**Notes from RPA product evaluation:**

**Table of Contents:**

1. Observations while running Bots on Citrix

2. Observations from Q&A with UiPath during product evaluation

3. UiPath vs Automation Anywhere (in addition to what’s on LinkedIn)

4. Enhancements suggested to the UiPath REFramework

**Observations while running Bots on Citrix:**

In certain shops, bulk of the business users are provided the XenApp variant of a Citrix VM, and the business applications on it are provisioned in one of the 3 ways below. XenApp is a dynamically assembled Citrix profile with no persistent storage.

1) Application packaged with the base image, operating in the same Citrix session – Easy for the Bot since object capture works.

2) Application installed in the Citrix farm, but not in base image – Depending on the Citrix version, this may run in session sharing mode (which would allow Object Capture), or open a different session, thus mandating OCR.

3) Application streamed from outside Citrix, where OCR must be used. Also in such cases the resolution difference may be significant depending on the implementation / infrastructure.

Technology staff in such shops are typically given the XenDesktop variant which is a persistent one. However streamed applications still are a problem to solve. If a streamed app, happens to be a Java based app, hosted as a web app on a URL, (Oracle Forms 11g server is an example), then, provided the app’s Java version and RPA tool’s Java version is consistent, the objects (such as applets) downloaded from the app server in the browser (assume that the browser is a base image app), we can get Object capture to work.

Citrix Windows environment with XenApp also restricts running certain Windows services by default. Some kick-off on the base image and some from a base server within Citrix. Some RPA product installations may require starting / stopping / enabling access to some of these services. In most cases these are exceptions taken but are non-disruptive. However troubleshooting can be tedious.

The installs succeed, however it is only the errors / exceptions that occur while performing specific operations in the tool that uncover these gaps.

**Observations from Q&A with UiPath during product evaluation:**

|  |  |  |
| --- | --- | --- |
|  | Description | Limitation |
| Testing of Bots | UiPath has templates for loading test data and running workflow. But is still brute force. Mocking of data not possible beyond a certain extent | Creating a full workflow just for testing is tedious |
| Integration with Git etc | UiPath provides integration with SVN. | Git itself allows multiple checkouts, versions etc to be active in multiple environment. Tying this together with UiPath Orchestrator which also accepts multiple versions of a package running in different environments is tedious and UiPath is still working on it. |
| Licensing and Bot costs (Pooled, Node locked, Concurrent) | License for a Bot runner – Locked with the Orchestrator it is registered with.? (This may have changed). However since Orchestrator allows ToggleRoles, if you have multiple processes, each needing different role, running at different times, the same bot may be used for it. | This may still need some customization on the Windows side. Since while the Bot may switch role, if it ties to the Windows credentials, then the machine set up to run Bot on may not have all the business applications required to run the process, or its credentials on those apps may be different.  This may not be an issue if ToggleRole also allows different Windows/credentials itself while logging into the same Bot (i.e. using that bot license). |
| Password renewal and first time password set up | Password renewal scripts can be written once the Update Credentials Orchestrator API is available. Another Bot or a script can Invoke it. | In certain shops, the first time password set up requires a human to call in the service desk person and verify his personal info (e.g. SSN last 4 digits). Some exception has to be provided for first time set up and then the Bot user must be set up by a human, and then the password must be immediately reset by the script calling the Orchestrator API. Script must adhere to password policy while generating password. Such a password change script needs to be scheduled. |
| How to handle cascading failures | If one Bot fails due to an issue related to a business application, which is touched by several bots. We need to contain the damage, and proactively notify other bots running to halt if they are not in the middle of a transaction. Orchestrator API allows Stop Bot / Kill Bot, and other Bots can prescribe to the Should Stop signal from Orchestrator. | Downside is that all the Bots have to keep issuing Should Stop check every once in a while. In case of a Bot iterating over multiple inputs you can check before starting every new transaction. There may be recent Orchestrator API enhancements to accommodate this. |
| Product Roadmap | UiPath roadmap for 2018 speaks about connectors being built to various NoSQL stores, Mulesoft etc |  |
| Debugging | Local variables, ToString and Iterator methods | System.Data.DataRow items don’t display well for some reason. No type specific Iterator methods. |

