variables in `b:e2-r1`
  - FLBC variables such as `b:t1`.
  - Prolog variables such as `T1`.

# Inter-party communication (I)

- Our current prototype passes Prolog terms over TCP/IP between parties.
  - The transitive dependencies due to disquotation are automatically sent too.
- All parties maintain their own database of FLBC events and messages.
  - An event may be deemed to “count as” another event, but it is best to also maintain the original separate events.
  - This allows for much richer historical queries.

# Inter-party communication (II)

- Our system does not currently include a trusted third party (TTP).
  - A simplistic TTP implementation would simply repeat the inferences being asserted by either party.
  - More complex TTPs might provide useful services relating to the interchange agreement in effect.
- Our communication layer is actually a simple publish/subscribe system.
  - We expect this to well suit group communication.
  - At human-speed, email with FLBC annotations or attachments may be sufficient though!

# Future directions (I)

- There are many aspects of future work.
  - Extension of FLBC coding, and our communication infrastructure.
- Employing the Event Calculus.
  - Initial experimentation shows the EC to add a powerful framework for analysing the changes in deontic state of interacting parties.
- An RDF ontology
  - Our RDF-like structure was previously noted.
  - Creating an RDF ontology might be generally useful even if not used with our exact FLBC semantics.

# Future directions (II)

- Development environment
  - A management environment is being constructed for non-technologist participation in FLBC message exchanges.
- Hypothetical action simulations
  - Because each party stores their own FLBC state, simulation of hypothetical actions can be undertaken.
  - This is particularly relevant with respect to what further obligations, prohibitions, etc may come into effect when analysing a combination of actions.

# Conclusion

- We have introduced the F&LT project.
  - There is much work left to do (and, admittedly, probably always will be).
  - All suggestions and help greatly appreciated!
- Any questions?

# Stages of formalisation

- We transform from informal to formal communication in a number of stages:
  - Natural language description. As we have presented so far.
  - Semi-formal representations. Tag speech acts and make clear where terms are understood to come from.
  - Formal representation. A Free Logic form with scoped terms and predicates.
  - Logic programming. A computer-executable form of the above.

# Semi-formal representation

[o] <REQUESTS> [f] to join a dialogue [D1] described by [o:D1]. [o] <CLAIMS> [o:D1] participants are [o] and [f], and uses term-set [f], then [FLBCv3].

- Constructs in red are speech acts.
- Yellow items are terms.
  - The [o:D1] style notation is intended to be equivalent to shorthand used in RDF. In this case the name 'D1' is scoped within 'o's name-space.

# Formal representation

$\text{assert}(o:e1) \wedge \text{Speaker}(o:e1, o) \wedge \text{Addressee}(o:e1, f) \wedge$   
 $\text{Cul}(o:e1, \text{now}) \wedge \text{Content}(o:e1, \lceil \text{dialog}(o:D1) \wedge$   
 $\text{participates-in}(o:D1, [o, j]) \wedge \text{Hold}(o:D1, \text{now}) \wedge$   
 $\text{uses-term-set}(o:D1, [f, \text{FLBCv3}]) \rceil)$

$\wedge$

$\text{request}(o:e2) \wedge \text{Speaker}(o:e2, o) \wedge \text{Addressee}(o:e2, f) \wedge$   
 $\text{Cul}(o:e2, \text{now}) \wedge \text{Content}(o:e2, \lceil \text{join}(e3) \wedge \text{Agent}(e3, f) \wedge$   
 $\text{Theme}(e3, o:D1) \wedge \text{Cul}(e3, \text{now}) \rceil)$

# Formal representation syntax

- Standard predicates are:
  - **Speaker**(o:e1, o)
  - **Addressee**(o:e1, f)
  - **Cul**(o:e1, now)
  - **Content**(o:e1, [ **disquoted content** ] )
  - **Hold**(o:D1, now)
  - **Agent**(e3, f)
  - **Theme**(e3, o:D1)

# Formal representation syntax (I)

- Higher level predicates:
  - assert
  - dialog
  - participates-in
  - uses-term-set
  - request
- We plan to use URIs in the style of the semantic-web to facilitate decentralised predicate name creation and sharing.