FIX and Operations Efficiency

March 14 2019

John Greenan, Alignment Systems
Alyona Bulda, Senior VP, Technology
Jim Northey, GTC Co-chair, Americas and ISO TC68 Chair
Opening Questions

- Conference Code to access the polling system
- Do you believe FIX suffers from operational inefficiency?
  - Yes
  - No
- Do you or colleagues from your firm participate in Orchestra working group?
  - Yes
  - No
- Are you looking to invest in Connectivity technology in 2019?
  - Yes
  - No
Operational Use Cases

- Onboarding
- Certification
- Normalization
- Documentation
- Testing
  - Acceptance testing
  - Generation of test datasets
- DevOps
  - Deployment
  - Automation
  - Change Management
Where are the pain points?

- **Sell Side**
  - Multiple flavors of FIX to support from the buy-side
    - Need to normalize
    - Need to certify
    - Need to provision
  - Multiple internal messaging applications
    - Some FIX
    - Some middleware
    - Some bespoke
  - Multiple venue gateways
    - Need to normalize
    - Need to certify
    - Need to provision

- Orchestra has a machine readable specification is designed to address these issues and others, such as algorithmic trading parameters and FIXatdl distribution
How to engage a FIX counterparty without Orchestra
How to engage a FIX counterparty with Orchestra

![Diagram showing the process of engaging a FIX counterparty with Orchestra]
What problems are we trying to solve?

- Fix protocol was loosely specified from the start. Plenty of room for interpretation.
- Specifications are usually documented in human readable documents that are exchanged between counterparties. Humans must interpret the specs and turn them into executables and configurations.
  - Conditionally required fields are explained in text which must be interpreted and converted to code.
- Fix standards tell the universe of possible values.
  - Which values of OrdType and TimeInForce are accepted by my counterparty?
- Workflow is often not well documented.
  - Under what conditions do I get a Session Level Reject, a Business Message Reject, or an Execution Report with ExecType=Rejected?
  - The same message type may have different contents for different scenarios, e.g. Execution Report for order accepted versus an execution.
- In short, the information we have is sparse and not directly actionable.
Why Orchestra?
Remember early Computer Interfaces?

- Having to configure device drivers?
- Setting DIP Switches on interface cards?
- Each interface needed an expert to get operational
  - Terminals
  - Printers
  - Modems
  - Network Interface Cards
- Then the magic of plug and play
- **The goal of Orchestra is to make financial messaging protocols more plug and play**
Questions Midstream

- Conference Code to access the polling system
- Are you looking to move connectivity technology to the cloud in 2019?
- Will your technology budget for 2019 versus 2018 be:
  - Up >10%
  - Up >5, <10
  - Up <5%
  - Flat
  - Down <5%
  - Down >5, <10%
  - Down >10%
- Would you like a customised presentation made to your firm about the benefits of Orchestra from the FIX Global Technical Committee?
What is Orchestra?
DevOps (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between Development and IT Operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between the two business units.

The DevOps culture puts a focus on creating a fast and stable work flow through development and IT operations. One main goal of DevOps is to deploy features into production quickly and to detect and correct problems when they occur, without disrupting other services.

Webopedia

DevOps - Wikipedia Entry
FIX Orchestra spans the DevOps life cycle
What is FIX Orchestra and what does it do?

- **FIX Orchestra is a standard for exchanging machine-readable rules of engagement.**
- **FIX remains the protocol on the wire.**
- **No changes required to your existing FIX engine (but you may wish to enhance it to read an Orchestra file).**
- **FIX Orchestra is metadata about a specific implementation of FIX.**
- **Orchestra is not a product, although FIX Trading Community is kickstarting open-source demonstrations.**
- **Vendors and firms are free to develop proprietary implementations, so long as they are conformant to the standard.**
What is FIX Orchestra and what does it do?

Orchestra content, all machine readable

- Message structure by each scenario. Implemented as an extension of FIX Repository.
- Accepted values of enumerations by message scenario
- Workflow: when I send this message type under this condition, what can I expect back?
- How external states affect messages, e.g. market phases, order state, price
- Express a condition such as for a conditionally required field using an expression language
What is FIX Orchestra and what does it do?

● Content is a composed of multiple feature categories.
  ● Application layer structure and behavior – independent of encoding such as tag=value, FIXML, SBE
  ● Session layer behavior
  ● Operational: session configuration—identifiers and transport settings

● A firm does not need to implement every feature of FIX Orchestra to gain some benefit.
  ● Just want to share message definitions and conditional fields? That’s doable
  ● Want to extend to describe message responses, scenarios, and basic states? That’s doable
  ● Want to fully model in detail the FIX service? That’s doable
What is FIX Orchestra and what does it do?

