Brian Dutremble
CPSC 59700
Lewis University
Romeoville, IL, USA
Brian.dutremble@lewisu.edu

The Robots Can Have My Job:
Opportunities for Automation

Abstract

Many aspects of my current position are repetitive and tedious. Such tasks represent and opportunity for automation – particularly for robotic process automation. In this paper I discuss my role at my company and which elements of it are the best targets for automation. I also address the need for verification of the correctness of any programs that might be released into our system, as well as the benefits automated theorem proving might provide for CALM, our main data research tool.

Maintenance Contracts Analyst

In my position as a maintenance contracts operations analyst I support a renewal sales representative. Together, we provide customers with quotes for maintenance coverage on their existing assets. My tasks include refining assets records to ensure accurate quoting and generating ad hoc or push maintenance quotes for customers. In order to accomplish these tasks I utilize about a dozen different database access and quote manipulation systems. The quoting process consists of dozens of small tasks designed to produce a maintenance quote that is free of errors and accurately reflects the customer's current assets. This quoting

process is outlined in an eighteen-page SOP manual. It would obviously be ideal for the sales representative to be able to press a button and have a quote instantly generated rather than having to wait for what can sometimes be days or weeks for a finalized quote, so there is a big push to automate our processes as quickly as possible.

Despite the automation-ready nature of my group's processes there are things slowing the implementation of robotics. Compliance issues, a byzantine approvals process, and a general distrust of robotics must all be accounted for and overcome in the automation implementation process. These issues, coupled with the large amount of money at stake and the complexity and messiness of our databases, create an environment that isn't especially automation-friendly and make necessary a tremendous amount of research and testing before any sort of large-scale implementation can be considered. The automation implementation currently has about a two-year timeline due to these complexities. And, given the highly lucrative nature of our maintenance contracts renewals, there will likely continue to be significant human oversight once the robotics implementation is complete.

The majority of my time at work is spent verifying customers' assets and the relationships between those devices. Changes are likely to occur between renewals, particularly when it comes to the configurations for individual assets, and it is my responsibility to ensure that the customer is properly charged. Complicating this process are ancient systems that only sometimes talk to one another and the unlikelihood of point-of-entry data being correctly input when items are first purchased and installed. A great deal of detective work is required to determine the

assets' current configurations and original warranty information. While this aspect of the job might require judgment beyond the abilities of the current robotics program, I believe that this area is still represents the biggest opportunity for automation and that much of the need for complicated judgment decisions will be ameliorated by enhancements and automation that clean up the data before it reaches the quoting process.

The Quote

As I stated above, the final product that the employees in my group must generate for the sales representatives is a maintenance quote. A maintenance quote consists of the assets belonging to a customer and the cost for maintenance coverage on those assets. The quote must contain certain identifying information for the assets so that the sales representatives will be able to easily show the customer what exactly it is that the customer is paying for. Among the attributes listed on the quote output there are some that are most important to highlight for the purposes of discussing automation in our group.

Among the items on the output that are automatically generated by the system and are generally immutable are Serial Number, which is a unique identifier that any piece of hardware and certain instances of software has, Model Number and Model Description give the customer some indication of what it is that they're paying for maintenance coverage on, and the annual cost for maintenance coverage for that asset.

The other elements of the quote are either modified to the representative's specifications, as in the case of Service Level and Discount, or a result of the links between the assets on the quote and their sales orders and related assets. The HW Identifier field, for instance, is used to show to which top-level hardware a piece of software or dependent hardware belongs. Ensuring that the information in these fields is correct can involve hours of research related to the history and links of the assets. The Contract and Coverage Dates fields can also require some tedious work to determine just what coverage a customer was originally entitled to and whether the assets somehow ended up on multiple contracts or in multiple locations. Our system's data is far from clean and a necessary component of any sort of automation would be the ability to clean up the data in as part of its process.

Verifying asset quantities is another important and time-consuming part of our quoting process. We refer to this aspect of our process as "configuring" a system and it requires that we ensure that the quantities in our system accurately reflect those of the customer's assets. To do this, we have to utilize a different system for each product line and only a few of those systems have outputs that are easily read. Most of the product systems require extensive guides for their interpretation and entry into our proprietary systems.

