Can you think that you can see my screen?

OK. So first of all again thanks all for joining this uh, this

meeting today. And as I said the the goal of this, of this

meeting is to.

Present what we have done for the supporting development of

recommender systems and the idea is that we are going to present

some research questions just to to, I mean to lead the

discussion that you would like to, to have. So we'd like to

have some feedback from you from your side and so and this is

what we're going to do is.

I'm going to have a first you know a presentation about the

challenges and now we want to I mean we we try to to to mitigate

and address the challenges around the development within

their systems and then I passed the the, the the floor to to

cloud that is going to to to give some more details and also

to to show demo about the tool that that you're going to again

that we developed for supporting development of recommender

systems. OK.

So yeah and why we are all here because as I said we, we, we

want to apply this methodology that is applied in many, I mean

many that they I mean many domains and recently this methodology is also I mean I mean is gaining you know

becoming popular also in in our domain in software engineering

in in general.

Had been involved in a couple of focus groups, uh, in the past

and I think that it's a, I mean it's a lightweight manner for

doing evaluation. Of course, as I said that you cannot do any

quantitative assessment, but it's a.

Good way to to do some qualitative assessment about

technology or the methodology that that you are proposing.

So the participants, um, I would like to to have a kind of a

round table because we don't know, I mean especially [omitted]

does not know all of us. So yeah, I would like to, I mean to

ask all of us to do a short presentation so that [omitted] will

understand who we are and why we we are here. So yeah, [omitted], do

you want to to say something?

Yes. Uh, so I'm [omitted], work as professor at the Western

Norway University of Flight Sciences. Very long name.

And they've been working with this model driven engineering

for uh.

17 years Now, I think.

Uh.

Yeah, so my involvement in this is maybe from the perspective of

model repair and constraint programming and these things,

so.

I think that's enough from me. Thank you. Thank you again.

[omitted] is not here. So [omitted]. Yeah. You want to do a short

presentation about about you? Yeah.

Um, so I'm working at Siemens industry.

It's digital industry software. So that's the business unit

within Siemens who sell any kind of industry related software.

It's software for product design and then also for simulations.

So solving all sorts of physical questions like simulating the

flow around a car for example is a typical or what you see behind

me is simulations that you would do to optimize.

Any kind of turbine.

So that's that's the the background that I'm working in

from studies. I'm, I'm a physicist.

And now within Siemens, I work in pre development in a strategy

and Innovation department. In working on pre development

topics related to artificial intelligence and machine

learning. They are also recommended systems and other

things like quantum computing.

Yeah. So and we have a couple of.

Activities related to quantum related to recommender systems.

And there are many challenges.

And from for many things we need to build ontologies.

Which we but there are other scenarios where we can can be

purely data-driven. So yeah that's I'm very interested to

see what what you guys come up with. OK. Thank you. Thank you.

[omitted].

Yeah, right. Yeah. Well, I I'm professor here at the University

of [omitted].

And yeah, my group is working on model driven engineering, all

sorts of uh all sort of solutions to automate different

things using typically models. And recently we have started to

develop a model driven solution to automatically generate

recommender systems for modeling languages for typically for DSS

and again that's.

My relation with recommender systems.

Thank you, [omitted].

Yes, hello. So my name is [omitted] Vimma. Uh. I'm a professor at

the Johannes Kepler University in Linz.

Um. And we we do software engineering, uh on working on several topics like modeling software architectures. We do a

lot of automation based on AI. And of course here also there is

the intersections with recommender systems.

And um.

Yeah.

We are also doing some quantum computing nowadays, so it was

interesting to hear that Simmons is also working on this. Yeah,

and I'm looking forward to our expert group meeting today.

Thank you. Uh. So I'm also going to do also short presentation

and then I will pass through the words to [omitted] [omitted] and the

[omitted]. So I'm I'm from the University of Lapira and my

research interested related to model driven engineering and.

We have been working on automating the different model

management operations.

Making evolution convolutions and in the context of a European

project, we mean three or four years ago we started to work on

recommender systems in domain of software development. So what we

started to do was developing recommender systems to support,

you know, developers, different activities of developers and

then we at certain points, I mean we've got many, many

results in this, in this respect.

And then we, uh, we are trying to bring all the knowledge that

we gained now in developing recommender systems for for

software engineering to the modeling world. And this is what

we're going to present today is one of the results that that we

got for during these activity.

[omitted].

