2024.03.08 Meeting notes

Date

08 Mar 2024
Time: 16:00 (CET)

Meeting link https://tib-eu.webex.com/meet/georgy.litvinov

Attendees

- Georgy Litvinov
- William Welling
- Dragan Ivanovic
- Brian Lowe
- Mark Vanin
- Miloš Popovi
- Ivan Mrsulja

Agenda

- Updated Dynamic API branches for future 1.15, authorization policies
  - https://github.com/litvinovg/Vitro/tree/dynapi-1.14.1-snapshot
  - https://github.com/litvinovg/VIVO/tree/dynapi-1.14.1-snapshot
- Publishing updated branches
- Check style alignment
  - ~9000 style errors to be fixed
  - First the PR has to be rebased and merged
- Report generator endpoint fixes
  - https://github.com/litvinovg/Vitro/compare/dynapi-1.14.1-snapshot...litvinovg:Vitro:report-generator-endpoints?expand=1
- Binary file conversion fix
  - https://github.com/litvinovg/Vitro/compare/dynapi-1.14.1-snapshot...litvinovg:Vitro:binary-file-conversion?expand=1
- Configuration classes
  - Alignment of Configuration related classes with main VIVO branch
    - https://github.com/vivo-project/Vitro/pull/446
  - Removing servlet context dependency
    - https://github.com/litvinovg/Vitro/compare/dynapi-1.14.1-snapshot...litvinovg:Vitro:remove-servlet-context-dependency
  - Configuration classes check style alignent
  - Ontology Individual to Java implementation linking issue
    - Ontology inheritance hierarcy
    - Java inheritance hierarchy
    - Inferencing?
    - Graph storage

Meeting notes (transcribed automatically)

Georgy: okay, let's maybe start going through the agenda and discuss it.
Yeah, as always, the meeting will be recorded and I will make transcribe meeting notes out of it. Okay, so today, yesterday and today, I created new branches for Future 1.15,
I suppose, because there were a lot of modifications and there was an effort, you know, to align with that. And in agenda, if you can, can you see my screen, right?
Yeah, so there are two rings, two latest vitro and vivo dynamic API branches, so if you have time, please try it out. It should work as the branch before. Dynamic API 1.14. And Dragan, I think it makes sense for me to push these two Dynamic API 1.14.1 snapshot branches into the Vivo repository so that the pull requests could be changed against former branches and aligned with this one.

Dragan: So at the moment, the branches are only in your repository, right?

Georgy: Yeah, yes. I haven't pushed it into the Vivo today.

Dragan: But for me, there is some sense in that. The question is, you know, I was hoping that we are close to 1.15, but not sure when we will have that. But are you planning also then to align that for with 1.15?

Georgy: Yes.

Dragan: And again, to create the new branch? OK.

Georgy: Yes, of course.

Dragan: We might be lost in those branches at some moment. Are you deleting the previous one, Dynamic API 1.14, or it's still? So at the moment, there is one Dynamic API 1.14, right? And we will put the little bit of progress.
And now, if we move this to the virtual, Are you going to delete the previous branches or to keep it for some time?

Georgy: No, I'm going to keep it for, at least I plan to keep it for some time. And when we decide that it's not needed anymore, then we can delete it. But no, I don't, I wasn't going to remove it just because we have a newer version.
Yeah, and I wanted to add that those branches are needed temporarily before we made 1.15 release because we already have some something to do to align with 1.15 and for example as in agenda you see that there is a check style alignment.
So in this branches I added all the dynamic API files into check style suppressions, but I'm preparing an almost completed patch that would fix the check style errors and then we will not need these suppressions anymore.
And we will be aligned with people.

Dragan: So we argue what is confusing me here actually in the case that let's say next Thursday at the committee's meeting we decide to merge one small pull request. Are you going to update those branches?
Or you know the name of this branch is dynamic API 1.14.1 snapshot.

Georgy: Yeah.

Dragan: At which point it was snapshot? Because you know we merged maybe 15 pull requests from the moment of 1.14 released to the moment of 1.15 released. And this is some of those snapshots right in between.
So I want to say if we merge the next Thursday one small pull request and then the Thursday after that the next pull request it will be still in the 1.14.1 snapshot.
But are you going to update your branch or how we will know with which last commit it was based on?

Georgy: So I'm not going to rebase branch with this name, but if for some reason we would need next branch and we still didn't release 1.15, then I can create a snapshot too, for example.
But yeah, unfortunately, maybe we should address it at some point on committers meeting that maybe we need a more flexible naming of middle builds. For example, in some projects like LibreOffice, you always have some hash sum and you can base on that at least.