Time-consuming processes like ascertaining the relationships between software and hardware and verifying asset quantities are aspects of our process that would be fairly easily automated. Entering the results of our research also takes a great deal of time and such mundane data entry should absolutely be on the list of things to be automated first. These especially tedious tasks make up the bulk of our

work and are the biggest reasons for delays in the quoting process. Also, our main composite research tool, CALM, is not as agile as we would like, taking an entire day to register changes, and it is my hope that automation could also be used to speed up its refresh rate.

Though some of our quoting and cleanup processes are currently out of the reach of the robots, we have already seen automation take over portions of our workload. For example, our EverFresh program uses robotic process automation to generate exact copies of the last year's renewal, saving countless hours that would have otherwise been spent creating new versions of what can be dozens of contracts for each customer. Additionally, contracts under a certain dollar value are also handled by automation. Both of these instances of automation have been successful, though not without occasional hiccups and growing pains. The main method of automating our processes will continue to be robotic process automation (RPA) for the foreseeable future.

Robotic Process Automation

Robotic process automation (RPA), a term coined by the company Blue Prism, refers to software robots that replicate the action of a human being interacting with a software application. The technology captures all of the essential keystrokes, mouse clicks, and other inputs from the human and is trained to perform tasks just as a human would, but without any mistakes. In a shared services model, like that of my company, RPA can really shine. It is great for "swivel chair" processes, which are processes where human workers take data from one set of

systems, process those data according to a set of rules, then enter the results into another set of systems [1]. My current position consists almost entirely of swivel chair tasks, as I receive a request or other information from a sales representative or other system prompt, research and modify that information, then send the new enter the new information into other systems and send a newly created quote to the sales representative. While RPA is currently best suited for repetitive tasks that affect a predictable stream of data, it is constantly improving, becoming capable of adapting and completing tasks of increasing complexity.

Benefits of RPA

RPA has many benefits for companies choosing to employ it. Chief among these benefits is cost savings: robots can take the place of many humans and are thus extremely cost effective. Beyond the cost savings, software robots have many other advantages over human labor. Robots, for example, do not need to rest and do not need to stop and think about what to do next. This decreases cycle times and improves throughput. RPA also offers increased scalability, as once a process is defined as a series of instructions it can be completed by as many robots as are needed on whatever schedule is desired. RPA is more flexible than humans as its robots are often capable of performing many types of processes and can be reassigned as needs shift [2].

The non-invasive nature of RPA is another significant benefit. Companies have invested an enormous amount of time and money in their existing systems and are understandably reluctant to employ a strategy that involves further developing

them or replacing them. RPA interacts with the existing systems just as human workers do, accessing them through the presentation layer and avoiding touching any underlying systems programming logic.

Improved accuracy is another benefit of utilizing RPA. Even the most experienced human workers make mistakes. If given the proper instructions, robots will not. They follow the rules exactly, always generating the correct results. Completely following the rules also allows for an accurate audit trail that enables further process improvement and easier regulatory compliance. Lastly, the workers whose jobs are not replaced by RPA often experience an increase in morale, as the tasks that are most likely to be automated are those that are tedious and least enjoyable. RPA frees up employees to focus on higher value, more judgment-intensive tasks [2].

Intelligent Automation

Judgment-heavy tasks won't remain safe from robotic process automation for long, as artificial intelligence is already beginning to make it possible to automate non-routine tasks (those that involve judgment, creativity, intuition, or other skills once thought to be beyond the reach of software robots). The use of artificial intelligence, analytics, and robotic automation together is known as Intelligent Automation (IA). Decreases in the cost of processing power and data storage have accelerated development in the field of AI and enabled technologies with capabilities including natural language processing and handwriting recognition.