Yeah, from from the same University of [omitted]. And I'm a

third year PhD student and I started. I mean, maybe PhD.

And by investigating this local platform for recommendation

system of course and also.

During these three years I working uh in applying uh new

new technologies like for instance machine learning

models, data mining approaches to support their development,

the development of recommender system and recently we we

started to investigate the local proach.

To support the design of these complex systems, basically.

Thank you, Frank.

Good morning everyone, my name is full and I am originally

originally from Vietnam and now within the University of Lapila

together with the diabetic with joy and cardio and before coming

to work with the Polytechnical university body and when I worked with Thomas and they the group was is working mainly

about recommender system linked open data.

So I've got uh experience with those topics and then when I

came to Laguna we continue working with the same topic

about recommender system and also to deploy you know many

applications using deep learning, machine learning and

for mining software repository. So like for it GitHub or Stack

Overflow. So this is what we are doing right now and we.

Would like to you know, to get to know you and of course to get

some feedback from you and and we are very you know happy to to

see see here today. Thank you very much.

Thanks punk. Uh [omitted]. Hello everyone, I'm [omitted]. I'm a

researcher at the University of Languilla. I'm working with

David de [omitted] and [omitted] so. The topics are very similar. In

particular, my topics is related to mining software repository,

mining model software repository, and then applying

this knowledge to recommend items in, for instance, for

modeling assistance.

Thank you. Thank you very much. So, yeah, let's.

Move on. And as I said we are going to have a two-part. So in

the in the first part I'm going to to say something, I mean to make an introduction about the problem of automating the design of recommender systems. What are the existing solutions, what are the open challenges and also I mean we have a set of research, I mean research questions and and then we we have a first discussion about.

Most of these part so and we are all involved and and we would like to to see what's your opinion about this first part.

And the second part is about a demo. So therefore rack is the tool that we've been developing, developing for a developing recommender systems, recommender systems that are.

Not only related to modeling but he's supposed to to be to be working on any application domain. That's the ambition and let's see. So we'd like to get collect feedback and opinions from for from your side and of course we have a set of research questions that you're going to use for you know driving the discussion. So yeah automatically the design of recommender systems. What's the problem here? So the context of course is I mean we want to focus on this.

Uh, context. So the the context is recommender systems in software engineering, right? So we want to support software engineering activities in and we want to reduce the burden on the. Typically software developers are to travel to,

access and browsing the various sources they're using. I mean

selecting and using items that might be relevant to their need

and and to their current development.

Tasks and this is when when recommender systems came to

play. So essentially tools that simplify these activities here.

However, the development of what we call the customer recommender

systems is a challenging tasks. As I said during the Cross minor

project, European project we developed at different

recommender systems and we developed the recommender system

that work I mean we're asked by the use case partners. So in the

project we had we had a number of research use case, I mean

industrial partners that I mean identify with the.

Some requirements, um some activities that they wanted to

be supported by tools like recommender systems and I mean

we this means that we developed a number of recommender systems

and this was a challenging task because we have to understand, I

mean probably understand what was the problem and understand

how to automate that part, how to create, you know, a data set.

Starting from raw data mining, this raw data filter out

elements that were not elevated. And I mean, you all know that this is, you know, very challenging, challenging tasks.

And the currently the bus, the best that you have is that you

have a kind of block black box systems. So think for instance

to.

The the tool from from GitHub, not the that they introduced

this environment to to support the development activities. But

this is a black box, right? So you don't have any ways to

customize any part of the of the overall process. So you have to

to use and you're supposed to to use the recommended system as it

is no without any possibilities.

Of touching it. So what we then what we?

The intuition here is that we want to develop a local

environment, so mean a modeling environment to simplify the

development of recommender systems, right?

So we want to propose the, the, the, the, the user what you

suppose then and the user of the recommender systems with a

number of.

Constructs that can be used to you know simplify all the

problems that are typically, I mean all the tasks that are

typically performed when you have to develop recommender

systems. So and of course here we have also to to take into

into account to you know, to aspects. So automation versus

interaction because here you can for instance.

On the on the right hand, essentially you have a

recommender systems that make use of user feedback while the,

I mean the recommender system is used. So for instance the

recommender system you know recommended the adoption of

summer library and on the left hand side essentially you OK,

you get these silicon rendition, but you don't have any mean to

say OK this recommendation for me to this context is not

relevant you know and here.

