Journal of Behavioral and Experimental Finance 23 (2019) 12–22

Contents lists available at ScienceDirect

Journal of Behavioral and Experimental Finance

journal homepage: www.elsevier.com/locate/jbef

Full length article

Simulation-based learning using the RIT market simulator and RIT decision cases

Kevin Mak[a,b], Thomas H. McCurdy[c,d,*,1]

[a] Real-time Analysis and Investment Lab (RAIL), United States
[b] Lecturer in Management, Graduate School of Business, Stanford University, United States
[c] Academic Director, BMO Financial Group Finance Research and Trading Lab (FRTL), Canada
[d] Professor of Finance and Bonham Chair in International Finance, Rotman School of Management, University of Toronto, Canada

ARTICLE INFO

Article history:
Received 1 December 2018
Received in revised form 5 May 2019
Accepted 6 May 2019
Available online 15 May 2019

ABSTRACT

RIT is a custom-designed product consisting of a market simulator platform and learning-by-doing decision cases that simulate risks and opportunities for various financial securities, investment and risk management strategies. Using the market simulator with custom designed cases, linked in real-time to decision support models applying the relevant theory, participants learn how to make good real-time decisions in complex environments for which there is material uncertainty. The market aggregates participants' decisions and provides immediate feedback concerning the success of their strategies, allowing them to adapt their strategies after each replication of the case. Besides supporting teaching and training, the decision cases facilitate competitions and events at many different levels. Given the high degree of flexibility and customization that is available to the market administrator, low-friction engagement for participants, and high-resolution data logging, the RIT product is also a valuable resource for investigating research questions across a wide-range of topics.

1. Introduction and motivation [2]

Uncertainty about the future is pervasive. Decision-making under uncertainty is an important skill. Simulation-based learning provides participants with a hands-on approach to practice decision making in a controlled environment where they can immediately observe the outcomes of their decisions. By being able to analyze the consequences of their decisions in different situations, students are able to learn how to make good decisions for complex problems given varying degrees of uncertainty about the future. The simulation-based tools enable students to apply and develop what they have learned in the classroom in order to solve problems they would find in the workplace, but in a 'safe' and controlled environment.

The impact of simulations on university-level learning has been investigated most thoroughly in aviation and medical and health professional education/training. For example, a systematic review of medical simulations by Issenberg et al. (2005) concludes that simulations benefit learning when they: are integrated into curricula and capture a variety of (clinical) conditions; provide clear goals and outcomes; provide a range of difficulty levels and adapt to multiple learning strategies; require repetitive practice and feedback; promote active participation in learning; and provide a safe environment for errors. In addition, a systematic review and meta-analysis provided by Cook et al. (2011) concludes that "in comparison with no intervention, technology-enhanced simulation training in health professions education is consistently associated with large effects for outcomes of knowledge, skills, and behaviors and moderate effects for patient-related outcomes".

Salas et al. (2009) discuss advantages associated with simulation-based training for management education, including: providing a more complex and realistic learning environment than other training strategies while still allowing reality to be simplified and manageable; providing a (relatively) risk-free environment for learning and experimentation; leading to learning in reduced time; is an ideal method for training infrequently engaged but critical skills; is more engaging than other training methods; and can impart complex applied competencies. Woodhouse and McCurdy (2014) discuss educational benefits of

* Correspondence to: Rotman School of Management, University of Toronto, 105 St. George Street, Toronto ON M5S 3E6, Canada.
E-mail addresses: kevinmak@stanford.edu (K. Mak), tmccurdy@rotman.utoronto.ca (T.H. McCurdy).

1 The authors are grateful to Jasper Chan, Marco Salerno, Eric Kang and FRTL Assistants for their contributions to the RIT product. We are also grateful for many helpful suggestions for this paper from two anonymous reviewers, Stefan Palan (Editor), and R. Woodhouse. The RIT webpages can be accessed at http://rit.rotman.utoronto.ca.

2 This section draws on McCurdy and Woodhouse (2002) and Woodhouse and McCurdy (2014).

https://doi.org/10.1016/j.jbef.2019.05.003

interactive simulations, for example: facilitating deep learning; clarifying troublesome and threshold concepts; providing motivation and opportunities for practice; and allowing immediate feedback and assessment.

As highlighted in later sections of this paper, all of the above advantages of simulation-based learning and training are achieved using the RIT Decision Cases implemented on the RIT (Rotman Interactive Trader) market simulator platform (an industry strength order-driven market). The RIT Decision Cases simulate risks and opportunities for a broad range of tasks/jobs. Students can use their classroom lessons to try out decisions. The simulated market aggregates the decisions of all participants and provides immediate feedback on the success of their strategies allowing them to adapt their strategies after each replication. Multiple replications of the case simulation allow students to practice finding a robust strategy for the types of risks and opportunities they face for a particular task. The RIT product contains many performance feedback tools, which facilitate discussion, accelerate learning, and provide an opportunity for instructors to provide incentives for participants to focus on the learning objectives of each case. As a result, students learn how to develop and use their knowledge and skills in complex, real-time environments for which uncertainty is material.

There are few clinics or teaching practice settings to enable students in the social sciences to obtain experience acquiring skills using simulation. Why is simulation-based learning more entrenched in aviation and medical training? Perhaps because the occurrence of errors in those settings are usually more easily observed and the implications of making errors in those settings can be catastrophic.

In social science applications, such as finance and economics, the risks and uncertainties associated with decisions are perhaps even more complex since they are also affected by model uncertainty, parameter uncertainty, and signal extraction issues for the varying signal-to-noise ratios associated with most decisions. Recall the literature on detecting skill versus luck associated with portfolio managers' performance. For example, Fama and French (2010) use bootstrap simulations from long histories of fund returns in order to confidently identify skill. This research example highlights the necessity of using simulations to separate signal from noise in many empirical finance applications; an important skill for both students and practitioners.

Simulations can train participants to better identify the implications of parameter and model uncertainty and acquire skills to manage those risks. There is also pedagogical value associated with sequencing skill acquisition from mastering single skills (dealing with one risk) and then including additional risks as we sequence through a set of cases on a particular topic until they can manage the capstone case in which several risks can interact.

Here again, the analogy with a flight simulator is useful since airplane accidents often happen when risks cascade so pilots have to be trained to deal with those situations. They do so by practicing acquiring skills one at a time and then combining them; and they often unintentionally crash the plane in simulation during this learning-by-doing. They do not learn or practice with a new airplane full of people. The RIT Decision Cases train participants to practice dealing with market or economy dynamics, including potential crises and other complex events, as well as the endogenous uncertainty introduced by the market participants themselves. They practice these skills before proceeding to advise clients about risk and opportunities associated with the assigned task(s). In addition, management problems themselves are dynamic and often novel. Simulation-based learning can contribute to learning 'how-to-learn', a skill that is conducive to life-long learning and innovation.

Another useful analogy with a flight simulator is the automation required to deal with 'big data' and fast decision making. The RIT package has built-in 'Application Program Interfaces' (APIs) that can be turned on for any case so that participants can practice writing simple scripts to implement some decisions automatically. Automated systems can be efficient but fragile which is why the pilot's role in monitoring the cockpit instruments and skill in responding quickly to problems is so crucial. That is why sequencing the RIT cases such that participants make the decisions manually first (practicing at lower speeds and incrementally) so that they fully understand the scope of potential outcomes before writing an algorithm and implementing strategies automatically.