Dragan: I just wanted to say that the name of the branch is not self-described how to say and not clearly reflect at which moment of 1.14.1 snapshot it is based on,
so of course we can find that I suppose from the the list of commits or something like that, but it's not directly from the name.

Georgy: Yes, yes, of course it's not easy to understand from the name, so but I noted that that's a temporary solution. So for this temporary solution, we need to be able to right now not when not wait a month
or yeah until we have a release and yeah fix styles and address current issues that we that we have. And yeah. So let's go further. And okay, then I'll publish these branches.
If you have a better name, just let me know. We can use some better name naming. Okay, so. And the next point was this check style alignment.
As I said, there are a lot of errors and I mostly solved that already, but I haven't applied that on top of that branch. Yeah. And also there is a pull request. One second, I'll show it to you.
There is a pull request that I'm going to change its base here, not to dynamic API 1.14, but this snapshot, and apply this style alignments only after this is merged.
This is only some refactoring to get rid of JSON array, I think even tried that but it didn't work. But it didn't work for some other reason because of the configuration bin loader issue but we will get that a little bit later.
Okay. So yeah I'm just wanted to explain in what order I'm going to do so. Also, before Mark went on vacation, he was busy with repo generator, a web interface, and he asked me if there was some issues with endpoints.
And we created a small fix for endpoints, but it just replaces the PUT method for some endpoints with POST method. And second fix is also minor. It just fixes the serialization of binary files into base64.
Yeah, for some reason it was some serious mistake of me. Yeah, and that fixes are required for report generators. And we are getting to the biggest issue that we currently have is configuration classes.
So we have a configuration bin loader, it's called configuration bin loader, there are also classes like configuration or dev parser. And the main purpose of this set of classes, of that model I would say, is to load Java instances
from link data description. So we have a graph that describes how some Java instances should be loaded and will help of some annotations. These instances are loaded and they work.
So the same way is loaded application entry in Viva configuration, for example, and some other classes. But there is a problem related to how to link ontology classes and ontology hierarchy with Java classes and Java hierarchy
because we don't want to have constraints on, for example, ontology hierarchy because of the way we implemented it in Java. So we need to decouple that and the other way around.
And currently, that's one of the, that's the biggest problem why this, how to call it, array view. ArrayView with help of container arrays based on JSON arrays. They didn't work because they use a special type called array type parameter
parameter type. And because at the time when configuration bin loader tries to load this from the graph description, it confuses because it has two types, one of simple parameter type and another one is array parameter type.
And we need to be able to distinguish between more specifically set types and generic types, some other types that are in hierarchy, but on some other levels in that hierarchy.

Dragan: I'm just thinking, you know, I suppose in some cases for the complete ontology hierarchy, we need just one Java class, right? Or we always need for ontology hierarchy the Java hierarchy.

Georgy: For each individual we are going to load we need some java class and then we need this individual bind to that java class. Initially in vivo it's used without ontology.
So you just have individual and you just state that it's individual of type and you write this java URI and by this java URI this individual will be loaded.
But in case we have a lot of individuals and the way we work right now with procedures and endpoints, we deal with ontology because it's much easier to work with ontology and check what we made with ontology in future.
There is a problem. I faced that problem something like a year ago, I think the first time. And now this linking between the profiles, maybe I can show it to you. Linking between the profiles is made by you.

Dragan: I'm just thinking at the moment, maybe I can think loudly. I'm not sure how much this issue or this problem is overlapping with Oramo tools or JPA. You know that there is the way to say one class per hierarchy
or when class per instance and so on, because, you know, in the relation model, you could have let's say that the more than one tables for and then in Java, you could have the hierarchy and then you can say it will be
all in one table or it will be all in for any single class from that hierarchy, there should be a table in the database and so on. Can we take something from that idea and apply here? Because we here have also the Java
hierarchy and the Anthology hierarchy.

Georgy: What project did you mention right now?

Dragan: I'm talking about the object relationship mapping tools or JPA. Java Presence API, and there you can say if you have the hierarchy of the API entities, you can specify whether any single class will create the
separated table in the ratio model, or you would like to have...

