Brian Sentance

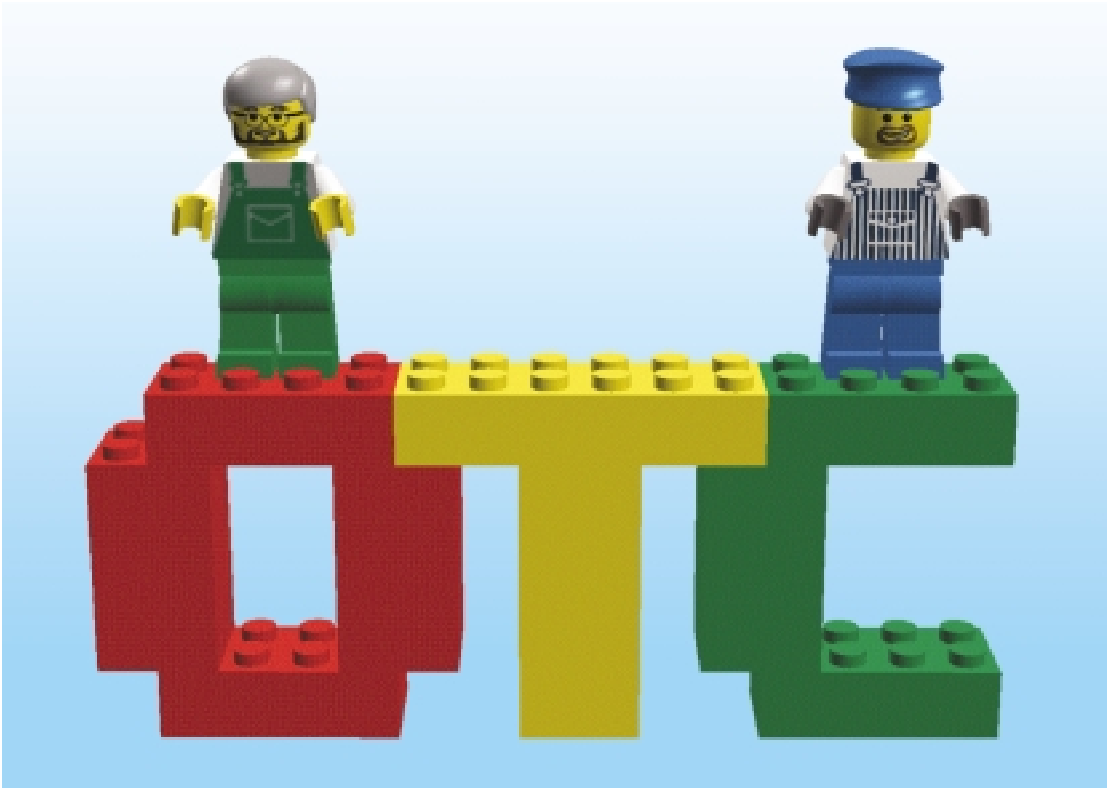
FpML — The Building Blocks for Trade Automation?

What can plastic playthings from Denmark contribute to OTC derivative processing?

Whilst not as mathematically challenging as writing a new pricing model, the world of OTC derivative processing is gathering momentum given the enormous amounts of money that can be saved through trade automation, and taking manual effort and human error out of the process. However, as much as technological standards such as the Financial products Markup Language (FpML) provide hope of increased automation, the reality is different with market players modifying standards, and poor uptake by the buy side. In answering the question about what happens when the pricing model is done and trading begins, then maybe OTC derivative processing still has a lot to learn, maybe even from one of my favorite childhood toys, Legos.

A (slight) digression on building blocks

Going back almost 40 years now, as a young child I adored playing with Lego® bricks, spending many hours building houses, cars, planes, rockets, and anything else that caught my imagination. Fast-forwarding to the present, the advent of having children of my own has given me the legitimate excuse *excuse* being the operative word according to my wife!) to reacquaint myself with Lego® and how it has attempted to keep up with the competitive and changing world of computer gaming and the Internet.



Making OTC Customization Look Far Too Easy

So what has Lego® got to do with OTC derivatives? Not a lot directly (now there's a surprise...). I guess one tenuous analogy is that the range of products has expanded (become more exotic?) as the more standard (vanilla?) bricks have faced stiffer competition from other toys. Not a particularly great analogy as I said, but a reasonably valid one for the development of any product market.