The remaining sections of this paper are organized as follows: Section 2 summarizes the structure of the RIT Server Application; Section 3 reviews some features of the RIT Client Application as well as available real-time links with support models; Section 4 discusses some examples of the RIT Decision Cases; Section 5 provides some examples of applications for research, competitions and events; and Section 6 concludes.

2. Structure and features of the RIT server

2.1. Server structure

The RIT package is currently available for the Microsoft Windows OS. The RIT Server application operates a **matching engine** that continuously accepts and matches limit and market orders from market participants and computer-generated AI (Artificial Intelligence) order flow. These orders are segregated based on their association with different securities, for example: multiple stocks, options, futures or bonds. Further segmentation can be achieved by creating different marketplaces for the same security, for example: a single stock trading on two different exchanges. Each security market and marketplace can be individually parameterized to create differences in fees, tick sizes and market permissions to name a few. In terms of robustness, the matching engine can accept and clear over 50,000 orders per second using a modern personal computer.

The matching engine operates as **a continuous double auction** and follows a price–time priority ruleset. Order types are limited to market and limit orders, and limit orders are treated (partially or completely) as marketable if the limit-order price crosses existing orders on the other side of the order book. **Market clearing prices** are established on a per marketplace basis, so there is no aggregate order book generated by the server. This creates requirements for participants to efficiently route their orders (in the case of a security trading on multiple market places), as poor execution will cause price distortions and create arbitrage opportunities.[3] The RIT markets also allow for "upstairs market" trading where participants can directly submit negotiated trades that fit within market price constraints (i.e. blocks must trade within 5% of National Best Bid Offer). All back-office functionality (clearing and settlement) is done at the time of the trade by the server since all funds and securities are virtual.

The RIT server currently **allows equities, fixed income securities, physical commodities, foreign exchange, options and futures** to be transacted. Of course, synthetic products can also be created by combining existing instruments. All of these securities can be denoted in any currency.

Information can be distributed to market participants publicly or privately in plain text or HTML format. Information content and arrival times can be customized (and/or randomized) to create asymmetry across market participants. Participants can also be queried and required to provide information,

[3] Participants can build their own aggregate order book using the RTD (real-time data) links to the alternative markets. A start-up decision-support model does this for the market microstructure cases.

feedback, and/or answer questions in real-time while the markets are running.

All information and actions available to market participants can be controlled by the server on an aggregate or individual level. For example, limit-order book "level 2" or time-and-sales data can be selectively hidden from market participants. Individual markets can be halted and resumed, slowed down or sped up, and securities can pay dividends, be settled, expire or be liquidated.

2.2. Defining participant roles

2.2.1. Optional AI programmed order flow

One of the most innovative features of the RIT platform is the **AI Order Flow** which can be programmed as uninformed (liquidity or noise order flow) or, alternatively, as informed order flow. N orders are generated per minute, each order has a price that is Normally distributed around the midpoint[4] of the bid and ask prices at the top of the limit-order book (hereafter referred to as the midmarket price). The Normally distributed prices are rounded to the nearest tick, based on a specified tick-size (i.e. nearest penny). The order is determined to be a buy order or a sell order based on an equal probability draw. All orders are generated as limit orders, however, when a buy order is generated with a price that is above the current best ask price, the matching engine treats it as a marketable limit order. As a result: roughly 25% of orders are limit orders to buy, adding liquidity to the market; 25% are marketable limit orders to buy, removing liquidity from the market; 25% of orders are limit orders to sell; and 25% of orders are marketable limit orders to sell.

Some benefits of the liquidity or noise order flow are as follows:

(1) In class settings with few participants ($n<10$), human traders typically lack the trading frequency to facilitate an orderly marketplace. The AI liquidity order flow can act as price-agnostic market makers who absorb market orders or trade against limit orders and hold temporary inventory positions until they can be transferred from one human participant to another.
(2) The previous point is particularly important for participants who are practicing in preparation for a class or event. The RIT cases can be set to run 24/7 with remote access so those practicing may be accessing the market at times when there are few human participants.
(3) The AI order flow allows an instructor to predetermine a certain level of market liquidity, instead of relying solely on human participants. This can be used to illustrate the difference between very liquid or illiquid markets and can be changed dynamically.
(4) In class settings with few or many participants, the AI order flow parameterized as noise traders provides opportunities for informed human traders to generate trading profits, even if the human traders hold the same information. Without noise traders, human participants who possess the same information will never trade with one another. This is the most important feature of noise orders because it allows the administrator to use the trading results to distinguish skill. The trading results require some market participants to systematically lose money in order to provide profits to the informed and skilled participants.
(5) Finally, for participants at the introductory level, the market activity generated by the AI order flow creates an environment that is less intimidating to enter.

[4] If no midpoint exists, the last traded price, or the parameterized starting price is used instead.

As opposed to liquidity AI Order Flow, one can also utilize **Informed AI Order Flow**. In this case, N orders are generated per minute, each order has a price that is Normally distributed around the current midmarket price. These orders are programmed to be informed in the following sense. The current midmarket price is compared to a predetermined P^* "informationally efficient" price, and the direction of a programmed order is determined based on whether the current midmarket price is below or above the P^* price. If the market is currently below the P^* price, the programmed order will be a buy order, otherwise it is a sell order. All AI orders are generated as limit orders, however, when a buy order is generated with a price that is above the current best ask price, the matching engine treats it as a marketable limit order. As a result, when the market is currently below the P^* price, roughly 50% of orders are limit orders to buy, adding liquidity to the market, and 50% are marketable limit orders to buy, removing liquidity from the market. Vice versa if the market is above the current P^* price.

The **AI Informed Order Flow** provides many customizable benefits, including:

(1) Allowing the market to be informationally efficient even when human agents act irrationally.
(2) Creating competitive forces against human traders; this can be useful when there are a relatively small number of humans participating in the market.
(3) Creating uncertainty for human participants when they are trading with the AI order flow. The human trader does not know, ex-ante, whether or not she is trading with an informed or uninformed programmed participant (both are labeled with the same trader ID 'ANON' in the limit-order book).

The result of allowing both uninformed and informed AI order flow in the market is that the human participants have the potential to generate excess profits, but cannot do so with impunity because they may be trading against informed traders at any time. They also cannot easily observe or "follow" the actions from the informed AI order flow because it is difficult to distinguish noise from informed trades. This creates a semi-informationally efficient market where market participants' actions, both programmed and human, are constantly being obfuscated. Put another way, a very realistic marketplace environment in which to practice and learn.

Consider the following example of how the Informed and Uninformed AI Order Flow generated by the computerized agents can interact with and affect the market prices. Suppose that there is a stock that is initialized with information that it will pay a liquidating dividend of V as its only cash flow and that the risk free rate is zero. The P^* price is parameterized to begin at V so that the informed computerized traders are aware that the fair value of the asset is V. At t_s, the case is programmed to announce to all human traders that the liquidating dividend has been increased by B to V+B. The P^* price is programmed to increase by B for $t > t_s$. In other words, as soon as the news is released, P^* increases by B. Informed AI Order Flow will now observe the market price of P being considerably below P^*, and react by placing a significant number of buy orders, thus causing observed market prices to increase. Uninformed AI Order Flow will continue to trade in a manner that is completely ignorant of P^*, i.e. they will randomly buy and sell.

