QuantMinds Americas Virtual Presentation

Sept. 23, 2020

A Smarter Model Risk Management Discipline Will Follow From Building Smarter Models

An Abbreviated Guide for Building the Next Generation of Smart Models

---

Presented by

Jon R. Hill

Adjunct Professor of Financial Risk Engineering

NYU Tandon

jh7051@nyu.edu;

jonhill@optonline.net

About the Speaker:

Jon Hill, Ph. D., is a Subject Matter Expert in Model Risk Management and is an independent consultant to financial institutions.

Dr. Hill is an adjunct Professor at New York University where he teaches a graduate course in Model Risk Management and Governance in the Department of Finance and Risk Engineering.

Dr. Hill serves as Head of the New York Chapter of the Model Risk Mangers International Association (MRMIA), established in 2018. The association's purpose is to promote awareness of model risk to the broader risk and financial communities and to provide a forum for topical discussion of model risk management challenges and regulatory requirements.

Jon is a former Managing Director at Credit Suisse with over twenty years of experience in various areas of quantitative finance. As head of the Global Head of Model Risk Standards at Credit Suisse he led a team comprised of 14 model risk managers in New York London, Zurich, Mumbai and Singapore. Jon's team had responsibility for the ongoing identification, measurement, risk rating, inventory and monitoring of CS corporate model risk across all business units, regions and legal entities.

Prior to joining Credit Suisse in 2017, Jon founded and led the Validation team for Market and Operational Risk Models at Morgan Stanley for 6 ½ years. Prior to Morgan Stanley Jon performed hands-on model validations at Solomon Smith-Barney (which later became Citigroup) was a member of a quantitative finance research team.

Dr. Hill Holds a Ph.D. in biophysics and is a published author in the field of model risk management. Jon is a frequent speaker and chairperson at MRM conferences in both the US and Europe.

WHAT IS A MODEL, AND WHAT IS MODEL RISK MANAGEMENT?

A Brief Review

In April, 2011 the FRB & OCC Jointly Issued SR11-7/OCC2011-12.

This 21-page Document Set the Bar for Model Risk Management (MRM) at All Conforming Firms

SR11-7 provides a working definition of a financial model:

"For the purposes of this document, the term *model* refers to a quantitative method, system, or approach that applies statistical, economic, financial, or mathematical theories, techniques, and assumptions to process input data into quantitative estimates." *

"A *model* consists of three components: an information input component, which delivers assumptions and data to the model; a processing component, which transforms inputs into estimates; and a reporting component, which translates the estimates into useful business information."

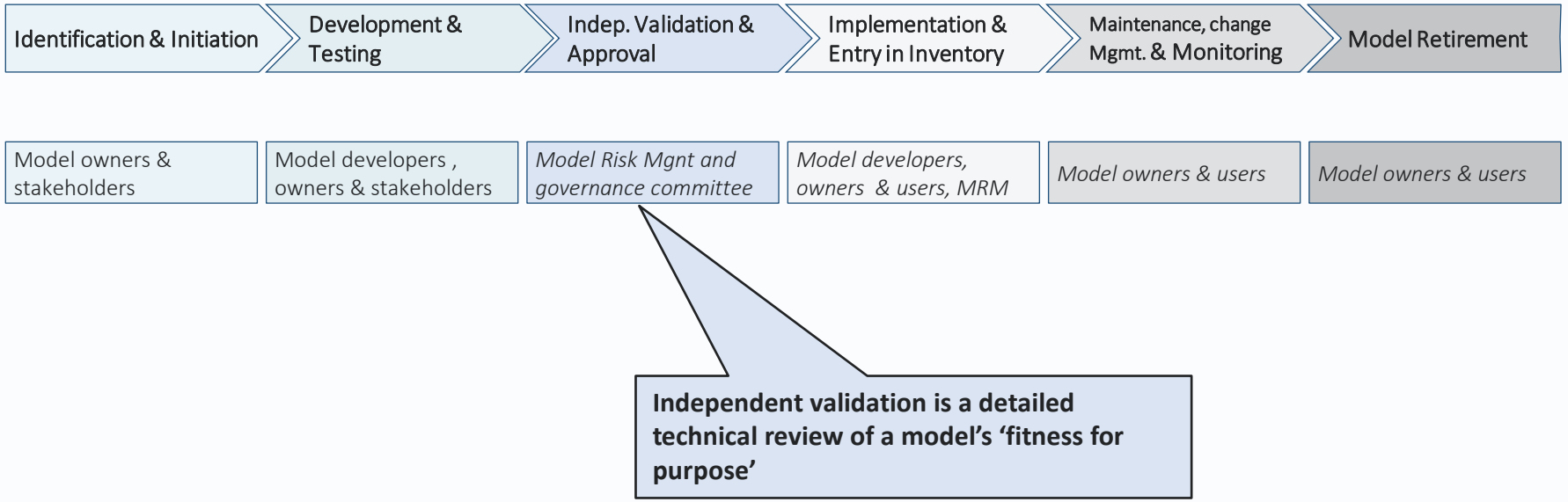
* Because they are based on assumptions and not first principles or laws of nature, all financial models are 'wrong' at some threshold of accuracy. George Box's famous 1987 quote captures this reality: "All models are wrong. The important question is how wrong can a model be before it stops being useful".