We have the need of also employing the same user feedback

mechanism. There is also another dimension is if the system is a

proactive or reactive. So without explicitly asking for

recommendation, this system should mean can not recommend

items or recommender system should be.

Explicitly asked to provide the recommendation that respect to

the current context and also the way the context is actually

drawn and defined.

It can be the case that the user is the one that is going to

establish and define the context or the context is dynamically

automatically created.

So what are the existing approaches? I mean I'm sorry

that the veto is not here because I mean you, I mean he

was supposed to be part of this, of this group because Vito is

working with the Tomasso Dinoia and and then that group

conceived Elliot, which is I mean a framework for supporting

the evaluation of recommender systems. So they have a

language.

They're essentially can be used to specify, uh, how to evaluate

the recommender systems and the language is, I mean based on the

YAML format, and you specify what was supposed to be kind of

requirements of evaluation and the the the frame of

automatically.

Perform the evaluation itself. OK you specify. For instance

will target evaluation metrics or target the parameters? What

are the splitting rules of the of the data set and the system?

I mean as I mean number of automated activities that

simplify the evaluation of the recommender system.

And then uh, we have also Droid and and one, correct me if I say

something wrong because the Droid has been developing the

proposed by the Group of Guadalajara and Esther Guerra.

And essentially Droid is a local approach again to simplify development of recommender system by focusing the modeling

domain. So for modeling assistance, OK and.

And your proach is based on a customer they sell that specify

I mean the relevant the concepts that are needed to configure and

build.

The different components that are part of that compose the

recommender system.

And then of course we have out on the, you know, line of

research paradigm and essentially this field aims to

automatically identify what's the approach that perform better

for certain for certain task. OK, so essentially even the

understand what are the value for the different parameters of

people model.

Then this is automatically done or automatically supported by

outernet.

So what got you? Yes. You know Western from [omitted]. Yeah.

Right. Quick question. Yeah.

Are you going to share the slides?

Afterwards, because I would like to go through some of the

literature that you sure, sure, sure. We are happy to share

these slides. Perfect. Thank you. Thank you. Welcome.

So yeah, the I, I don't see any of you while I mean uh sharing my screen. So [omitted] interrupt me if there is some raised. So what are the open challenges, at least the challenges that we consider relevant to be you know, high priority challenges.

So first of all is managing interagency. What the, what does it mean? It means that you we all know.

That when you have to.

Uh, to design, develop, recommender system and then for instance you and mainly when you have to.

Mind, I mean mind the data that then will be used to to feed or to to to support the recorded the recommendations you see that you have two possibly mine data from atherogenesis sources right? And and most of the cases managing intelligibility represent amount. I mean is among the the, the the big beast and now that you have to.

And to manage in order to bring the data in your progenies format or in any way that you can.

Define in Asia in in more convenient manner OK by the subset in subsequent components of the commander systems.

Uh, OK.

What does it mean? It means that the existing approach is working in very specific domain so.

And what we would like to do is again try to conceive an

approach that can be used on any application domain. So we could

use the approach for developing publico under system,

potentially for supporting software engineering tasks, but

also for modeling tasks. And again we need to understand how

far we can go, not with this ambition.

So now, uh, yeah, the, this is the, the first set of research

questions and and the idea here is to again do kind of

discussion and presentation about ourselves with respect to

the experience that we had on the recommender systems and the

experience that we have with the tools that support if any, that

support the design of recommender systems and also we

would like to.

Have a discussion about the experience that we have in

modeling the complex software systems now of course now since

the group came up to be quite restricted.

Let's say only only [omitted] is. You know he's a.

Is new in this in this group, but I mean let's.

I mean stick to the plan and let's have a a quick discussion

about this research questions. So yeah, let's do again another.

Around the table to to answer these questions here.

Um, yeah. Let's start from [omitted].

So [omitted], the first question is again what's what you, what is

your experience with the recommender systems in general

and what are I mean the experience that you had with the

tool to design this recommender systems and also if you have any

experience with the modeling complex software systems from a

modeling I mean and then the perspective.

OK, can you show the the question once again? Or maybe

you can if you make them small and exactly like this. So what

is my experience with recommender systems in general?

So first of all when we use recommender systems?

The persona that we are addressing is the user of our

software.

These are typically not software engineers, but um.

Designers.

Or um, simulation engineers depending on on the software

that we are using or that the customer is using and then.

What we are working on is for example well assist the customer

in using our software, for example predicting the next

command which.

Which should follow what the user has been done previously.