To make the computerized traders partially-informed, the case could adjust P^* by V+B+e where e is a noisy error term, centered on zero. The computerized traders will now react to the news in a correct, but imprecise manner. They may under or over react to the result, even though on average they are correct.

2.2.2. Human participants

Human participants can be defined with both common and role-specific permissions that determine the information that they receive (both news and price data), the securities (and/or physical infrastructure) to which they have access, and the trading capital that is available to them.

The most common application of human participants' roles is to create homogeneous roles (that is, all humans have the exact same parameters) and allow for their individual analyses, utility, and skill to create dispersion. In theory, if all participants have the same utility function, information and skill, no trades would occur. However, even with homogeneous roles and common information, heterogeneous skill levels can result in trading activity.

To generate an even more dynamic marketplace, we can force dispersion in the participants' actions by setting up a non-homogeneous environment. The administrator can generate capital differences, trading constraints, and information asymmetry to further accentuate disagreement and incentivize more trading to occur. Note that, since participants are not exogenous to the market, **endogenous uncertainty** can be generated by trader activity and the resultant behavioral effects studied by **behavioral economics and finance** can occur in relevant situations.

2.3. Administering the market

2.3.1. Adaptive control of the market

With respect to **running the market**, the market begins in an offline-state while the market administrator changes parameters or case-specific variables and checks that all participants are linked to the market. On their RIT Client App, participants will be able to observe which case is loaded and set up their monitoring and decision interface accordingly (see further descriptions in Section 3 below).

Once the market is started, orders can be submitted and potentially matched, markets clear, and the results – including information flows – are revealed to participants through their RIT Client modules. While online, the market administrator can pause and subsequently resume the market. In addition, the administrator can adjust the speed of the market in real time. Lastly, the server can be run 24/7 on an "automatic reset" mode where a simulation replication is run, and then, after a pause, the market automatically resets by reloading the case with new random seeds and runs again. This allows for a server to be left on in an un-administered state but participants can login and "practice" at their convenience. The practice mode allows for students to have considerably more exposure (contact-time) with a fully functioning market. The enables students to "learn at their own pace" either prior to, or after, the traditional classroom experience.

There are many implications for learning associated with adaptive control of the market. While seemingly trivial on the surface, the ability to pause an active market is crucial for classroom learning. This feature allows the instructor to completely halt the market and take time (and have the undivided attention of the students) to discuss current market observations without being concerned with data and parameters changing. An extension of this is the ability to slow down a case. This allows participants to be more thoughtful about their decision making because the pressure of a "fast moving market" is alleviated. In the authors' experience, it is common to run the market at half (50%) speed and then progress to 100% speed as students master the skills required by the simulation.

2.3.2. Real-time identification of participants and monitoring

Fig. 1 illustrates one example configuration of the RIT Server interface. Note that modules can be opened or closed quickly as one sequences through the various roles of the instructor: loading a decision case, setting parameters, checking market participants' links, adjusting market speed, monitoring the market and participants' performance, providing feedback, saving results, posting results, etc.

All trade, position, and profit & loss data are available in real-time to the instructor. This is crucial because it allows the instructor to identify and **monitor** students who are struggling (or excelling) and address their needs in an immediate manner. These data can also be broadcast to participants so that they can see their performance relative to their peers and calibrate their strategies accordingly. Displays in the 'Monitoring' module allow one to match real names with 'Trader Id' or optionally hide the real names when posting results.

With respect to **saving the results**, all trade, position, and profit and loss (P&L) data can be exported to a high-frequency excel file that timestamps all participant actions. Most commonly, the aggregated P&L data are exported and used to illustrate performance to a class of participants. However, the entire trading histories can be rebuilt and replayed for the class using the saved detailed data.

After every replication of a case, individual-specific **performance reports** are distributed to all participants. These reports provide detailed time-series charts of their actions relative to the market dynamics, as well as tabular summaries of their profit & loss resulting from those actions. The PDF format of these performance reports is designed to be easy to read and interpret; as compared to the high-frequency data (saved in Excel) which are complete but more cumbersome to generate meaningful reports that can be consumed easily after each replication of the case.

There are many **benefits for learning associated with timely and useful feedback**. Creating an engaging and challenging simulation that focuses on the most important components of a required decision provides the basis of useful hands-on learning. Equally important is timely feedback concerning the outcomes of decisions. In the RIT cases, the market aggregates participants' decisions and provides immediate feedback.

Being able to report the sequence of their actions in a way that allows participants to assess whether or not their decisions were correct is extremely important for the learning process. Simply showing aggregate P&L data is an extremely blunt tool to understand whether a participant truly understood the subject matter. Making volumes of high-frequency data available to participants creates the potential for deep and thoughtful analytics; but those data are often too detailed to be useful between case replications which sometimes results in the feedback being ignored. The RIT performance reports strike a balance by presenting time series plots of the relevant decisions (for example, Fig. 5) and tabular summary data of results (for example, Fig. 6) in a way that can be easily interpreted and discussed. As a result, participants can learn from their mistakes and adjust their strategies in follow-up replications of the case.

Examples of non-P&L data that can be reported using the granular and high-frequency data would be: Value-At-Risk for a given student's portfolio if the objective is to manage the risk of a portfolio, Options pay-off graphs displaying the possible outcomes for a student who is building a hedge, or arbitrage vs. speculative profits in a case where the objective is generate arbitrage profits.

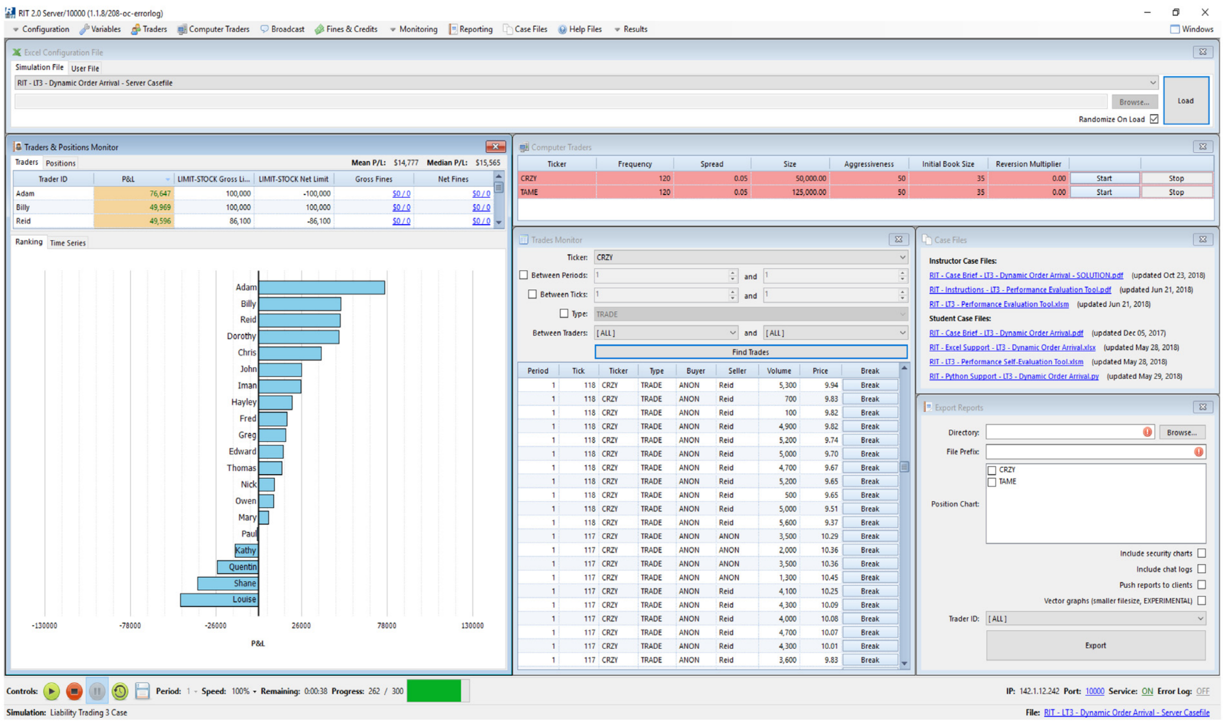

Fig. 1. Sample configuration of the RIT server interface.