No, the real analogy I think comes in when you look at what Lego® has done with "custom manufacturing" and the use of wiki/online community ideas to generate more product and business. After buying the first of several sets of Legos for my son and daughter (ok, me...), I became frustrated that I didn't have all the bricks that we needed to build and extend the basic models you could build. Upon visiting www.lego.com, I found that not only could I buy standard "extension packs" of bricks, but I could now browse every individual brick of every type and color in the whole range, decide how many of each I wanted, and have this priced, boxed, and delivered to my door within a couple of days. This is good (from a Lego® geek point of view, it is brilliant), but in another part of their site they go a few stages further.

The Lego® Web site has the version 2.0 of the "Lego Digital Designer" (http://ldd.lego.com/ download/), a downloadable application that allows you to design the toy of your choice (think CAD for kids) then have all the bricks necessary packaged into your own box set, complete with a picture of your model on the box lid and delivered to your door. Additionally, you can design your toy from scratch or you can start based on a number of template models that Lego® supplies. Going further than that, your newly designed model becomes a box set that other people can view on the Lego.com Web site, and if they like it they too can purchase it and have it delivered to their home too. Customers get exactly the product they want, and Lego® gets more sales of its bricks plus generates more "box set" products without lifting a finger—pure "Wikinomics" in its best form, combined with a great lesson in custom manufacturing!

So why can't we take the above approach and, with a few commercial modifications, apply a similar approach to OTC derivatives processing? No, I don't mean let's build a Lego® set for processing of derivative trades. I really mean let's look at how custom manufacturing could help in derivatives processing, whether a "wiki" approach might get more people involved in standards and their development. I guess a key to the success of the Lego® site is that it has a standard component, the "brick," which fits together with other bricks and is understandable (visually) to other potential buyers of the resultant "toy." While this standard is proprietary, it is mature (the first bricks were created in about 1947, I think) and in this way is well understood by its users and is very robust.

Building blocks for OTCs?

Who is providing the "bricks" for processing of OTCs? Taking the industry picture as a whole, it does not make for simple viewing, with a range of organizations and companies providing various services to the market, often aligned upon asset class. So the focus I have chosen is to look at the base component, the data, and how this is being standardized.

The FpML (Financial products Markup Language) Organization (www.fpml.org) has as it is mission *"to streamline the process supporting trading activities in the financial derivatives domain through the creation, maintenance and promotion of an e-business language for describing these products and associated business interactions based on industry standards."* Put a different (simpler) way, it aims to be *the* standard language for the electronic processing of derivatives. It is probably worth emphasizing here that FpML is not an application, transport, or network for electronic trading, but rather an XML-based language for communication between the systems involved in the processing of derivatives.

> I don't mean let's build a Lego® set for processing of derivative trades. I really mean let's look at how custom manufacturing could help in derivatives processing

History of FpML

Following the growth and success of the interest-rate swaps market in the 1980s and 1990s, FpML was an initiative started as a research project by JPMorgan in 1997. In 1999, JPMorgan, by then in conjunction with PricewaterhouseCoopers, was able to put forward a draft standard for the processing of interest-rate swaps. In 2001, it was agreed that FpML development should be taken forward through the International Swaps and Derivatives Association (ISDA). Since then, the standard has iterated through various versions, adding more products and functionality, now with version 4.2 at the "recommended" stage, with version 5.0 of FpML at the "working draft" stage.

So what products does FpML cover?

Despite its initial focus on interest-rate swaps, the product range contained within FpML has expanded to include the data necessary to describe the following asset classes:

- **Interest Rate:** Interest Swaps, Swaptions, FRAs, Caps and Floors, Inflation Swaps, and Bullet Payments.
- **FX:** Foreign Exchange Swaps, Spots, Forwards, and FX Options.
- **Credit:** Single-name Credit Default Swaps, Loan Credit Default Swap, CDS on Mortgages, Credit Default Swap Indices, Options, and Baskets.
- **Equity:** Equity Swaps, Equity Options, Variance Swaps, Correlation Swaps, Dividend Swaps, and Total Return Swaps.
- **Bond:** Bond Options and Convertible Bond Options.

In addition:

- **Strategies:** Combinations of the above, for example, to support option "straddles."
- **Underlyings:** Bond, CB, Cash, Deposit, FX, Rate, Equity, Exchange-Traded Funds, Future, Index, Mutual Funds, FRA, Swap, CDS.