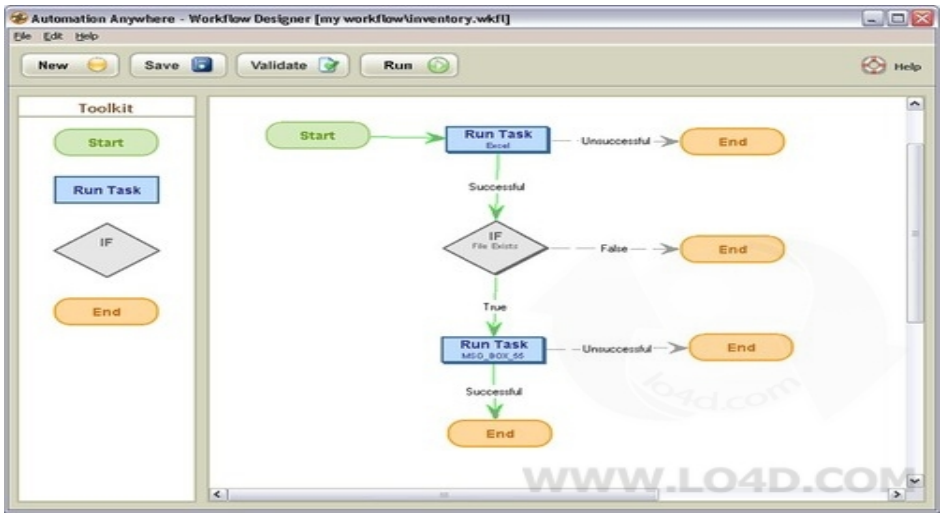
Intelligent automation is already being used in many industries. Banks are using it to improve their ability to adhere to regulations by monitoring all electronic communications while wealth management funds use it to analyze portfolio data and generate reports for their customers. IA is much more useful than RPA would be in these kinds of situations as it has the advantage of being able to come to conclusions rather than just follow instructions. It is, however, more expensive than RPA and narrower in its purpose, and an IA solution also generally takes longer to implement.

Current State of RPA

While intelligent automation will very likely consume more and more of the enterprise automation market as it becomes more cost effective, robotic process automation is currently the method of choice for organizations seeking to automate as many of their tasks as possible. Some large organizations employ hundreds of robots to achieve their automation goals and have greatly reduced the cost of their operations. There are many companies that provide RPA services and while their terminology varies, they all utilize the same three core elements: a set of developer tools, a robot controller, and software robots.

Developer tools are used to write instructions for the bots of the RPA solution to follow. The tools are designed to be used by people who have no coding experience, often featuring drag-and-drop functionality or configuration wizards that automatically generate code that will govern the behavior of the bots. Despite the simplicity of the developer tools, the instructions that need to be provided for

the robots can be very complex, so many kits feature an interaction diagram like the one below:



Automation Anywhere, one of the simplest RPA options, is the RPA software that my group's robotics team is using to automate my team's processes. The core feature of its developer kit is its process recorder. The process recorder captures user actions and creates a repeatable process (this includes everything from keystrokes to mouse movement). Automation Anywhere allows for modifications to the generated processes like building in time for system lag and the removal of steps deemed unnecessary. The resulting refined process is much more efficient and thus drastically increases throughput.

The robot controller has three main functions in robotic process automation. First, it provides a repository for all of the instructions that have been created for the robots. Often included in the repository are all of the credentials needed for applications that need to be accessed. Second, the robot controller provides controls and workflow guides for users. The controller guides users through the entire process, from the initial creation of the robots through deployment. Lastly, the robot

controller schedules and assigns jobs. This function increases in importance as the numbers of robots and processes grow.

The third element of robotic process automation consists of the software robots that interact directly with the systems. The robots are versatile, tireless entities that act out the instructions created with the developer tools when the robot controller activates them.

Robotic process automation in its current form is adequate for the current needs of our company and is a good first step in the long-term automation plan. The visibility of RPA's impact on our systems can already be seen and its effectiveness is winning people over, combating the skepticism that many have toward new things (particularly toward technology-related things). As employees become more comfortable with robots taking over tedious tasks they will be more likely to welcome the more advanced automation that artificial intelligence can make possible.

