CACAO Technical Committee

Meeting Minutes

Date: 09 June 2020

Attendees

Allan Thomson  Michael Rosa
Bret Jordan    Patrick Maroney
Emily Ratliff  RoseAnn Guttierrez
Gerald Stueve  Stephanie Hazlewood
John Morris    Toby Considine
Lior Kolnik    Vasileios Mavroeidis
Marco Caselli

Meeting Notes

Committee Specification Draft

Last call for comments on the existing draft. It is coming along nicely. We are at Working Draft 2. We will create a working draft 3 and then use that as the Committee Specification Draft. There are a few comments that need to be addressed. Review the document in earnest in the next two week. We can open a ballot after the comments are addressed.

Allan discovered a couple of things while creating an example that need to be addressed before going to a CSD.

Stephanie’s team at IBM used the specification during a recent Hackathon and has some feedback. It would be helpful to have a place to store the version of the technology that the playbook is intended to run on. Allan - the target information may contain some of this. Are you looking for what executes the playbook as a whole? Stephanie - yes, different technologies can supply different capabilities. Some products with extensions may be able to run some of the steps but not others. This could be part of the playbook metadata as technology compatibility statements.
Marco Caselli commented that the target is in the steps and in the actions which caused some duplication of information. Allan encountered the same issue when writing up the example. The distinction between action and step/target is not entirely clear. It is a bit of overkill the way it is currently specified.

OASIS is hosting a webinar in about a month and it is getting a lot of attention from the market.

Examples
Looking Glass added an example on “Alert Investigations and Analysis”. Action is to gather information. Outcome is to build a report. They defined variables that were needed throughout the notebook which brought up an immediate question about how you define the variables and how you use them.

There was also a comment about the playbook type. The example uses investigation but Allan also wanted a playbook type of analysis. The specification is lacking an explicit type called analysis. Do we want an explicit type for analysis?
Toby - notion of schedule is fairly rigid. Someone should do analysis for exactly 3 hours or no longer than 3 hours? What happens after that? 3 parameters of time: start, end, duration. There is a flexibility of playbooks that our models don’t do right now. Toby plans to write this up. Allan - the intention of this label is metadata about the playbook. How it does it is in the steps, so the question to the group is whether we want this as a playbook type. Bret - fine with as many types as we want to create as long as we are clear on what the differences are. We need to come up with very clear definitions of the types that get added. Marco - if we have very broad labels, people may default to broad labels rather than putting in the effort to correctly and specifically classifying the playbook. Allan - agreed based on that we should omit the ‘analysis’ label. Allan is leaning towards removing the label, so members should go into the draft and add a comment to register their opinion.

Patrick - is there an asynchronous trigger? Bret - yes, you can run things in parallel and asynchronously.

The next comment was around the property called source. It isn’t immediately clear what this property is for. Allan wants this to be obvious without reading the definition. Bret - The source is for information that is not a URL. Allan - Isn’t that external ID? Bret - External ID is any specific identifier that references a document. Allan recommends making this an optional property. Toby interpreted this property differently based on his experience with VA Standards of Care - these are healthcare playbooks which are licensed from Cleveland Clinics playbooks. In this example, the source links back to where these playbooks originated.
Bret - need to clean up definitions of name and source and this may lead to renaming one or the other. For now, source has been made optional. A comment added to the draft document: Add example where source is not the primary reference of the third party source.
The next comment from Looking Glass: When we declare variables, do we want to be consistent and have the double dollar sign in the definition ($$variable) or just use $$variable in the source. This doesn’t matter when playbook generation is automated, but does matter when handcrafting playbooks. Allan has a preference for using $$variable in the definition because this is easier to search for. Consensus was that either is fine. Bret added a note to update variables to use $$ for the declaration for other examples.

Looking Glass commented on local variables versus variables that are passed into the playbook. Should the playbook declare those variables that it assumes will be passed into it? There could be problems where variables external are declared differently than the type used inside the playbook. Bret - we could flag a variable during declaration that shows that it is passed in. Allan but you don’t want to declare type and value of external references. Allan - should we add scope to variables which defaults to local? Bret - we could add another boolean that says that this variable is coming in from outside of the playbook. Allan - that would be useful. Patrick - this would be useful for the concept of a dryrun.

Toby - How deep do we want the stack to be able to go? Do I want to see all levels of variables when debugging? Allan - regular debug you want to have the option to go and do that. You need to be able to inspect it and possibly assign values to it. Toby - How do we want to dereference variable names? Allan - that is why we might want to explicitly call out variables coming in from outside to prevent local variables from having the same name as global variables. Some programming languages support this, but for this specification it would be considered an error. Toby - global variables might be defined later which conflict with local variables. Bret - we do allow redefinition of variables. Allan - this can cause problems Bret added a comment to the document to add passed in context proposal to discuss next time.

Predefined prefix for local variables? We could do it that way. That would stop namespace collisions.

Looking Glass wants to be able to define structured types and not just basic types. The current definitions are just basic types. Allan wants a list or a hash map or a hashmap of hashmaps. Allan wants to be able to define a new type. Need a proposal on how to do this.

The specification doesn’t have a type for report or a blob (like PDF), we need a type for that so that we can create a variable with the report contents and then email it.

Timeout is currently a number but we may want to set it to a variable so that the timeout can be changed globally. Timeout should be changed to a string so that we can use variables to set and change the timeout globally. Toby recommends that the timestamp be ISO8601 and RFC3339 compliant. Bret said that timestamps all follow those standards but timeout is called out as a millisecond value. Making timeout a string makes it easier for interoperability.

Allan - “Action is duplicating target. Should target just be the host/auth part and the method and argos defined in the action part.” This is similar to Marco’s comment earlier in the meeting. This is a big discussion so this will be discussed during the next meeting so please come prepared to discuss.