Of note is that there is some inconsistency in the representation of data across different asset classes (due in part to the way FpML has evolved product-wise), but resolving this is a focus of the latest draft, FpML 5.0. Focusing on one particular asset class, equity derivatives, then the short description above does the FpML standard a disservice, in that the "equity options" category covers the standard American or European put-and-call options, since version 4.0 has also covered, Bermudans, Basket Underlyings, Forward Starts, Quantos and Composites, Averaging, Barriers, and Binary Options. Not the world's most exhaustive list, but certainly better than I expected since last time I checked, and I would suggest it is much more extensive than the list that many "multiasset" systems claim to support.

What else does FpML do?

FpML is not just about defining the data needed to describe a derivative product. From their "vision" statement, FpML.org state that *"FpML will facilitate OTC business communication: between companies participating in a transaction; within a company seeking to integrate information across systems; and between a company participating in a transaction and a firm offering a service related to that transaction, e.g. confirmation matching or valuation."* Further to this, they add *"FpML should also be able to be used for a variety of business processes, including: structuring and negotiating the terms of a transaction; executing and confirming the transaction; communicating settlement details about the transaction."* So trying to break down an FpML "document" into its constituent parts, it seems to go something like this:

- **Message**—describing what action/information the document is to convey, including:
- **Pre-trade**—quote requests and acceptance messages
- **Trade Execution/Confirmation**—trade affirmation, confirmation, and matching
- **Post-trade**—mainly trade lifetime administration, through to termination
- **Reporting**—valuation and position messages (more of which later)
- **Parties**—identification of the parties to the trade
- **Trade**—the transaction identification, plus product and additional information
- **Portfolio**—a structure allowing collections of trades to be grouped

In addition to the above, there are additional document "building blocks" such as currency and location information, date and business day convention information, and payment-related information.

Validation

Sitting above all the information contained in an FpML document is the "validation framework." This sits at a higher level than checking whether the document is syntactically valid XML, or that the XML conforms to the FpML schema. Rather, it also validates the business meaning, the semantics, of the information in the document. For instance, it validates dates (termination date before transaction date anyone?), structural constraints (can't have stub payment period element without specifying how the payment is calculated), data value–based constraints (can't have "business day following" convention value without specifying where the "business days" are taken from) and some other additional special cases. Validation is an important topic in reducing errors and increasing automation, but has not always received the attention it probably deserves, as anyone who has had to provide technical support for pricing analytics libraries will testify to!

Given FpML's stated aim to support extensibility, this is not quite reflected in its reputation in the market, and maybe more (end-user-focused) tools could be developed to automated/standardize these extensions

Customizing and extending FpML

Given the evolution of new products in the OTC markets, then, how easy is it to extend FpML? FpML.org says that extensibility is one of the main features of FpML, and there seems to be two main techniques for doing this. The first is "wrapping," in which the FpML document is "wrapped" inside another, institution-specific one that contains the information to be added. Conceptually, it is quite simple to understand and has been used by a number of brokers to extend FpML. It does, however, have its difficulties—for example, it does not support the restriction of content within the FpML document (since the FpML is left unchanged).

The second family of techniques is primarily derived from extending FpML by building upon the existing types within FpML while providing a clear identification of the fact that these are institution-specific extensions. For example, you take an existing product such as a "Forward" as a base type and then add your extensions to it within the FpML document. This is the recommended approach going forward for extending FpML and has a number of advantages over the wrapping technique, one of the fundamental ones being that it is closely linked with the original FpML. Given FpML's stated aim to support extensibility, this is not quite reflected in its reputation in the market, and maybe more (end-user-focused) tools could be developed to automated/standardize these extensions. To go back to the Lego® analogy for just a moment, then maybe the building blocks are clear to some (technologists using FpML) but not easy to use for others (the end-users who know the product detail).

Pricing and risk — The working group for quants?

FpML.org has a number of working groups developing the standard. There are product working groups covering the main asset classes: Credit, Commodities, Equity, Interest Rate, Loan, and Repos. There are the technical groups concerned with: Architecture, Business Process, FIX, and Validation. Finally, there is the "Pricing and Risk" working group. This group has the remit to "create a cross-product framework for supporting pricing/valuation and market and credit risk reporting." From what you can see from the pricing and risk examples (take a look at the diagram on http://www.fpml.org/about/ataglance.html—

click on the examples section), then this group seems to be mainly about how valuation and sensitivity reports can be requested between two institutions, which is a very valid and useful scope in itself. For the moment (until later drafts), it stays away from the parameters describing the pricing techniques (Monte Carlo, trinomial, etc.) and the input data to support these techniques, but this is an obvious area where input from quants using and developing pricing models would be useful and productive for the industry.