**UiPath vs Automation Anywhere (in addition to what’s on LinkedIn/Quora)**

|  |  |  |
| --- | --- | --- |
| Criteria | UiPath | Automation Anywhere |
| Development environment | UiPath studio is a variant of Visual Studio itself. Complete visual coding. No scripting mode at all. So simple things like looping, IF etc have to be debugged in visual mode which is tedious. State Machine as a template comes out of the box. | Coding possible in Scripting mode and visual mode both. Simple coding can be done as a script, and we can switch back to GUI. Possible complex scenarios where scripting doesn’t convert correctly to GUI, or vice-versa?? As per vendor there are cases like these, where GUI improvements are on the roadmap |
| Deployable artifact | Package consisting of .xaml files can be pushed. They can be read through frameworks. Hence with some custom effort, it could be easy to plug them in to source code mgmt tools. Impact analysis is easy, if you have to look for which business UI screens are touched by the bot flow by reading the xaml.  .xaml being a fairly standard format and can be shared across the community.  Can also deploy a .nupkg? Possibly but not tried | Proprietary format called .atmx, not readable by other systems But some bot script meta data / data is stored in tables as well? |
| Installation experience | No relaxation of permissions was needed, no additional services had to be started / stopped by our Windows admin, to accommodate the install of UiPath Enterprise on Windows VM’s within Citrix. | Some Windows services had to be stopped/ started, and/or access level had to be changed by Windows admin |
| Adding Bots on demand | UiPath does not have a feature / facility to rapidly scale to accommodate surge in volumes | Automation Anywhere has Bot farm functionality |
| Utilization of Bots (High Density Robot) | UiPath allows running multiple Bots (runtimes) on the same machine registered with Orchestrator to fully utilize its computing capacity | Not sure AA has something similar |
| Bot as a Service | Orchestrator API is provided. Bot machine directly hosting the endpoint, works in some cases, but not really supported by UiPath | MetaBot seems to be a dll file. Not sure how Bot as a Service really works. |
| Debugging | Debugging feature seems good. But memory issues with large workloads. | Limited debugging capability |
| Citrix | Citrix performance is faster. Vendor demonstrated it | Citrix capability available but not as good. |
| Technology used | ELK stack.  UiPath has attempted to make a standard procedural framework, but is tedious. | Full Microsoft stack? But as per their Roadmap, AA is looking at rewriting client and server both in J2EE TomCat? Will the IDE change as well. Will the mappings / flows be saved as .xml, .json etc? |
| Training and Community | Free advanced online training. Challenging / gruelling courses. UiPath itself and the community is receptive to new product enhancement ideas and VoC | No training. No community |
| Administrator UI | Orchestrator is web based, and latest version is mobile friendly. | AA probably did not have a Control Room GUI earlier for mobile. Possibly added now |
| Front Office Bot | UiPath has a good template for Front Office Bot | AA has front office bots too but no template. |
| Exception Processing | UiPath has try/catch, Exception propagation up the stack etc. But little tedious to debug | AA may have better exception processing, but not analyzed thoroughly during the pilot |
| Registering a bot | UiPath process is too manual. Copying bot key from Orchestrator and pasting it in Bot settings in System tray on the Bot machine | AA has a more elegant Bot registration process |
| Invoke Code | In addition to Invoke Workflow, now UiPath is coming up with Invoke Code (where instead of just a single VB expression, you can write full code). This will come in very handy | Not sure AA has this. But in their J2EE version they may be able to allow a JVM language. |
| Troubleshooting and debugging on the fly | Debug mode. You cannot pause and modify code or variables during the running of the workflow. In that case you have to laboriously go through and start from the beginning. Probably can do this in BluePrism | AA claimed to have this functionality but did not demonstrate |

In most other areas, both products seem to have similar strengths and weaknesses:

Common weaknesses:

Heavy workflows lead to memory issues.

SAP scripting in most cases needs to be enabled.

**Enhancements suggested to the UiPath REFramework**

(avoid mutating the framework and provide more elegant packaging)

**https://github.com/UiPath/ReFrameWork**

Iterator for looping over variety of data sources:

UiPath RE Framework provides a COBOL program like Looping and Transaction processing template.  
By default the Get Transaction Data logic in the loop is for an Orchestrator Queue item. Hence data types of transaction variables are set accordingly. This is tedious while debugging, and you have to modify the framework itself when your data source changes.   
Hence, implement an iterator each for various data sources like Queue, Datatable, Tabs in a spreadsheet, files of a certain name pattern or certain extension, in a folder etc. Then use a generic types for the Iterator. Allow our data source to be passed to the Framework Iterator implementation.

Can we have a right-click menu option available on the Invoke Workflow activity icon, that shows links to all workflows calling it?

A UiPath framework based project is just a collection of workflows. They are not bound as one project. Hence identiying the caller is difficult since they are not connected as a tree. Per UiPath, Since this is not a packaged IIS or Tomcat application with all components, dependencies and links bound together

Application Context and Properties file:

Just like a Spring project, an application context holding beans (e.g. a browser object identified at the start, to attach to everytime you want to work on that application), and holding properties such as configuration, credentials (or credential names at least) so you dont have to load them off spreadsheets and they are available throughout the workflow.

Allow writing .NET or JVM code, allow compilation wherever possible (nice to have)

Currently all the code is published as .xaml. It is declarative, and needs to be parsed, interpreted, and in some cases JIT compiled. For complex bots this may slow down the processing. Especially with the Invoke Code functionality, if that part of code (if allowed to write C# or Java instead of VB) and if it was compiled) may run faster. That .exe could be packaged with the project.

Full fledged use of project.json file like a project object model:

The project.json file could be leveraged, and once they figure out how to assemble all the flows in a package as a project, then this file could act as a pom.xml equivalent.  
Common components could be added as Maven like dependencies.   
Alternative – Use Maven like tool itself and it’s pom, instead of the project.json