Georgy: In general, I see. Yeah. I don't think that any of these ORM tools fit for us because they mostly work with relational databases and our problem is just to traverse the graph.
So I think the biggest problem here is only this figuring out in our implementation of the graphs which is the most specific type of that individual in the graph. We do that, we solve this issue in vivo assertions graph where we store all the data
one way with specific defect, I think, defect reasoner. But it's slow, first it's slow and second it's a little bit outdated. So I think there should be a better way to do so and I wanted to show you on the screen.
So that's the way how right now the dynamic API procedures and dynamic API is the ontology connected with Java classes. So you see the components is start with Java and that's how the class is found and loaded.
But in case, for example, here you see here two types. And in that case, so that's just bad example because there shouldn't be the first one. But because array parameter type is a subclass of parameter type, you still will have as a result
two types as types of this individual. And that's due to inferencing. So if you turn off inferencing, then you no longer have both. And to get the most specific, you should make a sparkle query which is not, I think, a big deal for that case.
But yeah, there could be different approaches. For example, instead of just using this, so here I use subclass of which in the end will result in a dev type of the same thing. But maybe we could use some object property which is not used by the
reasoner in a transitive way, so it will not be duplicated to the top of the inheritance hierarchy. So the first approach is to use some other object property and in configuration bin loader make a query and first find options that use this special
object property that we created and as a fallback use a dev types. Another approach could be still use the same thing, but just modify, once again, do the SPARQL query. And in that SPARQL query, without inferencing, try to distinguish the hierarchy.
Because there could be cases when you have these Java types specified on the second level, some other Java types specified on the third level, on the four level, and you need to have this precedence one or other.
But in that case, we would need to turn off inferencing, and I'm not sure that currently in configuration triple store. It's a configurable option, I would say, because everything I've seen there are just ontology models.
And by default, if I'm not wrong, ontology models have inferencing. So yeah, we have a few options, but we'll see what it will result to, but something to think about. Yeah, so if you would like to say something,
some ideas more, yeah, we can discuss it now or maybe later. Okay, and also I created this pull request as you might have noticed against the vivo because we already have some modification for done in dynamic API branch for configuration bin loader,
configuration rdf parser, property and property types and wrap instance. And as I said in the description, Some of the modifications, like for example, this modification are solving some old issues. And so there was a circularity prevention that implemented,
but I found a way how we can still allow circularity and yeah and this is covered by the test and also we have a few more uh a few more methods so this is an annotation if you have seen this property annotation.
And now it's also able to provide a URI as a string not not create new object but use this URI only as a string value to save it. And yeah, we have more data properties like boolean data properties, integer data properties.
I think there was something like float. Yeah, float was extended with XSD decimal.

Dragan: And this is the pull request created over the main branch.

Georgy: Yes, yes, yes. That was cherry-picked from Dynamic API. In Dynamic API, I had to split a few commits, maybe five or six, to extract this. So, for example, this one, the first one, that's from this commit and the rest
of commits is the same but without these two files. So I just extracted that and it's also in dynamic API branch right now but if we merge that into the main branch and then after that we will rebase our current dynamic API
on top of that modifications so the commits will be just removed from this from dynamic API branch and they will just stay in the main.

Dragan: And in that case we should be careful right with the strategy for merging. It should be rebase and merge.

Georgy: Yes yeah it should be rebase. Yes exactly yeah.

Dragan: Okay just not to make the mistake there and if we you know, squash the merge into one, we could have an issue, right?

Georgy: Yeah, that will be an issue on rebase. Yeah, I hope we will not do that. But yeah, it can be fixed anyway. But yeah, you're right. So, yeah, so we have this one also, as I... Yeah.
Also, we don't need any more servlet context in some classes related to configuration bin loader and some other classes that are connected to the configuration bin loader.
So I removed servlet context and it's a safe operation because for example, this model access on context. Nowadays, it just forwards the call to get instance.
And that would allow us to decouple a lot of code and make it simpler. For example, maybe I can show it here too. So that's a configuration being loaded and we just don't need a lot of different constructors and save checks.
We just can safely use more or less one or two constructors, one with standard model, another one that create the standard model that wraps model with log safety and another provided log safety.

Dragan: Again, I have a question here. This is just a difference between the main and your. Are you going to create a pull request?

Georgy: This is the difference between dynamic API snapshot and this remove dependency code. I will create a pull request when we finish with this one. I will create a pull request to fix that.
And after that, yeah, and after that, I also going to clean up the classes to align them with check style rules, because now, as you might have noticed, they don't look well.
So for example, these cherry-picked classes, yeah, sometimes you see indentation is, with done with spaces, but sometimes indentation is made with tabs, I suppose. Yeah, with tabs.
And yeah, that should be, I think, in a separate pull request because that's just cherry-picked what was already done. And also I plan to cover configuration bean loaded in main branch after that with more tests
because I think the amount of tests is not enough for us at the moment. And then when it's merged into the main we will be able to rebase on top of this main and use improved configuration bean loader in dynamic API
and we will have a reliable tool, more reliable than it is at the moment. Yeah. So I already said about the check style alignments. I also mentioned this linking issue.
Yeah, and also we might address at the same point the way how the graph are stored because now the two ONT model are taken and they added as a sub model for new created model.
And this main model is used to instantiate Java instances, but I'm not sure that will be the final way how to do that, especially from the perspective that we were going to work on this graph safety, so graph isolation.
So maybe we will be able to address it in the same issue, maybe not. Yeah.