Some final thoughts

My aim above has been to provide an accessible summary for those who know little about FpML—so please excuse the Lego® digression at the start, and also note that accessibility drove my desire to not include a single piece of FpML in this article (apologies to those of you who really enjoy reading tagged text). FpML and OTC processing are both increasing in importance as financial markets become more and more automated, with cost and profitability (and operational risk!) incentives driving the front office to become more integrated with middle and back office.

A key area for me in terms of usage of a standard is to make it understandable to the end consumers of the data—another way of saying that in my view data transparency (to both IT and end-users) is an important requirement and one that is vital to promote increased productivity (and adoption) in the market. Given its technical origins (in electronic processing, in using XML, etc.), it is probably unsurprising that www.fpml.org comes across as dry and technical (ok, dull if you are a non-techie!). Maybe more could be done here to at least make the product definitions easy to view and understand (i.e., provide the option to view them in non-XML format). Related to this, FpML says it is an open-source project for all, but would it not be good if their standard editor/viewer application for FpML were available free, rather than being rolled up in the cost of an ISDA membership? Come on, let's get the standard out there in the market!

Maybe more could also be done to promote the product definition schema rolled out and used within institutional and vendor systems. For instance, given all the interest in derivatives on the buy side, how many buy-side IT departments are at this moment coming up with different and proprietary database schemas to support OTC derivatives trading? It seems like a missed opportunity to standardize, and to ask why the FpML product definitions are not being used.

I am also puzzled by the seeming lack of involvement in FpML by the pricing model vendors, who do not seem to be officially participating in the development of FpML (at least from what is on the Web site). I know some vendors can "talk" FpML (Numerix, for instance), but might I suggest that some pricing model vendors would regard standardization as commoditization, and as such are not commercially that motivated to? In addition, maybe the validation framework could be "productized" in some way, so that its business logic could be used within both vendor and institutional pricing frameworks—so that not very helpful error messages such as "Invalid Input Data" could be replaced by something more meaningful.

To quote the computer science academic Andrew S. Tanenbaum: *"The nice thing about standards is that there are so many to choose from."* My understanding is that even within the "scope" of FpML usage, then many of the tier-one sell-side institutions have implemented their own versions of FpML, with JPMorgan Chase leading the way with one of the most successful variants. So much for standardization! Beyond FpML, then I find it difficult to be totally clear how the various business standards fit together. FpML does having a joint working group concerning itself with FpML/FIX integration. The FIX protocol (see www.fixprotocol.org) has its origins in trade automation for equities and has done a good job of marketing itself while riding the rise in algorithmic trading. Also, the Pricing and Risk working group for FpML says that it wants to use the Market Data Definition Language, MDDL (see www.mddl.org). The problem I have here is understanding where, for example, more complex data like volatility surfaces sit. Certainly, it doesn't look as though MDDL deals with data like this yet, so will FPML? Another example is that one institution I know wants to specify standard market data scenarios across multiple risk management systems, so do we need a "RiskML" or similar? I would hope not, but where should it fit?

There are plenty of other standards around, all with a broad (and reasonable) scope, but as always the devil is in the details. Some of these would include RIXML for analyst research, NewsML for news, XBRL for financial reporting, and ISO15022 for settlement. And now the International Organization for Standardization (ISO) is progressing with "UNIFI," which is the nickname of the ISO 20022-UNIversal Financial Industry message scheme. The approach taken by ISO is that UNIFI does not describe the financial messages themselves, but rather focuses on providing a "recipe" to develop and consolidate existing and new financial messaging standards. How successful UNIFI becomes and how long it will take is difficult to say—certainly the market has struggled with much simpler issues for a long time (for instance, look at how the debate over defining a unique security identifier dragged on).

FpML and OTC processing are both increasing in importance as financial markets become more and more automated

To close, then, I think the FpML is worth a look at, at the very least from an instrument data definition point of view, and that it probably could do more to promote itself, make itself more accessible, and generate increased participation from market practitioners. And before anyone asks, yes, I would consider getting involved with the FpML organization myself (I will get off this soapbox!), particularly in the pricing and risk area. Until then, however, I think I'll join my son in building a castle with Lego® bricks.

W