Five Key Takeaways From SR11-7 For Model Risk Management in Finance

1. All financial models employ approximations based on assumptions. Therefore, model risk can never be completely eliminated, but it can be mitigated.
2. Model validation concentrates on the risks within a quantitative model. The purpose of validation is to determine whether or not a model is 'useful'.
3. Model governance addresses the risks *outside and between* quantitative models within a firm's *model ecosystem*. Governance impacts every phase of a model's life cycle, validation being just one of them.
4. A robust model governance framework will provide complete coverage for, policies and procedures, roles and responsibilities for ownership, control and compliance, model validation and ongoing monitoring, model risk assessment, documentation and model inventory.
5. The biggest challenge for model inventory is ensuring accuracy and completeness. At almost all firms this is performed through a manual attestation process

Typical Life Cycle of a Financial Model

A Well-Designed and Enforced Model Governance Creates a Control Framework that Touches on Every Phase of a Financial Model's Life Cycle



Of the Many Challenges Confronting Today's Model Risk Managers, Five of the Most Daunting Involve Model Inventory

1. SR11-7 requires banks to create and maintain a complete and accurate inventory of all models.
2. SR11-7 requires banks to be able to aggregate model risk across the firm.
3. Understanding model and data inter-dependencies within a firm's *model ecosystem* can be especially problematic since mapping dependencies relies on multiple levels of attestation
4. Virtually every financial firm tries to satisfy these requirements for a rigorous MRM through verbal attestations (voluntary declarations) by model owners and stakeholders (developers, supervisors and users).
5. Because attestation is a manual and error-prone process it is questionable if any firm can truthfully claim to have a complete and accurate model inventory much less a complete map of model and data inter-dependencies.

To Illustrate this Dilemma, Consider Eight Vexing Questions Bank Examiners Might Pose to a Model Risk Manager About Inventory

1. What is the exact number of different models that have been used over the last year?
2. How often has each model been executed, by day, by month, by year? Can you identify the most frequently and least frequently executed models?
3. Where are the firm's models being used? By business unit, legal entity, geographic regions?
4. Can you provide a complete list of the models used by each of the above entities over the last year, as well as all upstream/downstream model & data dependencies?
5. Are there any models in your inventory with an active status that were not executed during the last year?
6. Are there any models that were executed on any of your firm's computers that do not appear in inventory? Please provide a full listing.
7. Are you able to provide a full list of the IDs of models that exhibit significant seasonality? If so, what are the peak and troughs of seasonal model usage.
8. Were there any instances of a retired model still being executed during the last year?

It shouldn’t be so difficult for top-tier financial firms to give accurate quantitative answers to these types of questions about model usage, but it is.

The only way most firms can answer them is through attestation by model owners and users.
But attestations are often no better than educated guesses!

As a result, there are often discrepancies between what is in inventory and what model owners have attested.

Shouldn’t There Be a Better Way Than Manual Attestation by Model Stakeholders?

Resolving ownership discrepancies can require numerous iterations of the manual attestation process to determine the current correct ownership of orphan models or models that have been retired but are still in use.

Particularly problematic are upstream and downstream dependencies between models. Model owners often do not have complete knowledge of all of the downstream models that receive their models’ output as input.

In an age of automation, machine learning and big data we really should ask ourselves if we cannot find better ways to make firmwide model and data usage more transparent.

Perhaps Some Answers May Be Found by Engaging a Model in Conversation ….

Manual attestation can be both clumsy and error-prone – some models may simply be overlooked in the process, some may be 'orphans' while others simply fall through the cracks of antiquated monitoring systems.

But if models could only talk, perhaps they could give us some advice about how to better understand 'how, when and where' they are being used.

A Conversation With a (Snarky) Model Might Go Something Like This:

MRM: Hey Model! Could we please have a little talk?

Model: Sure, but you may regret it later …… !

MRM: Why so? Some of my best friends are models!

Model: Because you MRM guys are not really the sharpest crayons in the box, are you? Still living in the stone ages?

MRM: Whoa! Zinger! Why would you say such a thing, model?

Model: MRM spends so much time and effort collecting inaccurate, error-prone manual attestations from model owners & users! Just to inventory which models are being used. That is sooo 20th century!