Dragan: And for this, the issue with the hierarchy, so we have the hierarchy of ontologies, we have the hierarchy of Java classes, and we have the issue that it's not seen by each other, right, somehow.
Maybe to improve this ontology, specifying linking between those two, right?

Georgy: Yes, we need to find a way how to specify linking. I can show you maybe, one second, I need to move. So, maybe one second. Do we have a test here? So, I created that kind of sparkle query and tried to use it,
with current configuration rdf parser. And just to explain how this works. So it's kind of complicated query. So it tries to find all the classes that are ontology, ontology classes here and their Java implementations that bound to that classes.
This is basically a copy to be able to filter out cases when a class is a subclass of itself and find out if some of that classes have hierarchy between each other. And here you can see, for example, that class array parameter type is a subclass
of parameter type. But the problem is that I didn't find a way, simple way to distinguish between two results at the top, because both, so array parameter type have type both this Java and this one too, because of inferencing.
So inferencing just adds types to all of the, classes and all of the individuals to the top.

Dragan: Yeah, I see.

Georgy: And this is the same maybe that you have seen in Vivo and sometimes that's useful. But for cases to find out who was the real ancestor, who was the last one, the previous, so you have to do too much checks here and for example here.
Maybe just, you know, remove a graph related to, or farther to the parameter type from array parameter type.

Dragan: Maybe I don't understand that properly, but on the left side, you've got five, let's say, results of Java classes, which might be a candidate to be in the representation of some ontology, right?
Some individual from, yeah. And now you would like to know which one of those five lines, although it's just two, dilemma between two.

Georgy: Yes, the dilemma is only between two in this practical example.

Dragan: Which one should be used, right? But can we resolve that at the level of the Java, meaning that take all results and then select the most specific one where you're trying to check whether one is a subset,
is a subtype of all others, I mean the instance of others. So, but you would like to resolve that at the level of the information in the graph, right?

Georgy: Yes, because the problem is come from the graph. I suppose it should be resolved there because if we try to resolve the problems that we created by inferencing, but the problem are not in Java, then we add some limitations
on Java implementations and the way it is right now. And that's what causes this problem that... the Java implementation is already limited to an ontology.
And yeah, we need, I think, to resolve the source of this problem.
And you see five here only because the last two are how to code full bags. So when the type is just specified so URI or rdf type and this name and it's specified here and specified once again because of interesting...

Dragan: Yes...the issue for you is only first two lines. Yeah but basically you need to decide first two lines which one is...

Georgy: Yeah, and there is basically no way to do this.

Dragan: Except on the Java side to check whether array parameter type is instance of parameter type and whether parameter type is instance of array parameter type.
But as you said, it's not a good solution because it's on the Java side, not on the...

Georgy: Yeah... and I think it would be much easier to solve that in case, for example, in case I would have this assigned by another object property, not a rdf type or subclass of, but by creating my own object property that is
not propagated by the reasoner to the...

Dragan: Do you think that we should also represent in the graph inheritance between Java classes. I mean, can we say in the graph that this array parameter type on the left side is subclass of parameter type?
So do you see the left column, the first two rows?

Georgy: We need that for ontology purposes. So at the point when we would use ontology to build an editor, we would need to describe to that editor relationships between ontology entities.
And if we don't have inheritance on ontology site, for example, if you remove this relationship that array parameter type is a subclass of parameter type, then we wouldn't have this information.
So yeah, the problem is that one hierarchy now is bound to another hierarchy.

Dragan: And I think... I mean, can we say in the graph that you see this starting in Java, edu, cornel, manlib, and so on. That term array parameter type is subtype of the parameter type, which is just below that.
Not this on the right side.

Georgy: You mean implementations?

Dragan: Yeah. To represent the connection between or hierarchy between implementations in the graph. And then to use that from the graph to decide which one.

Georgy: Unfortunately, that will not change anything because basically inferencing already does that and does more than that.
So it copies it everywhere. So it's just a little bit of mess and you can't distinguish which one is a subclass of which one because they all have the same types in the end.

Dragan: Yeah. I'm confused now, but okay, if you say that, so it looks to me as that connection might describe that array parameter type is subtype of parameter type, but if you say it's not resolving this issue.

