

# Risk Software product of the year

## MiFi

**Innovative software produced by LexiFi will help the financial industry to assess the value of a formal language for contracts**

**T**he problem of defining contracts is becoming increasingly acute as the number and complexity of instruments grows. And as e-commerce portals bring electronic trading to the over-the-counter markets, the requirement to specify exactly what is being traded and under what conditions becomes more urgent. A standardised language in which to express contracts has been unattainable – until now.

Paris-based LexiFi Technologies has introduced mFi, a formal and exact language for specifying financial contracts. Not only does it enable traders to be more precise in constructing deals, but mFi's rigour means the specification can provide the basis for valuing contracts, as well as automating and managing their processing through the transaction lifecycle.

In the 1980s, the International Swaps and Derivatives Association helped create the swaps market by defining a set of master agreements for OTC deals. Furthermore, every trading, risk management and back-office system must have some method of representing deals. But both industry association standard agreements and systems' representations tend to focus on specific instrument types or asset classes, and mostly describe only vanilla contracts. They must also be customised for every new type of deal.

Ten years ago, Jean-Marc Eber, then a quant at Société Générale, realised there must be a better approach. He believed that all contracts boil down to a limited set of basic components, and that these are combined in different ways to create individual deals. But although the concept is simple, the process of identifying the components and finding a sufficiently rigorous framework for their combination proved a major challenge.

Eber met Simon Peyton Jones, a programming language expert at Microsoft's research laboratories in Cambridge, England, who immediately realised that what Eber was trying to do in finance matched perfectly what computer scientists faced when they wanted to instruct a machine to perform certain tasks. The solution for computers was functional programming languages, which break structures down into basic compo-

**Jean-Marc Eber, LexiFi**



ments, or a library of primitives, which can be combined to create whatever entity is required and to which can be applied operators to perform a variety of processes.

Because they work with primitive combinators and simple operators, such languages are flexible yet mathematically rigorous. Over the years, computer scientists have defined the semantics (combinators) and syntax (operators) of functional languages for a number of specific subject areas, such as hardware circuits and graphics animation.

But Eber found that distilling the financial combinators from the myriad forms of contracts was a substantial task. After several years of research, he finally produced a set of just 15. "So far, these can express all the instruments we have come across," he says. Eber's new language has few operators, and these are highly simplified, such as "and", "give", "acquire" and "time". And even though there are no special operators for things such as puts and calls,

Eber has found in tests on SocGen's exotics portfolio that there were no exotic options or other complex derivatives that the language could not handle.

With the fundamentals of mFi in place, Eber left SocGen in June 1999 to found LexiFi with partners to create a technology platform to implement the language.

The lack of standards for describing instruments has been a major obstacle to the integration of systems, both within institutions for processes such as the straight-through processing of transactions from the front to the back office and for consolidating trading

information for risk management, and between institutions for automating confirmations, clearing and settlement and so on.

The mFi language could make a significant contribution towards achieving such standards. For example, its contract descriptions could provide building blocks for emerging data standards for derivatives and other instruments, such as FpML (Financial products Markup Language). (Like FpML, mFi is based on the XML general data standard.) The mFi language could also provide a basic language in which organisations such as Isda could express their master agreements.

Furthermore, the functional nature of the language means that its description of contracts can become the direct input to pricing models, as well as to risk measurement models and back-office processes.

Although few people have been able to evaluate mFi as yet, several industry experts who have seen presentations are optimistic. Emanuel Derman, managing director of firm-wide risk at Goldman Sachs, says: "In all areas of computer science, there has been a natural evolution towards building high-level languages with a syntax and semantics specifically geared to the domain of the application. Much of the power of the Unix programming environment, as well as its popularity among scientists and engineers, came from the pleasure of working in this framework. [It makes] sense to do the same thing in the financial arena, and some of the tools we built at Goldman have actually been small steps in this direction."

Michael Dempster, director of research at the Judge Institute of Management's centre for financial research at the University of Cambridge, says: "[The mFi] approach will be a critical component of financial product trading, middle-office, back-office and risk management systems of the future." He has experimented with a similar approach to bond contracts in the past.

Eber plans to put mFi in the public domain and focus his company's efforts on creating tools and services to support the language. Dempster believes that "LexiFi's open strategy is the right business model to make this formal approach take off". LexiFi now offers a compiler and a technical environment for implementing mFi. ■