3. Structure and features of the RIT client and decision-support models

3.1. Client features

The RIT Server Application is analogous to a flight simulator machine that can be programmed to deliver practice for the pilot to acquire skills. The RIT Client Application is analogous to the cockpit displays that allow monitoring of the environment and provides instruments for inputting decisions. An important feature of the RIT Client is the fact that the case information and support files are dynamically integrated into the user interface. This means that the student can readily access help documents, case documentation, decision-support models, or other attachments any time a case is running. This seems trivial, but it ensures that the student is never in a situation where they are participating in a simulation "without access to the required information". In addition, having these documents integrated into the software is far more efficient than requiring the students to download them from an external website, Learning Management System, or network drive.

The RIT Client features a **customizable modular decision space** where the student can open, move, and resize RIT Client modules[5] anywhere on their screen. Each module serves a different purpose, for example, the 'Charting' module provides real-time charts of variables such as security prices, combinations of security prices such as spreads, P&L, etc.; and the 'Time and Sales' module provides all transactions that have occurred for a specific security. Allowing the student to show, hide, and move these modules serves multiple learning and efficiency purposes. Fig. 2 illustrates just one of many possible configurations of the RIT Client interface. Multiple screens allows participants to configure the RIT Client interface on one screen with a linked real-time decision-support model on another screen.

[5] These RIT Client modules include: Portfolio, Order Entry, Market Depth, Trade Blotter, Transactions Log, Assets, News, Trader Info, Time & Sales, Charting, Chat, Kill, Case Files, and Help Files.

(1) Useful modules vary across specific RIT Decision Cases. For example, modules relevant for physical assets, such as pipelines or refineries, are unlikely to be relevant for cases that only include decisions about financial securities.
(2) A customizable RIT Client interface makes the learning curve more gradual for students. By starting with a handful of information and decision-input modules available at any given time, the relevant information that they need is readily at their disposal instead of being hidden across a plethora of data. This allows them to focus on processing the data and making decisions, instead of spending time finding the data.
(3) Individuals learn at different paces and analyze data in different ways. Some may want to add many extra modules showing very minute details, whereas others may prefer to keep the displayed data to a minimum. The "one size fits all" model of displaying data does not match the needs of a diverse user base.
(4) Professional industry-standard applications typically use a modular layout due to the variation in roles and personal preferences of those using the same software application. The RIT Client also being modular is simply realistic and introduces students to what they can expect if they enter the securities trading industry.

The limit-order book can displayed either in ladder or book format or both. Fast order entry can be implemented with preset volume and price offset (improvement) by clicking on orders in the book. This feature is available for limit orders, market orders and sweep orders, and is particularly useful for tasks like market making. Alternatively, individual order entry screens can be toggled to be market or limit orders. As described below, orders can also be implemented automatically using algorithms which access the simulated market using alternative application program interfaces (APIs).

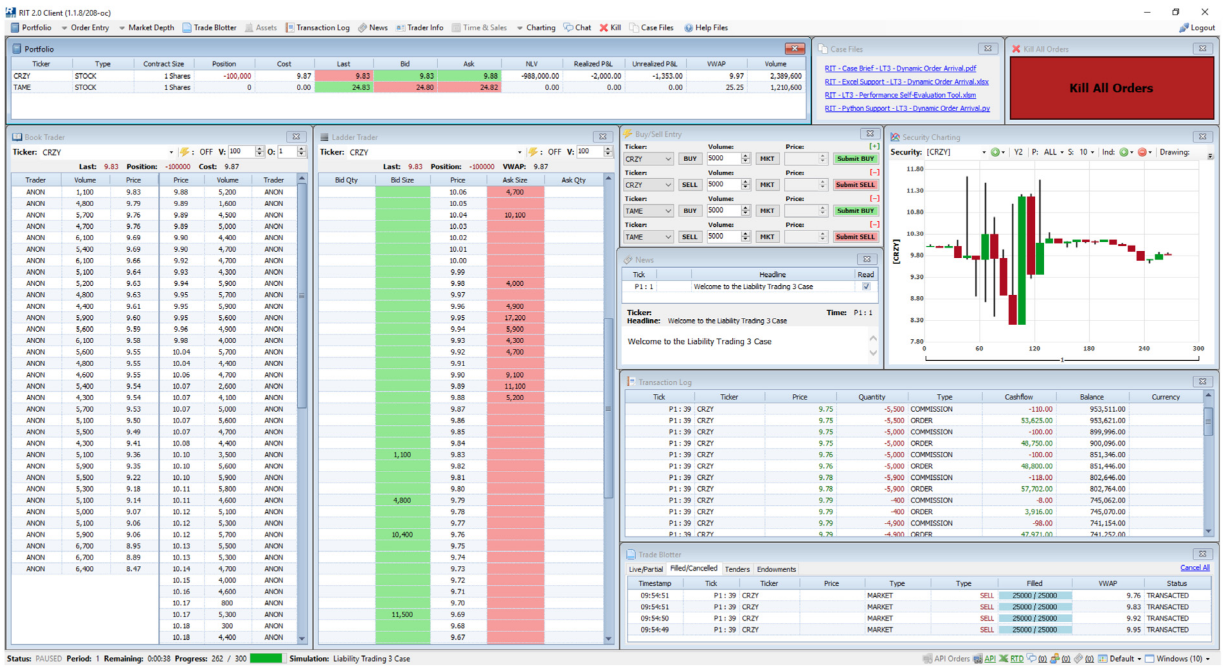

Fig. 2. Sample configuration of the RIT client interface.

3.2. Data interfacing between the RIT client and decision-support models

The RIT Client has three alternative data interfaces, and these are also scaled in terms of difficulty to facilitate a gentle but robust learning curve. **Real-Time Data (RTD) links** are available in order to drag-and-drop nearly all quantitative data elements directly from the RIT Client into Excel. This will create a data link that updates in real-time, and students can then use Excel formulas that they are already familiar with to manipulate the data. Students can also learn how to use the RTD Function Syntax to construct dynamic RTD links so that they can quickly and efficiently extract large amounts of data from the RIT Client for analytical purposes. This method of data analysis is extremely accessible given that it is based on drag-and-drop functionality augmented by Excel formulas. The RTD Links are uni-directional and can only facilitate data being exported from the RIT Client into Excel (i.e. trades cannot be submitted from Excel to the RIT client via RTD). Fig. 3 provides an example of a decision support template for a particular RIT Decision Case utilizing RTD links to the simulated market. Participants can build out the start-up templates by adding their own programming, for example, to include flashing signals for actions when a security is mispriced in the market.