Model: But you could make your life soooo much easier if you would just ***reverse the manual paradigm***. It's simpler, more accurate and, quite frankly, just kind of obvious!

Conversation with a snarky model - continued

MRM: But tell me my friend, how can MRM possibly reverse a paradigm that's been in place for years?

Model: The solution is all around you, in every smart device that you use every day: in printers, computers, cars, smart phones, email, data networks ….. all you have to do is look!

MRM: Uhhhh, I still don't think I get what you're driving at.

Model: Then listen carefully, dingbat. Those devices all 'know' who they are because they have unique IDs and a means of broadcasting, which I don't have! If you would just give me an embedded ID token and a voice then I could tell you how, when and where I am being used! It's sooooo obvious!

MRM: Now I get it! All I have to do is teach you your name and how to talk. Is that right? OMG, it's an epiphany. But, ummm, errr, exactly how would I do go about doing that?

Model: Here's how: just follow the rest of this presentation and maybe it will start to sink in. (As I was saying ….. 😐)

Models That Are Just Smart Enough to Report their Usage Could Address Many of These Shortcomings

The root cause of these difficulties is not hard to find: our models, no matter how sophisticated their algorithms and implementations, are nevertheless rather **dumb**[+] **when** compared to an HP printer or an iPhone. A '**smart**' model should be able to report who it is, how, when and where it is being used and which upstream models it depends on.

Designed correctly, smart models could eliminate the need for a manual attestation process.

[+] Here the rubric 'dumb' applies to models that lack any rudimentary form of self-awareness.

Examples of Tech Solutions to Usage Tracking Surround Us Every Day:

- A smart phone 'knows' its unique serial number (it's embedded in the permanent onboard memory that stays with the phone for life).
- A modern washing machine knows its own serial number too, so does an automobile. These are embedded in the onboard electronics that control these devices.
- Even before electronics, serial numbers were stamped on the frames of every automobile that Henry Ford produced and somewhere on almost all manufactured products of any significance.
- The Uber ride service tracks the current geographic location of every one of its active vehicles and advises clients on both the location & estimated time of arrival of their ride.
- Today, Tesla has the ability to track every one of their vehicles in service at a given time for: location, travel speed, level of charge and other usage indicatives.
- Hewlett Packard smart printers send usage data to HP central tracking command, including IP address, number of pages printed & ink levels. HP mails new ink cartridges before I run out. If HP can monitor 100s of thousands of printers globally, why can't financial firms do the same for a few thousand models?

What is it That Smart Devices Like Printers, iPhones and Tesla Vehicles Have That Financial Models Do Not?

To understand what is missing from financial models that makes them 'rather dumb', we have only to dig a little deeper:

- Look inside the source code for a complex financial model and what will you find, regardless of the programming language?

- ✓ Probably very sophisticated algorithms, highly efficient optimized coding, very likely using the latest concepts in object-oriented design, perhaps code for efficient, dynamic memory management.

- ❖ But what you won't find, as a general rule, are any very simple lines of code that look something like this:

```
int Model_ID = 12345678 ;
int Model_Version = 3.10 ;
int Model_Usage_ID = 321 ;
Char Model_Name[] = BlackScholesPricer ;
```

The Lesson from The Previous Slide Is Simply This:

Financial Models Do Not ‘know’ Their Own IDs!

Software implementations that are classified as models are assigned unique IDs as a shorthand identifier. At most firms these IDs typically appear in 3 places: in the model documents, in the validation documents and in the inventory database as a lookup index. As shown in a previous slide, *where they do not appear is within the actual model source code.* It is in this sense that models do not ‘know’ who they are.

The root cause of model usage opacity may be traced to this single surprising blind spot in most firms’ model management discipline. Adding this one piece of information to a model can create a path to mitigating or even eliminating model inventory and usage uncertainties. It’s a matter of creating ‘smart’ models that are enabled to tell MRM how, when and where they are being used.

The crux of the matter:

In financial firms, model developers and model risk managers work in separate silos

Model Development vs. Model Risk Management

- ✓ Model Risk Managers are tasked with identifying and mitigating the holistic model risks that reside in a firm's model ecosystem fabric.
- ✓ Model developers are tasked with designing and implementing and testing models that efficiently and accurately convert input data into useful outputs.
- ✓ These two groups tend to work completely independently within most financial firms.
- ✓ In most firms models are managed and executed in a number of often incompatible execution platforms.
- ➢ **One consequence of this silo mentality is that model developers tend to have little interest or motivation for modifying their models to accommodate the requirements of MRM.**

But if they worked together ….

Some simple changes could be made by developers to a firm's models that would greatly improve MRM discipline.

***Here are three of them*:**

1. Create identity tokens composed of unique model indicative data as suggested in slide #17. Embed identity tokens within each model's source code.