Orchestra process of engagement

- Counterparties exchange their Orchestra files, either statically or exposed through network interfaces for discovery.
- Counterparties compare their own file with that of their partner.
  - Discover differences and restrictions
- Automatically generate:
  - FIX engine configuration
  - Application configuration and code
  - Test cases and sample messages
  - Documentation for those pesky humans
Orchestra is an interface definition

- Orchestra defines an interface to service offerings or service endpoints
- You don’t need to modify the internals of your applications
- Almost all existing FIX infrastructure has the provisioning information of FIX connections and data dictionaries stored in multiple places and multiple formats.
- Orchestra can be used to define the service in one place within version control
- Then simple scripts can be created to read Orchestra files and update configuration files for various services
- The benefits of Orchestra can be available with minimal investment
FIX Orchestra supports innovation

Possible uses and tools

- Generate and run conformance tests
- Capture best practices as an Orchestra file instead of text
- Regulate internal flows within a large organization as well as between counterparties
- Orchestra is a contract for behavior – use it to generate an emulator for testing
- Capture an Orchestra file from FIX logs
- Analyze FIX logs for conformance to specified behavior
- Let’s go further…
  - Generate Execution Management, Order Management, Smart Order Routing, Order Matching behavior based upon exchange of state machine descriptions contained within FIX Orchestra files
FIX Orchestra roll-out and adoption

FIX Trading Community tasks

● Standardization
  ● Working group uses GitHub to collaborate on schema and samples
  ● Working group proposes standard to Global Technical Committee
  ● Big bang or roll out in phases? e.g. message structure/FIX Repository 2016 edition, scenarios and state machines, condition DSL, session layer

● Develop Orchestra files representing existing FIX standards
  ● Enhancement of FIX Repository
  ● Best practices by asset class, region, etc. – delegate to various working groups
  ● Publish in GitHub

● Develop open-source utilities
  ● File comparison and reporting tools
  ● Validation against schema
FIX Orchestra roll-out and adoption

Firm and vendor tasks

- Develop open-source or proprietary utilities
  - Compose Orchestra file from other sources
  - Orchestra file editors
  - Create or adapt configuration, test and code generators
  - Web interface for session configuration – web services or semantic web technologies
  - Certification tests
- Exchange files with counterparties
  - Pilot program for early adopters
- Give feedback to working group
How can you use Orchestra?
Integrating FIX Orchestra with Test Tools for Exchanges and Clearing
Exactpro Open Source Strategy

- Our testing tools software is underpinned by OSS
  (QuickFix/J, MySQL, PostgreSQL, set of open source development frameworks, Linux, etc)

- We always wanted to give back to the Open Source community

- Exactpro has donated 3 tools to Open Source:
  
  - **JackFish** is applied for UI test automation;
  
  - **Nostradamus** includes ML algorithms;
  
  - **Sailfish** is used for active real-time testing of Exchanges, MTF & broker systems;

  **ClearTH**, our core tool used for testing Post-Trade platforms, is next in line to be released to open source soon…
Exactpro Open Source timeline

- GitHub
- open source
- Sailfish Testing Tool
- Nostradamus
- JackFish

2018

2019
Sailfish and ClearTH

Sailfish & ClearTH test harnesses:
- have proven themselves across a number of our clients’ projects including integration and onboarding process on a variety of platforms (markets, RTC, CCP, CSDs, Ticker Plants, SORs)
- both test harnesses support multiple trading and post trade protocols

Extant plug-ins for Industry standards:
- FIX and dialects, FAST, SWIFT ISO, ITCH, etc.; and Proprietary protocols (MIT, SAIL, HSVF, RTF, RV, Reuters, Fidessa OA, Quant House, etc.)
- New plug-ins for additional protocols developed by request (codecs are compatible with all Exactpro test tools)

WIP (work in progress) for is to use FIX Orchestra as protocol dictionaries.
Collaborating with big data:

- supervised learning
- process mining
- clusterization
Using FIX Orchestra at the confluence of FT & NFT

- **FIX Orchestra** defines workflows for *static analysis* of test automation scripts that are part of functional and regression libraries. This is *before* executing. Use cases:
  - Conformance testing
  - Test coverage analysis

- **Process mining with FIX Orchestra workflows:**
  - Validation of invariants under random load
  - Comparison of coverage from random load/regression/production

**FIX Orchestra workflows created from:**

- Specifications
- Test runs (Logs,msgs) FT / NFT
- UAT/Production (Logs, msgs)

**Workflows comparison:**
- Gap analysis/coverage in the test library/run;
- Efficiency in test execution

- Edge cases/race conditions;
- Are all the use cases from Prod/UAT covered?