Students who master the RTD functions can then use an Excel VBA COM Object which creates custom VBA functions that can be utilized to submit data or trade requests to the RIT Client. As a result, this data link is bi-directional. Initializing the VBA code (functions or subroutines) and submitting trade requests is considerably more difficult than simply requesting data via RTD, so it provides students with a sequential challenge to their learning. Since the **VBA API** is built into Excel, students can still rely on Excel's graphical user interface and cell-based functions to perform most of the calculations, thus making the programming requirements less onerous.

In addition, a **REST (Representational State Transfer) API**, built on industry-standard protocols, is available. It can accept queries from nearly any programming language. The REST API is a very robust bi-directional messaging, trade and data integration option reserved for students with significant coding experience. Most commonly, students would use Python or Java to submit trades via REST. These languages require a significant amount of coding overhead that Excel would typically automatically handle for the student. For context, programming a simple two-stock arbitrage algorithm in VBA requires about 12 lines of code, whereas using REST would typically require many more lines of code. Nevertheless, for robust modeling, the flexibility and speed of the REST interface is unmatched.

3.3. Some implications for learning associated with the RIT client

The RIT Client was designed with one primary principle: make the 'software' part of the learning process as frictionless as possible. The goal is to make the software simple and straightforward so that the student can focus on the content and learning objectives of each RIT Decision Case. In other words, the goal is for the student to think about their strategy, such as, "should I buy or sell this security" and not "what screens do I need to look at, and then what buttons do I need to push, in order to buy or sell this security". Further, as users become more experienced and obtain a level of proficiency with the RIT Client, they should not feel like the software is "holding them back", that is, they should be able to design and customize their client interface so that information and decision input options are available to them in whatever manner best suits their learning and analysis style.

The challenge was designing software that was user and learning friendly for someone who had 2 h experience with it, while making it robust enough for someone who had 200 h of experience using it. This design philosophy also applies to the student's journey through data integration (RTD, VBA API, REST API), as well as case design structure and sequencing of cases for skill acquisition.

4. RIT decision cases

4.1. Summary of available cases

The RIT market simulator package has more than 50 different RIT Decision Cases available for instructors to choose from when

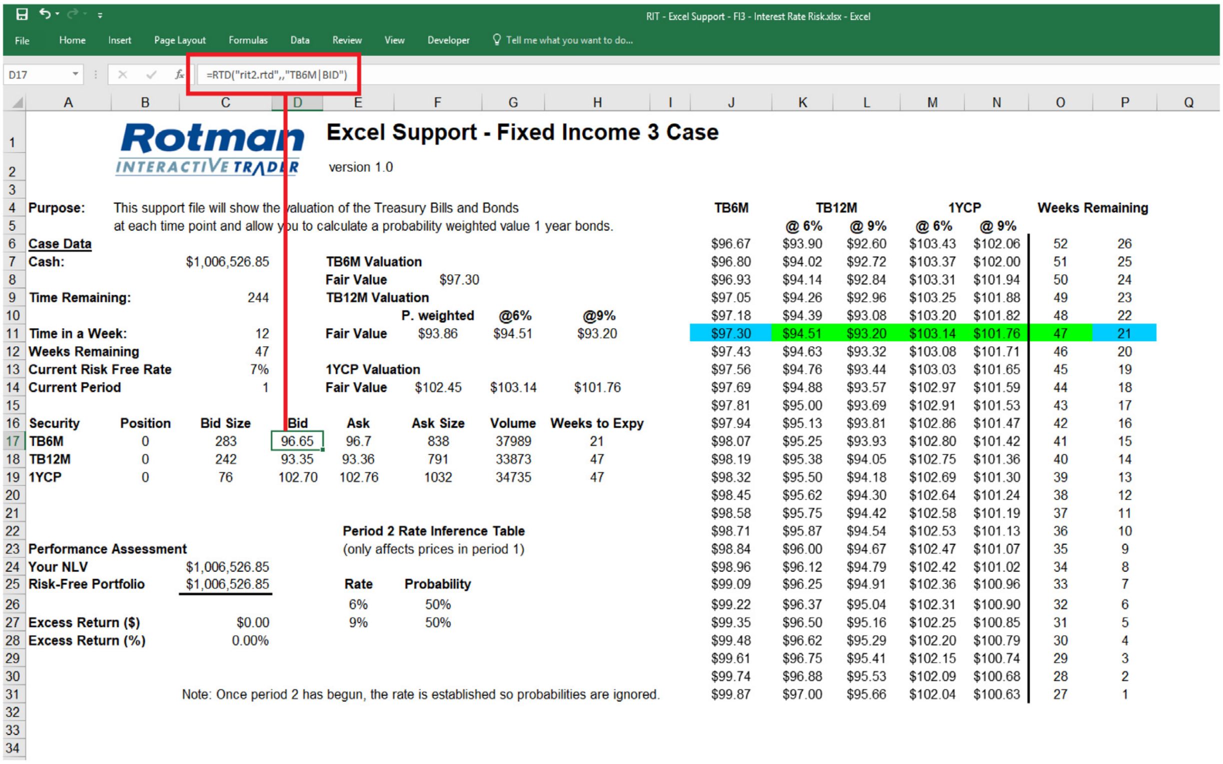

Fig. 3. Sample decision-support template with RTD links to the simulated market.

deciding what materials are most relevant to their learning objectives. These cases span the spectrum of securities including: bonds (corporate and sovereign), commodities (electricity, natural gas, oil, wheat, etc.), equities, ETFs, foreign currency, futures and options.

Using these securities, the cases mimic the decision space faced by professionals in many different roles including, but not limited to: Algorithmic Market Making and Arbitrage, Algorithmic Smart-Order Routers, Credit Analysts & Traders, Dealers, Equity Analysts & Traders, Pension Plan and other Portfolio Managers, Rates Analysts & Traders, Risk Managers including market risk, liquidity risk, etc., Commodity Speculators, Hedgers and Arbitragers.

4.2. Case design and calibration

Some cases are designed in such a way that students should apply specific **asset pricing models** to determine the *fair price* of the asset which they can then compare to the relevant market quotes, buying under-priced and selling over-priced securities. The inputs into their model are stochastic, so that each time the case is run, the results and correct actions will be different. In other words, the methodology of applying the solution to each case replication is the same, even the though the explicit solution will be different.

These cases are calibrated such that the AI order flow (and other, outside factors) purposely cause market mispricing, which generates opportunities for astute market participants to submit informed trades and accrue profits. Ultimately those with more accurate, better calibrated (more informed) asset pricing models, as well as better decision and execution skills, will generally be able generate higher returns from their decisions. This may not be the case for any particular replication, but we have calibrated the signal versus noise such that on average across multiple replications those with the best strategy will have the best performance.

As summarized in the summary of available cases in Section 4.1 above, many of the RIT Decision Cases feature learning objectives that use financial securities, such as derivatives, for strategies focused on **risk management or statistical arbitrage** rather than speculation. For example, cases that focus on real economy risks, agricultural crop hedging, commodity strategies (such as those related to oil, gas and electricity production, distribution and marketing objectives) and pension plan management are used to practice using financial securities to achieve those objectives.