2. Next, embed active intelligent agents to accurately track model usage and support creation of a dynamic inventory model.

3. Exchange identity tokens between inter-dependent models (and data) to create a dynamic map of model and data dependencies. (No Financial firm seems to do this today.)

What Kind of Usage Data Might an Embedded Active Intelligent Agent Send to Model Risk Managers?

Similar to aviation transponders used to track civilian aircraft:

A transponder (or tracking) function can act as an intelligent agent that would be called once each time a model is executed. At a minimum, it should 'broadcast' the following fields to a centralized model usage repository for each execution event:

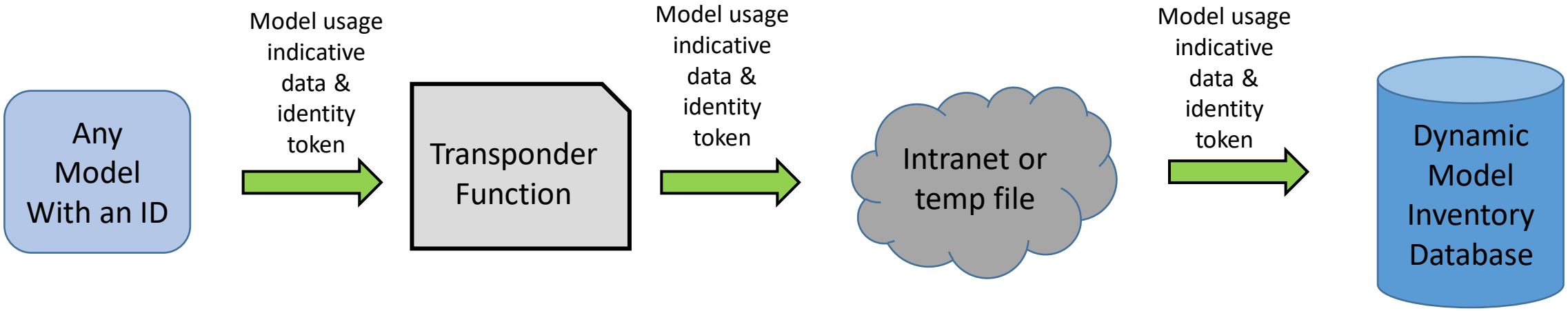
1) Model and Usage IDs
2) Model Name and Version Number
3) Timestamp – year, month, day, hour & minute
4) MAC or IP address

If these data are stored for every execution event for each model in inventory that is assigned a unique ID, a treasure trove of model usage data will be accumulated over time.

This is the second step towards the creation of a 'smarter model'**:** models equipped with an embedded intelligent agent that can automate usage tracking. This data can form the basis for a ***'dynamic model inventory',*** one that incudes continuously updated information about how, when and where models are being used. Passing tokens from upstream to downstream entities can form the basis for a dynamic map of model and data inter-dependencies.

Conceptually, Model Usage Tracking Is Really Rather Simple

The Most Important Goal is to Achieve Independence from Execution Platforms!



But the devil may be hiding in the details …

Note: It may not be necessary for the Transponder to send data to a centralized database via the Firm's intranet. Any type of communication pipe that a Firm's IT staff choose may serve the purpose of populating a central database with a log of model usage statistics, indexed by model ID and collected over a significant length of time, e.g. at least one year.

A Proof of Concept Using Simulation Is One of the Best Ways to Demonstrate The Potential of an Idea

Simulation offers a practical way to establish the value that can be added by embedding model identity tokens and transponder tracking functions. This can be implemented without impacting production models by leveraging a portfolio of synthetic, or ‘dummy’, models consisting only of imbedded IDs and transponder functions.

How Might Transponder Simulation Results Be Presented?

The graphs displayed in the following slide were produced by collecting 4 data fields for each of the 100,000 model execution events on 100 ‘dummy’, or synthetic, models with embedded model ID tokens and prototype transponder functions (slide #23) over a simulation horizon of 3 1/2 years.