Artificial Intelligence

There has been rapid progress in the fields related to artificial intelligence (AI), machine learning, and connected devices in recent years and these technologies will undoubtedly transform our world, irreparably disrupting some fields while vastly improving our abilities in others. The extent to which these devices and intelligences should play a role in our lives is hotly debated, with some arguing that we should drastically slow the pace of research in the fields while others say (particularly in the case of AI) that it should cease altogether, and still

others believe that we should increase pour even more resources into the effort to make ours a more connected and intelligent world. There does seem to be a consensus, however, when it comes to our general approach to these technologies: we should exercise care. Each of our decisions regarding AI and networked devices will have far-reaching impacts and if we are to benefit from these impacts, we must embrace the potential while acknowledging the risks and making important decisions regarding responsibility in these fields. My company is quite a ways off from being able to utilize any sort of sophisticated artificial intelligence to manage its data infrastructure but it does not take much of an imagination to be able to envision the path from its current RPA efforts to a work environment into which AI is fully integrated.

AI Safety

While AI technologies will very likely have a largely positive impact on the world, great care must be taken to ensure that some of their much less appealing impacts will not also manifest, and this is certainly the case in the shared services group at DellEMC. We depend on such innovation to keep us relevant and effective, but our robotics team is often extremely cautious when implementing new technologies because any errors can result in a poor customer experience and a potential loss of millions of dollars. This caution would be especially important with powerful AI tools. To avoid accidents, we must anticipate complications prior to the implementation of an AI and set parameters that compensate for those potential

accidents. In the following sections I will explore ways in which we can attempt to avoid accidents.

Avoiding Negative Byproducts

An AI will do whatever it can to accomplish its goals within the parameters it is given and without any special regard for its environment. For this reason, we must give the AI the ability to ascertain whether its actions will produce favorable outcomes. If an AI can evaluate the potential future states of an environment, it can minimize its negative impact on that environment. To be able to evaluate its impacts correctly, the AI must be given a base set of acceptable interactions (and unacceptable interactions) within the environment then regularize actions as it is rewarded for them. Employing a reward structure that incentivizes an AI to minimize its impact on the environment in which it is completing its tasks might be a good way of limiting negative side effects of an AI's behavior, for example. Given the interconnected nature of our company's databases and systems it's absolutely essential that an AI minimize its impact on the system, as modifying a record in one system can have far-reaching consequences in others.

Reward Structures

The way in which a reward system is structured can have a large impact on the performance of an AI. Rewards must be designed with every complexity of a system in mind so that the AI will fully accomplish its goals. Many tactics can be employed to create a proper set of parameters including formal verification and

practical testing of parts of the system, adversarial techniques intended to blind the agent to some parts of the system, or the installation of traps that a correctly functioning agent would not fall into [3]. Given the intricacy of the renewals maintenance database environment, developing a reward structure will be problematic, but doing so is essential in order to prevent an AI from negatively impacting the our data and that of other teams.

Oversight

As the role of AI at our company expands it will likely not be feasible for every AI in operation to be overseen by a human, so methods by which an AI can be given feedback on its own progress must be developed. Most strategies for moving toward a type of reinforcement learning that requires less human intervention involve designing an AI that only ask for status updates occasionally. When given information about its progress, the AI can infer applications for that information beyond the narrow portion of the process to which it directly applies. If the AI is able to identify methods that predict a reward, it will then it can learn the conditions under which those methods are valid, lessening the need for active supervision [3]. For instance, if an AI knows that a quote output with software locations that match the locations of the related hardware, it might be able to complete research and execute background decision making that would ensure that the locations are correct without every step of that process being part of its directives.

A hierarchical approach might be the most promising strategy. In such an approach, an AI would delegate actions to sub-agents, limiting the scope of the

reward structure for each of the sub-agents while itself taking only a small number of actions over a long period of time. Such an approach would also be useful in communicating the status of the testing and implementation to business leaders who might not be as automation-literate as the teams working with the AI. Whatever the strategies that are utilized for reining in the power of AI, effective oversight is essential.