For more advanced cases, the application of the relevant asset pricing models becomes considerably more complex, taking into account multiple asset correlations, risks, etc. An example, is the *RIT Fixed Income 7 Case* which combines dynamic yield curves with changing credit risks for corporate bonds. Less skilled participants may only be able to apply simple models to individual securities, whereas more experienced or more skilled participants may be able to identify more opportunities and manage more risks. Nevertheless, both can participate in the same markets, learning from their own level and from each other. This highlights the value of the cases for **individualized learning-by-doing**.

Inputs to markets are stochastic (exogenous uncertainty) and behavioral affects, including heterogeneous participants and model uncertainty, can generate endogenous uncertainty. Therefore, a participant's performance metrics such as P&L will be a function not only of that trader's skills, such as her informed trades, but also will be affected by market 'noise'. While the noise on a single replication may be large enough to overwhelm the signal generated by informed trades, cases are intended to be run over multiple replications such that the noise component becomes a smaller and smaller component of the participant's results. In other words, the **signal versus noise has been calibrated** such that on average across multiple replications those with the best strategy will have the best performance.[6]

[6] For example, in the RIT LT3 Case, the stochastic liquidity spread associated with tender offers is calibrated jointly with the market liquidity such that

4.3. Case sequencing

The RIT Decision Cases are intended to be deployed in a sequential manner, where each case builds upon the skills learned in the previous case. Following the same philosophy as the RIT Client software, data integration tools, and case design, the step-by-step sequences facilitate a gentle learning curve where participants can focus on mastering a sequence of learning objectives before being presented with more challenging content that combines exposure to multiple risks and opportunities.

An Example Case Sequence

Since the RIT market simulator package uses simulated markets to aggregate participants' decisions, a common case sequence with which to begin is a Market Microstructure series. The following is an example sequence.

(1) AT1 — This purpose of this case is to teach students how limit-order books work. Participants familiarize themselves with market orders, limit orders and the concept of market liquidity.
(2) AT2 — This case introduces the idea of short sales (negative positions in securities), and creates a market that is less liquid, so that students need to utilize limit orders. The price paths of the securities are not predictable over the day, so the objective of the agency trader is to fill customers' orders using volume-weighted or time-weighted average price (VWAP or TWAP) order-entry strategies.
(3) LT1 — This case teaches participants about the bid–ask spread, and the profits earned by market making (and implicitly, the costs of using market orders and paying the bid–ask spread).
(4) LT2 — This case requires participants to generate profits via the liquidity spread associated with a large block of shares. They must utilize their order execution (market and limit orders) skills to ensure they do not cause significant adverse price movements as they unwind their blocks.
(5) LT3 — Having mastered the skills practiced in the previous cases in this sequence, this case focuses on using links to a decision-support model that aggregates information in the limit-order books in order to quantify liquidity risk. The liability trader (LT) must make a decision as to whether the liquidity spread offered by the institution is adequate to compensate them for the liquidity risk and market risk they will face while unwinding the large block order. As such, this case combines model building and decision skills with their trade execution skills.
(6) LT4 — This case applies the skills acquired in the LT3 case to multiple exchanges. As such, participants need to build a 'global' order book and unwind the block trades across two exchanges. Given how difficult it is to manage liquidity and market risk in real time by manual order entry across more than one exchange trading the same security, while satisfying regulatory requirements such as fills at the NBBO (National Best Bid Offer), participants soon appreciate the value of building algorithms, such as a SOR (Smart Order Router), for automated order entry.

Having acquired the learning objectives and practiced the skills in the above sequence of cases, one can then move to the **sequence of ALGO cases** to learn how to translate their strategy to code for situations that require very fast decisions, such as, arbitrage (ALGO 1 Case), market making (ALGO 2 Case), smart order routing (ALGO 3 Case), etc. In fact, API order entry can be turned on for any RIT case, if appropriate, once the learning objectives associated with manual decision and execution have been mastered.

4.4. Feedback & performance evaluation: Implications for learning

The final step in a participant's learning-by-doing journey is their ability to easily understand, ex-post, the decisions that they made and the outcome of those decisions. Reliable, detailed, timely and easy to digest feedback and reporting is a crucial ingredient for this objective.

As discussed in Sections 2.3 and 2.3.2 above, the market administrator can observe the results of all market participants in real time. The market administrator can also export a microsecond timestamped report showing all actions taken by all participants, and also save the entire set of case parameters so that the exact same parameterized simulation could be run in the future.

A wealth of real-time information (including positions, P&L, etc.) is available to market participants on the RIT Client, allowing them to monitor their progress as they are participating in the markets. Fig. 4 provides one example that computes Value-at-Risk in real time to guide participants' risk taking decisions while managing a portfolio.

Following a case replication, market participants are given a detailed trading report in PDF format that visually shows their performance, as well as their trading actions through time, and a full attribution breakdown of their profit and loss across securities, as in Figs. 5 and 6.

In addition to the performance reporting options that are included in the RIT platform, some of the RIT cases include an instructor tool that generates custom-scripted reports to display very specific results tailor made to the case objectives. These instructor tools have been designed to decompose participants' performance (for example, Fig. 7) to illustrate their results as they relate to the case objective versus results associated with other strategies, such as, front-running or unwarranted speculation. These tools also allow instructors to penalize participants for inappropriate strategies or outcomes, for example: exceeding a chief risk officer's loss limit associated with their regulatory Value-at-Risk measure; or excess risk exposure associated with their delta hedging strategy. This feedback encourages participants to focus on the learning objectives of the case, and consequently to acquire the skills more quickly, allowing one to progress more effectively through the case sequence associated with a particular topic.

5. Examples of applications to research and events

Besides supporting teaching and training,[7] the RIT Decision Cases facilitate competitions and events at many different levels. Given the high degree of flexibility and customization that is available to the market administrator, low-friction engagement for participants, and high-resolution data logging, the RIT product is also a valuable resource for investigating research questions across a wide-range of topics.

In particular, the RIT platform has contributed to research publications in diverse subject areas including behavioral finance, experimental economics and trading. For example, Patterson (2014) and Nofsinger et al. (2018) studied behavioral effects in markets by utilizing the biometric readings of market participants. On the

participants who are managing the liquidity and market risk well will make a reasonable return on their bank's capital across a relatively small number of replications of the case.

7 In addition to applications for students, many financial institutions have utilized the RIT software as a training tool for their employees, either on site or as participants in academic executive programs.