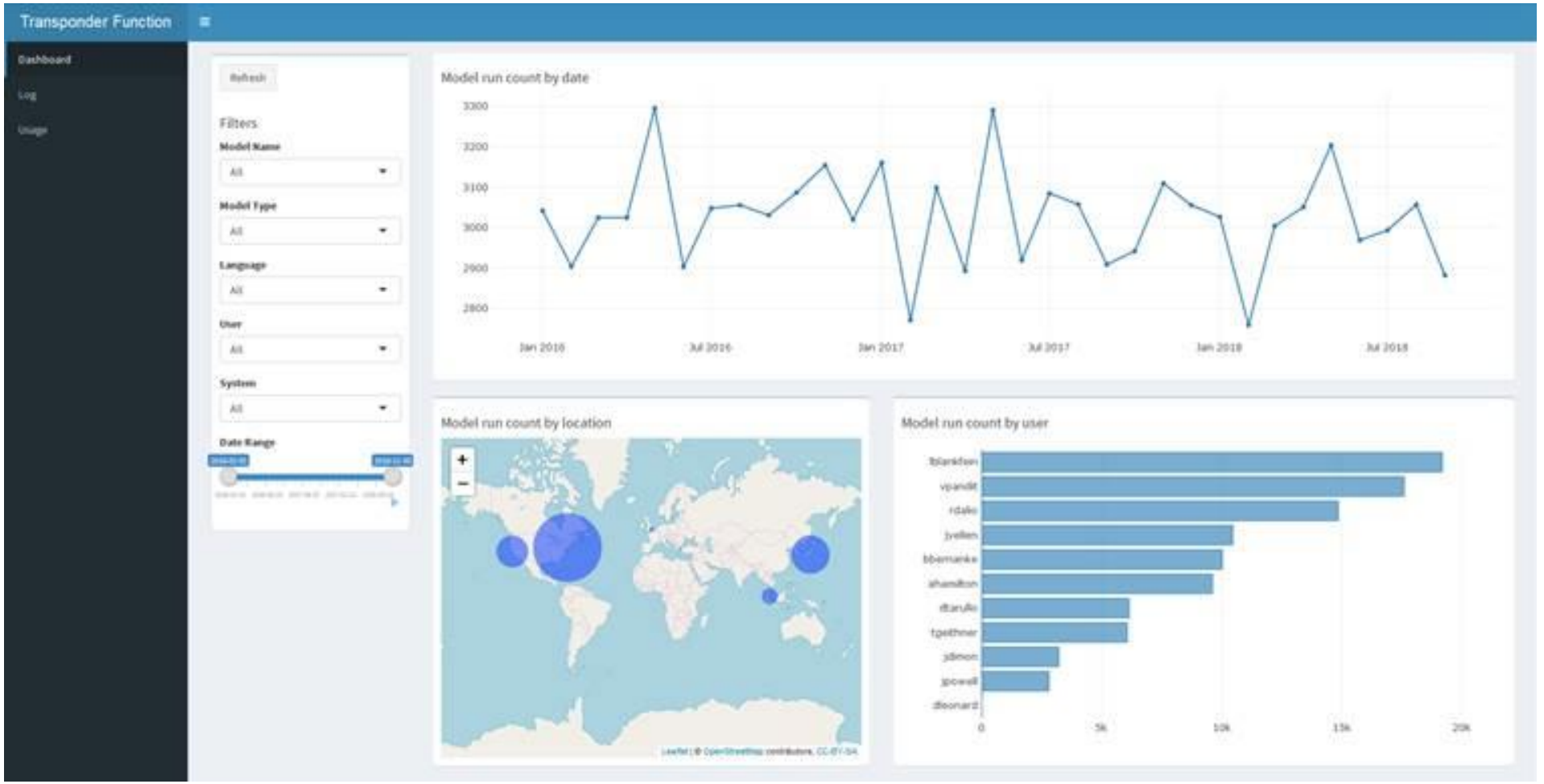
Model ID, Model Name, Time Stamp and MAC or IP Address are the only model usage data required to produce the following types of graphs:

a) Timeline plots of usage for any model or grouping of models

b) A histogram distribution of model execution frequencies

c) A global map showing concentration of model usage
   - With sufficient history, the global map can be a animated to illustrate changing usage patterns through time

Simulation Dashboard for a Prototype Embedded Transponder 100 Synthetic Models and 100,000 Random Execution Events*



* The prototype Transponder Function and dashboard display used for this simulation were developed in collaboration with the author by David Leonard at FI Consulting, Arlington, VA. The usage plots were produced by extracting 4 data points from each simulated event: ID, model name, timestamp and MAC/IP address. The graphical dashboard was implemented on an Amazon Web Services (AWS) cloud platform.

A Transponder Prototype Written in the R Language Might Look Something Like This [+]:

The Call to Execute the Transponder Function:

postLog("1500", “BondPricer", "CashFlow", "R") **(This is the single line to be embedded in model source code)**

The R source code for the Transponder Function used in the previous simulation

It is not necessary, or even recommended, for the Transponder source code to be inserted into the model’s source code, but rather as part of a compiled library that can be linked together with the model’s compiled code during the build process.

```
library(httr)
 postLog <- function(modelid, modelname, modeltype, language) {
  p <- POST(paste0(url,
                   'id=', runif(1)*10000000,
                   '&modelid=', modelid,
                   '&modelname=', modelname,
                   '&modeltype=', modeltype,
                   '&language=', language,
                   '&date=', as.Date(substr(gsub(" ", "_", Sys.time(), fixed = TRUE), 1, 10)),
                   '&time=', substr(gsub(" ", "_", Sys.time(), fixed = TRUE), 12, 19),
                   '&user=', as.character(Sys.info()['user']),
                   '&location=', location,
                   '&sysname=', as.character(Sys.info()['sysname']),
                   '&ip_address=', gsub(".*? ([[:digit:]])", "\\1",
                                        system("ipconfig", intern=TRUE)[grep("IPv4", system("ipconfig", intern=TRUE))])))
}
```

[+] R code for this prototype Transponder Function was developed by David Leonard at FI Consulting, Arlington, VA.