Georgy: It's already stated like that in ontology, as I believe, and yeah, in this...

Dragan: But in the anthology, you're saying that this array parameter type on the left side is subclass of parameter type, which is in the fourth column here, right?

Georgy: This array parameter type is subclass of this one.

Dragan: Yeah. And in the anthology, it's not said that the left array parameter type, so in the first column, is subclass of the parameter type just below that.

Georgy: It said that array parameter type is subclass of this class.

Dragan: Yes, and the right one.

Georgy: And because of inheritance, it's also a subclass of this one.

Dragan: Exactly, but you don't... Okay. It looks to me as you don't have the information about the Java hierarchy.

Georgy: About Java between this one and this one?

Dragan: That's the issue for you, right? Because you got as the input this class from the second column. Let's say the first line, right? Of the second column. And then you have to decide which one implementation
should be used for deserialization of that or representation of that into Java, right?

Georgy: Yeah, so you mean that maybe introducing these relationships between Java implementations might help too? I will try to investigate that. Interesting. I wasn't sure that it will work with cases.
Because once again inferencing just copies some information back and forth, but I don't know. I should try. I guess. Things that's very. That's very useful. So, okay we have that options. And yeah, I don't know if you noticed that I added the meeting notes for previous meeting and a few other previous meetings, I suppose.

Dragan: Are you consolidating, by the way, the automatically transcribed text or just copy it here?

Georgy: Sorry, what I do?

Dragan: Are you cleaning up the transcribed automatically of the meeting or just copy it here?

Georgy: I'm trying to clean up something. I can say that I've listened for everything, just trying to fix some obvious problems. For example, this transcribing mechanism is failing at identifying who was talking
and sometimes it fails to split in the correct place when one person stopped talking and another one started talking.

Dragan: Yeah, but I just, just one was wondering, you know, whether it's better solution than the summary used in Zoom or it's, it's hard to estimate, I suppose.

Georgy: Yeah, it's hard to estimate, but at least it's, you know, have more raw data and if one, would like to understand what was there really talk about. I think that's a case solution.
Yeah, but there was a few instances where for some reason there was my voice and this system didn't transcribe me, but I was on the record. Yeah, but transcribe everyone else.
So that was, that was strange, but we'll see how it works. I heard that my colleagues in TIB found some other solution that also can be used similar to what I use. I use this Whisper with diarization.
And yeah, that kind of work, but yeah, it could be better. But it all creates a lot of text and more or less correct text, I would say. But it takes time to transcribe that.
So I think something like an hour it takes on my PC to transcribe that and and it's it's not an old CPU so it's something like Core i7 14 700. So yeah it's kind of new. But yeah maybe that's not a bad solution too.
so just try it. Have you have you tried to read some meeting notes?

Dragan: Yes, from previous weeks, yeah, which I missed. I tried to read it and there are useful information here.

Georgy: Okay, so it's more or less meaningful transcriptions.

Dragan: Yeah yeah I think there is some sense in that.

Georgy: Yeah yeah okay and yeah I only recently added uh for the meeting notes for the december and january. So look if you haven't seen that i think I haven't added any messages about that in the slack.
Yeah so basically that's all I had for today. Do you have?

Dragan: I have one, maybe just brief question out of the topic of this meeting. So we then just wrote yesterday that he has an issue with the links from the emails for the resetting the password.
I think that even use the same basic links as it was used before when admin is, you know, let's say starting initiation, restarting the password. And so did you try giving you maybe those links from emails?
Because it works for me. I'm not sure what to do.

Georgy: I tried that before recent modifications and it worked for me.

Dragan: So it looks to me as there is something which is protecting from the email client or something because that it loses the email somehow protected maybe from from the emails mailing server or from the client?

Georgy: Maybe not sure what can what can be there. But yeah it worked for me before and each time I use this feature in general and view it works. But yeah, I wanted to review the PR today, but I didn't have enough time and test it today.
Yeah, but I'll try to do that maybe next week. Because today...

Dragan: In your testing, you will probably try again.

Georgy: Yeah, definitely.

Dragan: Please just when you do that, just inform maybe William whether it's working in your case or not.

Georgy: Yes, of course.

Dragan: Yeah, that's all. Yeah.

Georgy: Okay. Yeah. And with these resting points here, we will have to wait until this configuration bean loader issue is fixed and we are able to do so. Yeah. Okay. So thank you very much for your time.

Dragan: Yeah. See you. And see you next week.

Georgy: See you in two weeks.

Ivan: See you in two weeks. Bye-bye.