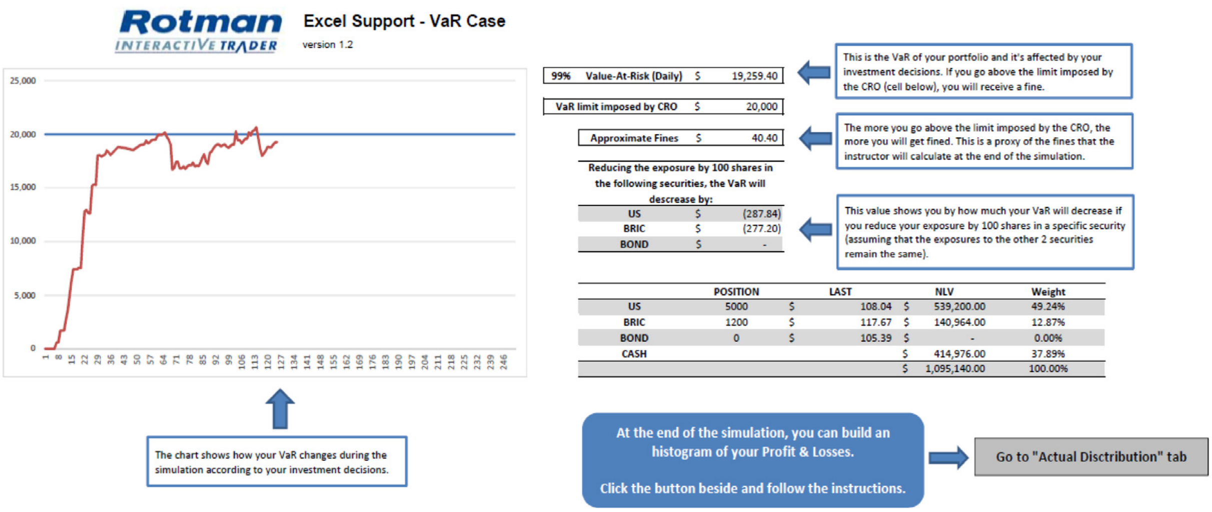

Fig. 4. Sample decision-support and monitoring template for a RIT value-at-risk case.

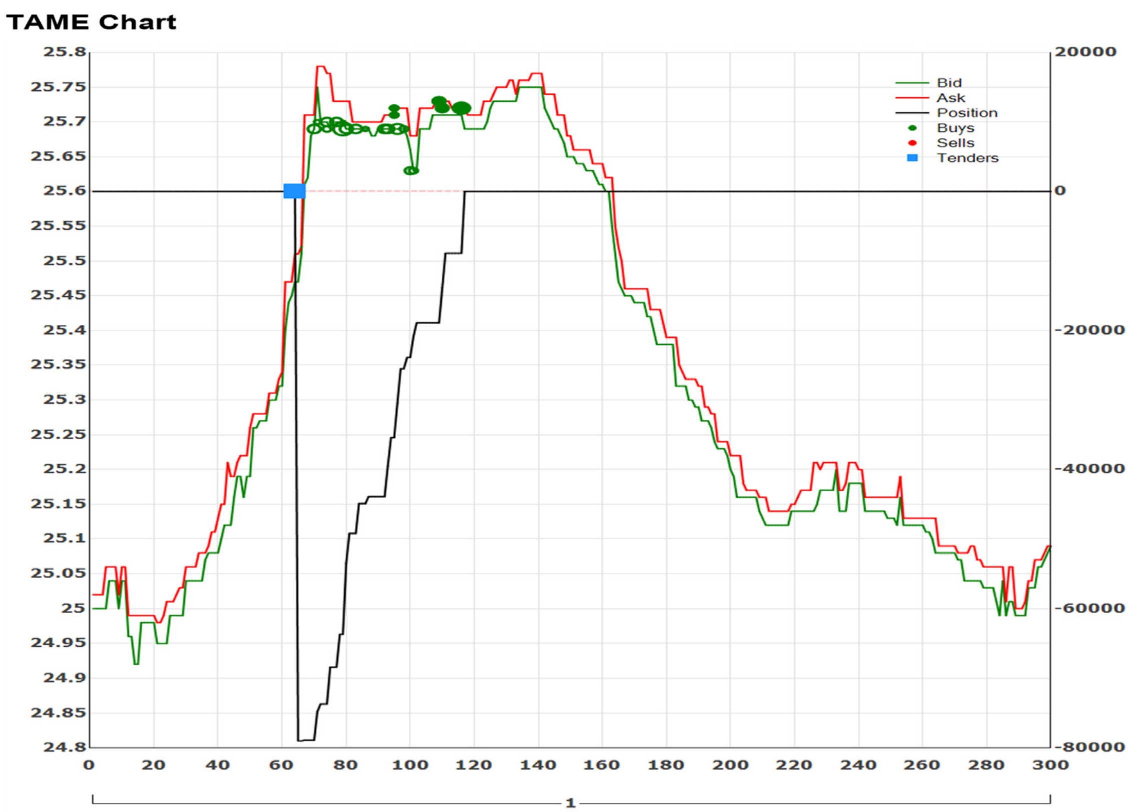

Fig. 5. Time-series of trades covering a tender offer in the liability trading 3 case.

other hand, Gould et al. (2010), Brousseau et al. (2014) and Glikstein and Kryzanowski (2017) have explored market efficiency, price discovery and volatility. The RIT platform and RIT Decision Cases have also been referenced by Latuszynska (2015) and Palan (2015) in their surveys of experimental software available to researchers.

In addition to testing specific research hypotheses, the RIT application can also be deployed to illustrate and extend existing experiments. For example, Brunnermeier and Morgan (2010) test experimentally the gains from waiting versus the risk of being preempted. This is a classic example of how asset bubbles can form, and persist, given certain circumstances. Brunnermeier and Morgan used a purpose-built web multi-user webpage to simulate their "market". A replica of their simulation parameters has been programmed for the RIT platform and applied to asset bubbles. The clock games experiment can be carried out with significantly less overhead using the RIT application. In addition, the user experience for test subjects would be better since the RIT platform has been optimized to reduce end-user frictions.

The RIT package can provide a strictly controlled environment and precise measurement of inputs and outputs which are fundamental for applications to research questions. The 'requirements for experimental asset market software' discussed in Palan (2015) are all features of the RIT package. These include his 'general requirements', for example: 'complete, time-stamped data record';

Tender Summary

Period	Tick	Type	Ticker	Volume	Price	BidOffer	Reserve
1	64	PRIVATE	TAME	-79,000.00	25.95	25.95	25.95
1	108	PRIVATE	CRZY	67,000.00	9.87	9.87	9.87
1	160	PRIVATE	CRZY	-73,000.00			10.30
1	208	PRIVATE	TAME	-56,000.00			25.07

Security P&L Summary

Ticker	Realized P&L	Unrealized P&L	Commissions	# of Trades	Buy VWAP	Sell VWAP	Buy Contracts	Sell Contracts
CRZY	8,587.00	0.00	1,340.00	18	9.8700	10.0182	67,000.00	67,000.00
TAME	18,434.00	0.00	1,580.00	19	25.6967	25.9500	79,000.00	79,000.00
TOTAL	27,021.00	0.00	2,920.00	37				

Fig. 6. Tabular performance report for the liability trading 3 case.