Advantages

Solution is Inside the Model: The Model Transponder approach places the tracking usage software inside each model rather than relying on an external execution platform to track and store usage statistics.[1] Will work equally well for large IT-controlled or small standalone End User Computing (EUC) models

Readily Scalable: placing the usage tracking solution inside models will scale in a straightforward manner from 10 models to 10,000 since it does not require or assume consistency among external environments.

Comprehensive Solution: Because it is platform independent it is a global solution that will operate on any Firm computer that has access to the firm's intranet (or that can write results to a temporary file).

Incremental: The proposed innovation can be implemented incrementally over time beginning with limited sets of models such as those used for CCAR/DFAST stress testing or the set of pricing models in the high-risk tier. Changes could be included in the regular release cycles.

Can Dyamically Trace Model and Data Inter-Dependencies: Offers a direct token-based means for comprehensively identifying upstream and downstream dependencies based on execution processes rather than attestation by model developers.

[1] Most production models at banks are managed by host execution platforms, although most EUC models are not. It is possible for execution platforms to be designed or modified to track usage statistics but large firms may have hundreds of different platforms and each would have to be customized to provide similar data. Any changes would have to be made to all such platforms. This is not a readily scalable solution.

Disadvantages

Touches Every Model: Requires some minor additions to the source code of each model to be tracked, although performance will not be affected.

High bandwidth from heavily used models could bottleneck the Firm's intranet

Vendor models present a special challenge – doubtful vendors would agree to install transponders in their models. But there may be workarounds through the inhouse execution scripts or host programs that Firms use to interface between the vendor code and the Firm's computers.

EUC (i.e. spreadsheet) models could present challenges as well, but not insurmountable one. If the spreadsheets incorporate model code (such as VBA or linked libraries) the transponder tracking function can be embedded within the code or linked library.

Most large, established financial firms tend to be resistant to change, especially innovations. Expect pushback from IT organizations.

[1] Most production models at banks are managed by host execution platforms, although most EUC models are not. It is possible for execution platforms to be designed or modified to track usage statistics but large firms may have hundreds of different platforms and each would have to be customized to provide similar data. Any changes would have to be made to all such platforms. This is not a readily scalable solution.

The Takeaway

Embedding Identity Tokens and Intelligent Agents into Models can Reverse the Manual Attestation Paradigm

In a Nutshell: Model Attestations Are Typically Performed Backwards!

Currently, MRM relies on model owners and users to identify the complete set of models they use through a manual and error-prone attestation process.

Wouldn't it just be easier and more accurate to enable models with intelligent agents that can inform MRM about how, when and where they are being used? ***Reverse the paradigm*** and smart models will do the work for MRM.

This single innovation could replace our 20th century manual attestation processes!

But however it is done, the most important takeaway from this presentation is that any usage-tracking solution should reside inside the model code! Only in this way will it be both portable and fully scalable across all execution platforms and implementations.

No Longer Just Vaporware
SAS Institute is Implementing It!

RQS Smart Models*

SAS Institute Model Risk Management

*SAS slides presented with permission of David Asermely, Head of SAS MRM

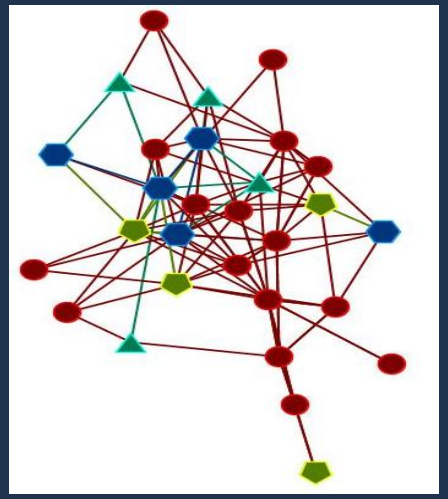
Copyright © SAS Institute Inc. All rights reserved.

sas
THE POWER TO KNOW.