Exploration

One of the most promising aspects of an AI is its ability to explore and better take advantage of its environment, but safety should be the primary concern when planning just how an AI might go about exploring. An AI needs to explore its environment to ensure that its processes are optimal, but doing so can be dangerous for the environment. DellEMC cannot put its data in jeopardy, as even a small corruption of the system would result in vast sums of lost revenue. For this reason, we need to employ strategies that aim to make such exploration safe. Such strategies include giving the AI opportunities to explore in a simulated environment, demonstrations showing safe boundaries, and restrained exploration in which an area is deemed safe and the AI is not allowed to operate outside of that area. Any strategy should include a set of risk-sensitive criteria that allows the AI to avoid catastrophic situations. In the case of our particular environment, I believe that it would be a good idea to construct a version of the actual systems environment that included all of the systems and a copy of a small portion of actual assets and relationships. An AI could then freely explore this simulated environment without

fear of affecting any live data. We could then set the AI loose in the actual environment and be reasonably certain that it would behave in a way that would not result in catastrophe.

Verification

To ensure that we avoid catastrophe, even at this RPA-only stage, it's extremely important to verify the correctness of the programs we use to manage our systems environment before they go live. Rather than verifying every possible input/output relationship of the system manually - which would take an incredibly long time - it would be best for my company to use some sort of model checker.

Model Checkers

Formal software verification consists of two main approaches: theorem proving and model checking. In theorem proving, verification is achieved through a proof of correctness through the derivation of a theorem. Theorem proving has the notable drawback of requiring interactivity that model checking does not [4]. An experienced user must guide the tool, which can be extremely time-consuming when attempting to verify particularly complex systems. Model checking, the more popular approach – and the one that would be best for this application - is a much easier, more reliable process for the user. Using a logical formula, model checking explores the different possible states of a system using brute force [5]. There are many model checkers that would be well suited for this application because the features and limits of the system's state space are so well defined.

I believe that NuSMV, a member of the Symbolic and Bounded model checker families that is based on the SMV (Symbolic Model Verifier) software and utilizes a symbolic representation of the specification to check a model against a property, would be a great choice because it allows for the for the representation of both synchronous and asynchronous finite state systems, with a discrete representation of time [6]. It would probably be, however, wise for the company to commission some sort of customized graphical toolset for working with NuSMV, as NuSMV can be difficult to work with.

Automated Theorem Proving

Mechanized proof assistants are becoming popular tools for formalizing and proving theorems about mathematics, computer programs, and programming-language semantics. These proof assistants often use satisfiability-modulo-theories (SMT) solvers as subroutines because SMT solvers provide a high level of automation and have become more powerful recently [7]. General proof assistants can be used to verify the correctness of computer programs, but there are also some verification tools specifically designed for this purpose. These verification tools use an SMT solver as a reasoning engine and move the user's interaction from the formula level to the program level. This sort of verifier is referred to as 'auto-active' because it is a blend of automation and user interaction at the program level [8].

CALM and Automated Theorem Proving

Automated theorem proving would be very useful in testing a system that would accelerate the rate at which the tool we use to inspect our data can be refreshed. The CALM tool, as I mentioned above, is very slow. Currently takes over twenty-four hours for changes to our database to be registered by CALM, our main data visualization tool. CALM is an extremely useful tool but often its information is too outdated to be useful in situations where a quick turnaround is needed. Its refresh rate is so slow because changes to the database are gathered for a twenty-four-hour period then added to the database all at once. It is done this way to prevent errors that might result from many concurrent attempts to edit a portion of the database. Data fidelity is prioritized over all else, for good reason, but the length of time required for the tool's view to update means that some of our quotes cannot be verified prior to deployment. For this reason it would be preferable to speed up the refresh process.

Accelerating the refresh process would require that we take into account the following variables: total size of database, number of changes made, areas affected by the changes, order of changes, and types of changes (among others).