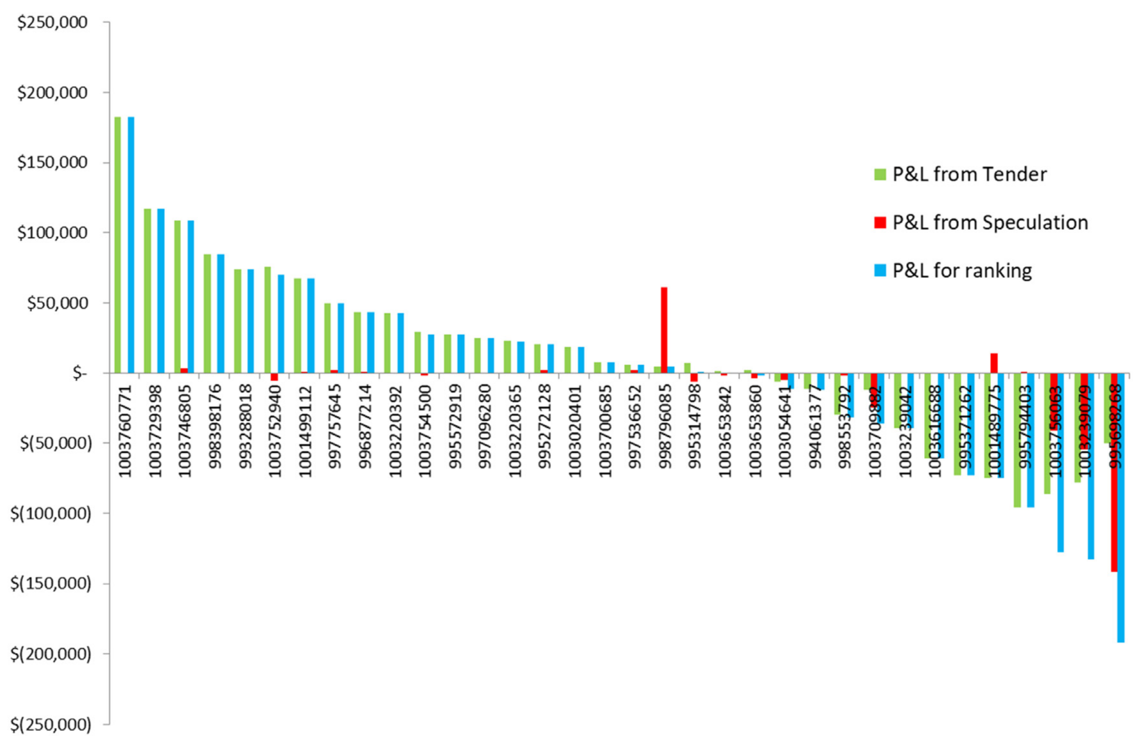

Fig. 7. Sample custom report decomposing participants' P & L.

'customizability and extensibility'; and 'reliability'; as well as his 'requirements regarding the market mechanism' [Palan (2015), section 1.2 and Table 1] including, for example: 'user-friendly interface'; 'choice between single and multi-unit trading with or without wealth transfer'; 'possibility to trade in multiple markets or over the counter'; 'parameter specification'; 'designated trader roles'; 'short selling'; 'order validation'; 'order types and priority'; and 'algorithmic trading'.

For example, given the access to a REST API to support decision-support modeling in RIT Decision Cases, combined with the 'complete time-stamped data record' at a millisecond frequency, one could use the RIT package to test various market microstructure features or design such as endogenous speed bumps.

Another example of how an experiment could be designed and carried out, would be a straightforward case involving the effects of "insider information" on market dynamics. Participants can be introduced to a case where asymmetric noisy information is released to all traders and that information is linked to final asset prices. Participants are happy to compete, in an attempt to generate profits when asset prices deviate from their perceived fair values (determined by the news). Total market liquidity can be measured based on the sum of trades from all participants. In subsequent iterations of the simulation, traders are made aware that some traders will receive the "inside information" prior to the rest of the population and their information will be perfect. The proportion of traders receiving insider information can increase over many iterations. Observing the volume of trade over the iterations with various treatments would provide insight into the ways markets may or may not be harmed by insider information. This is parameterized in the simulation by generating the final payout at the start of the case (but not revealing it), then generating news releases that are distributed based on that final payout with noise terms. Although the payout is "set" at the beginning of the case, that information is not revealed until the end, resulting in an experience where the payout is revealed over time.

Probably the most visible application of the RIT platform and RIT Decision Cases has been for special events or competitions. The Rotman International Trading Competition (http://ritc.rotman.utoronto.ca) attracts talented competitors every year from universities around the world; as do the regional competitions, such as the Rotman European Trading Competition (RETC) in Rome and the Rotman UNIST Trading Competition (RUTC) in South Korea. Furthermore, the adaptability of the RIT Decision Cases for participants with different backgrounds and levels of expertise, has resulted in frequent applications for recruiting and university advancement, student extra-curricular clubs and events, as well as course-based competitions for decision performance components of their course marks.

6. Concluding comments

As summarized above, the RIT product contains many performance feedback tools and reports which facilitate discussion, provide an opportunity for participants to adapt and improve their strategies after each replication, and allow instructors to use incentives to encourage participants to focus on the learning objectives of the case and consequently to accelerate learning.

In addition to providing case participants and instructors with the tools that they need to understand what is happening in real time, as well as upon completion of each replication of the simulated decision cases, it is even more important that the results are reflective of skill. The RIT cases have been calibrated, through extensive testing with a broad spectrum of participants, to deliver the mix of signal versus noise such that those participants who are pursuing good strategies will see their skill revealed in the results when averaged across multiple replications of a case.

Achieving the learning objectives associated with each RIT Decision Case reflects the culmination of careful design balance being applied at every level, from the trading interface, to the case calibration, case sequencing, and finally to the classroom or lab delivery, feedback and practice.

References

Brousseau, C., Gendron, M., Belanger, P., Coupland, J., 2014. Does fair value accounting contribute to market price volatility? An experimental approach. Account. Financ. 54, 1033–1061.

Brunnermeier, M.K., Morgan, J., 2010. Clock games: Theory and experiments. Games Econom. Behav. 68 (2), 352–550.

Cook, D.A., Hatala, R., Brydges, R., Zendejas, B., Szostek, J.H., Want, A.T., Erwin, P.J., Hamstra, S.J., 2011. Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. JAMA 306 (9), 978–988.

Fama, E.F., French, K.R., 2010. Luck versus skill in the cross-section of mutual fund returns. J. Finance 65 (5), 1915–1947.

Glikstein, E.D., Kryzanowski, L., 2017. Defeating the VPIN: Slipping into the noise. J. Trading 12 (3), 37–47.

Gould, S., Valenzuela, A., Kachersky, L., Holowczak, R., 2010. The behavioral dimensions of trading: Proximal and distal influences on performance. In: Campbell, M.C., Inman, J., Pieters, R. (Eds.), Advances in Consumer Research Vol. 37, 166-169. Association for Consumer Research, MN.

Issenberg, S.B., McGaghie, W.C., Petrusa, E.R., Lee, G.D., Scalese, R.J., 2005. Features and uses of high-fidelity medical simulations that lead to effective learning: A BEME systematic review. Med. Teach. 27 (1), 10–28.

Latuszynska, M., 2015. Experimental research in economics and computer simulation. In: Selected Issues in Experimental Economics, Proceedings of the 2015 Computational Methods in Experimental Economics (CMEE) Conference. pp. 151–169.

McCurdy, T.H., Woodhouse, R.A., 2002. Instructional innovation: experiential learning. Rotman Management Spring/Summer.

Nofsinger, J.R., Patterson, F.M., Shank, C.A., 2018. Decision-making, financial risk aversion, and behavioral biases: The role of testosterone and stress. Econ. Human Biol. 29, 1–16.

Palan, S., 2015. GIMS—Software for asset market experiments. J. Behav. Exp. Financ. 5, 1–14.

Patterson, F.M., 2014. The Relation of Steroid Hormones and Personality Factors To Financial Performance and Risk-Taking Behavior. PhD Dissertation. Florida International University.

Salas, E., Wildman, J.L., Piccolo, R.F., 2009. Using simulation-based training to enhance management education. Acad. Manage. Learn. Educ. 8 (4), 559–573.

Woodhouse, R.A., McCurdy, T.H., 2014. Innovation in management education: adding value with simulation based learning. Manuscript.