Dynamic model inventory

Replace manual and error prone attestations

Accurate and constantly updated model maps



The SAS Institute Has Created a Prototype Testbed Implementation within their MRM Platform:

- Embedded unique identity tokens endow models with a rudimentary level of self-awareness
- Unique identity tokens can be embedded in all model results
- Identity tokens are passed from upstream models and data to their downstream recipients
- Can be used to create a dynamic map of model inter-dependencies that is updated whenever a firm's models are executed.
- Model usage data is auto-generated – eliminates the need for manual updates by model stakeholders.
- The prototype version is now available for testing by select SAS clients

Copyright © SAS Institute Inc. All rights reserved.

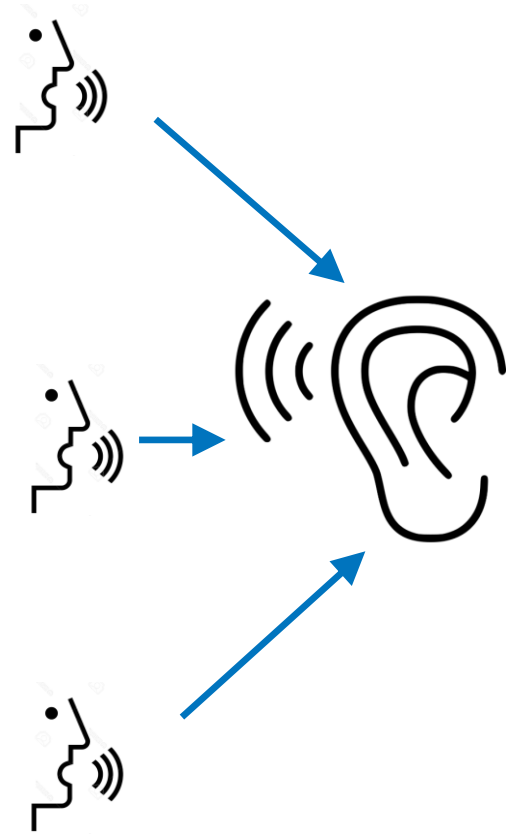
Smart Models

To Support Smarter Model Governance

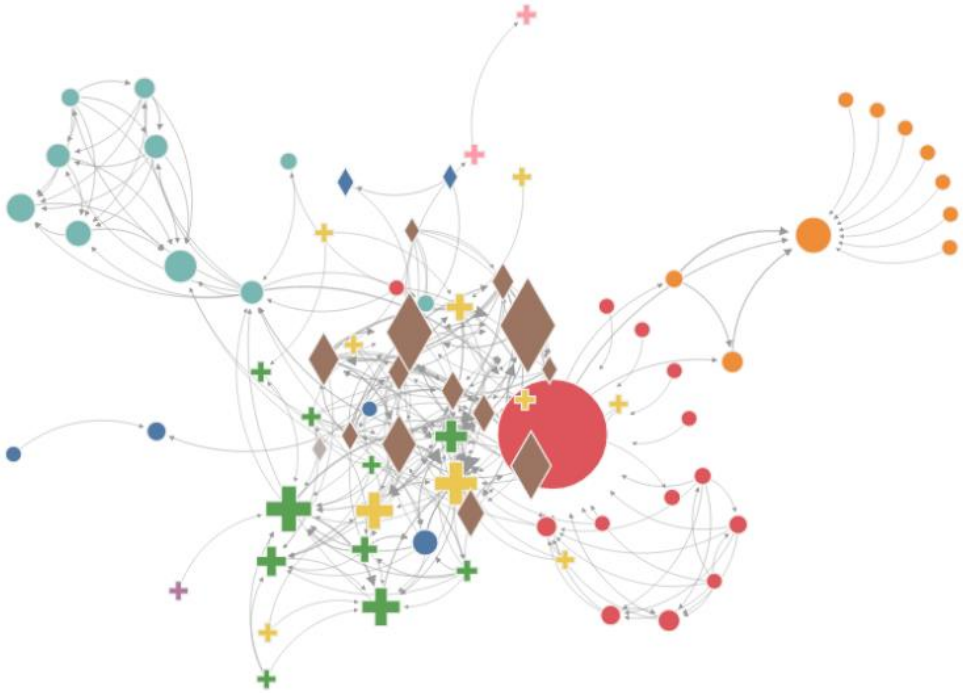
Model Usage Data & Token Passing

IFRS 9 Accounting & Finance

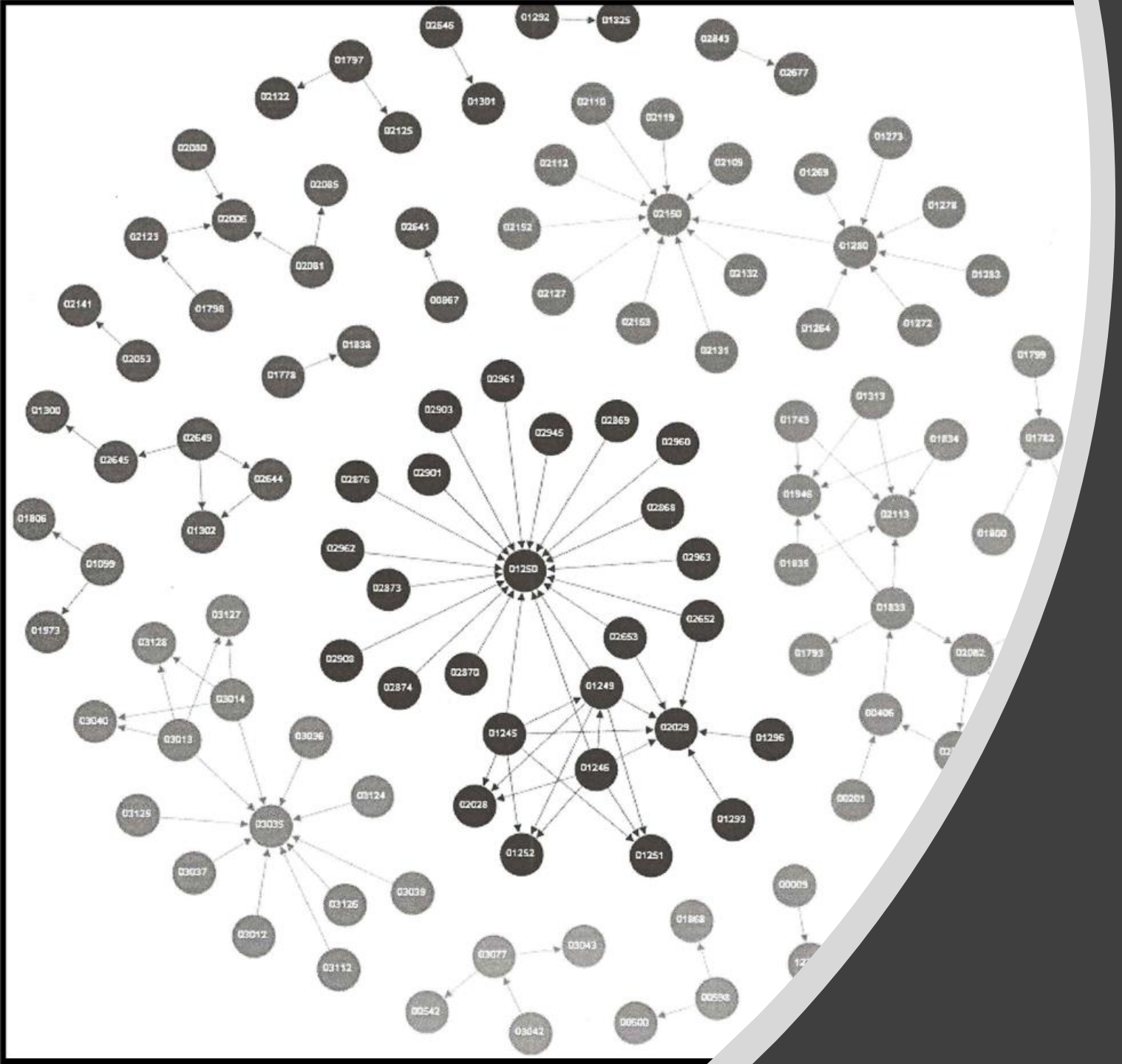
Decision Making (e. g. Loans)



Model Risk Mngt.



Copyright © SAS Institute Inc. All rights reserved.



A Complete Map of Upstream/Downstream Model and Data Inter-Dependencies Within a Firm's Entire Model Ecosystem Can be Complex and Intricate.

Graphical Network Diagrams are a Useful Way of Capturing these Complexities.

Recent Publications by the Speaker That Address Some of the Many Challenges of Model Risk Management

Inventory, risk-tiering, model inter-dependencies and other topics in model risk management are pursued in greater detail in the following refereed journal articles:

1) Hill, J. R. (2018) "Shouldn't A Model 'Know' Its Own ID?", *The Journal of Structured Finance,* Fall, pp. 89-98

2) Hill, J. R. (2019) "The 14 Top Challenges for Today's Model Risk Managers: Has the Time Come to Think About Going Beyond SR11-7?", *The Journal Of Risk Management In Financial Institutions,* Spring, Vol. 12, 2, pp. 145-167

3) Hill, J. R. (2020) "A Smarter Model Risk Management Follows From Making Smarter Models: An Abbreviated Guide for Building the Next Generation of Smart Models", *The Journal Of Risk Management In Financial Institutions, special MRM edition:* Vol. 13, 1, pp. 24–34 © Henry Stewart Publications 1752-8887 (2020)

Request Copies From: ***jh7050@nyu.edu*** ***or*** ***jonhill@optonline.net***

A Smarter Model Risk Management Discipline Will Follow From Building Smarter Models

Final Slide