An auto-active program verifier could be utilized to verify whichever algorithms could be designed to minimize the refresh time of the CALM tool. The CALM environment is large but its parameters are very clearly defined and the characteristics of the data it holds are limited. There are also very few outside factors that can affect the performance of the system. The very simple logic of the relationships between assets in the database lends itself to algorithms for its

cataloguing and processing. The team responsible for designing those algorithms and managing the database could certainly benefit from a tool like Dafny that would automate much of the program checking process and enable many non-ideal steps to be eliminated from the refresh process. Fidelity being such a high priority means that eliminating errors in every part of the system is paramount, and that verification is an essential step in the process of ensuring that the system remains error-free. Automated theorem provers are particularly attractive in this case because it allows the company to minimize the time and resources involved in testing refresh algorithms and ultimately minimize the time needed to refresh the system.

Conclusions

It is an exciting time to be a maintenance contracts analyst on my team. We are witnessing the beginnings of our automation program and can see its potential. Many aspects of our process can and should be automated. These particularly automatable tasks include researching asset quantities and relationships, and filling in fields in our contracts system. We have already seen some portions of our jobs be taken over by software robots and while they are acting on a relatively small scale and are often flawed, there is no doubt that refinements will soon enable the robots to do much more for us than they currently do. There are ultimately limits, however, to how far RPA can get us toward the goal of a sales representative being able to generate an accurate quote in seconds with the click of a button, and I believe that Intelligent Automation will be a necessary component of that process. Intelligent

Automation is required for the parts of the quoting process that require logic that is beyond the scope of the relatively simple software robots that are currently being developed in our group. Refining the CALM tool via Intelligent Automation will also be an important part of this transformation, as its speed (or lack thereof) will impair the ability of software robots to quickly generate quotes just as it does us.

Verification of the correctness of both our RPA and IA solutions will be crucial given how much money is at stake and how far-reaching any sort of error or downtime in a system the size of ours can be. Some combination of standard model checking and automated theorem proving will likely be the best choice for such verification.

The employees of the maintenance contracts team are looking forward to a future in which our most tedious tasks are taken from us so that we can be free to concentrate on more fulfilling endeavors, and automation promises to provide the tools needed to create that future. While we have some understandable worries regarding just what our roles will be in that future, none of us will be sad to see the mundane, data-entry-heavy aspects of our occupations disappear and I think there are many reasons to be optimistic about office work in the coming years. Robotic process automation and intelligent automation (and the verification of both) are already impacting the world in vast and exciting ways and I look forward to seeing what transpires in my group at Dell EMC as we welcome the robots to the fun of quoting renewals of maintenance contracts.

References

[1] M. Lacity, L. Willcocks, “Robotic Process Automation: The Next Transformation Lever for Shared Services ,” The Outsourcing Unit Working Research Paper Series, January 2016, pp.1-35.

[2] P. Lowes, F. Cannata, S. Chitre, J. Barkham, “Automate this: The business leader’s guide to robotic and intelligent automation,” Deloitte Service Delivery Transformation, 2017, pp.1-25.

[3] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mane, “Concrete Problems in AI Safety,” Research at Google, pp. 1–21, 25 July 2016.

[4] J. Woodcock, P. G. Larsen, J. Bicarregui, J. Fitzgerald, “Formal Methods: Practice and Experience,” ACM Computing Surveys, pp.1-40.

[5] R. Armstrong, R. Punnoose, M. Wong, J. Mayo, “Survey of Existing Tools for Formal Verification,” Sandia National Laboratories, pp.1-42. December 2014.

[6] M. Frappier, B. Fraikin, R. Chossart, R. Chane-Yack-Fa, M. Ouenzar, “Comparison of Model Checking Tools for Information Systems,” GRIL, Universite de Sherbooke, Quebec, Canada, June 16, 2010, pp.1- 12.

[7] W. Bibel, D. Korn, C. Kreitz, S. Schmitt, “Problem-Oriented Applications of Automated Theorem Proving,” Fachgebiet Intellektik, Fachbereich Informatik Technische Hochschule Darmstadt

[8] K. Rustan, M. Leino, “Automating Theorem Proving with SMT,” Microsoft Research, Manuscript KRML 234, 19 May 2013, pp.1-15.