If he if there is a sequence of clicks, then we want to optimize

the user interface according to what we expect the next

meaningful click would be.

That's that's the kind of recommender systems we are

looking at.

Other recommender systems um activities go into.

Um, I don't know whether it's the same understanding of

recommender systems, but if you have a.

Database of simulation models or um, cut models. For example CID

models. In large custom large companies you would have a

managed system, a data management system for those, but

then finding.

The relevant um model out of this database is difficult.

Therefore a recommendation. We are working on recommender

systems or we currently building or recommender systems for this.

OK. So and but in from your side essentially you are developers

of recommender systems, right? And you're developing

recommender systems that I mean user your clients are going to

use.

Exactly. So we are developing recommender systems, usually on the level of a proof of concept, and then we hand this over to

the real developers, the real developers, and we need to

really make sure.

That they yeah or.

Has a minimal burden of implement this properly into the

product. And yeah, I mean now you have a question that is

related to the second point. So do you have, I mean have you

found any commonalities or typical recurrent pattern while

developing recommender systems for different clients?

So what I'm saying is that have you noticed that the possibility

to exploit a common process for developing recommender systems

and also kind of reuse components that you defined?

I mean during your development activities.

Experience.

Um.

I have to say that we are at an early stage at this stage where

the underlying model is defined by us, trained by data that we

have.

And then we shipped this out to to the customers. There is now

infrastructure established and communicated to pilot customers

which would also allow some customization, extracting off the data, sending this to a.

To a cloud service where a machine learning model is

trained and this is feedback. Feedback OK and and well.

Underlying uh.

Like the the, the definition of, for example, of ontologies is

typically not dependent on the customer.

Yeah, it's, it's rather dependent on the product now.

So, and I mean this is the kind of reusability aspect that we

can say no in sense that you develop an ontology. And if this

does not mean that this ontology is going to be used only for one

specific customer, but maybe for a set of customers that work on

the same domain exactly, yeah. So exactly this can be seen as.

In some cases it's just a necessity to have an antology

because data is so rare. Industrial data is so precious

that you you, you cannot have this big data approach, but you

have to use any context which is available, any relation between

the data, in order to enhance whatever you want to do. So many

of those approaches have to rely on ontology and there you can

say, yeah, that's a.

From a reusable point of view, makes sense, yeah?

And since you said the data are so precious, of course not. For the for, for not.

Obvious reasons um, uh, how do you train your models?

So you I mean it's a big part of the recommender system

development is actually essentially train your model,

you do some experiments, you see that the accuracy that you get

is good or not. And then you it's an interview process now

and as a Siemens now as a developer recommender system for

your partner, do you have the availability of the data?

In order to do this training and this iterative process.

In some cases, yes.

And in case you don't have what, what, what what to do?

We have to prepare the data internally.

OK, so we mimic usage of the software.

And by this we create a data set and this allows us to define

ontologies and pipelines. But as I said, we are at an early stage

of of recommender system or utilizing recommender system

technology. Of course we are dreaming of a general assistant

when using our software like a theory which you can talk to and

then it's much easier and.

And then much easier to use the software. But it's it's very

complex software. It's also very complex user behavior.

And therefore it's it's it's tricky.

OK. And uh, do you have any?

Uh.

Uh, I mean, do you have any, uh, modeling, uh, support?

To develop this system. So what I'm saying is that you start

every time you start from scratch. And you.

Even I mean this is a question to to say to ask if you have or

if you would like to have a modeling approach for supporting

this activity for the supporting development of recommender

systems. I mean before I mean opening the development

environment and start working in developing all the recommender

systems. I am asking if you have any experience and all or need

of.

A modeling environment for supporting the system

development.

Uh, well, it could help, yes. So.

At this point we are. We would do this with Python scripts for

example. So also we would benefit from a low code, no code

platform in which you can quickly put together processes

and models. And for example if you want to compare the

performance of a graph neural network and then if you have something low code which just allows you to.

Quickly put this together, try out then this would help at the

pre development stage of those things, yeah.

OK, if it's if it's productized, then it's something else, then

it has to be probably written in C++ or or Java. But for for

developing the proof of concept, they are a local platform. Would

help.

Good. OK. Thank you. Thank you very much. Uh, yeah, let's move

on. We did the same. Follow [omitted].

I mean I'm going through Amen. I'm following uh the you know

the way you have my screen. I want to say something. Ohh,

sorry wrong wrong. Please phone.

Do I have just quick question?

We cannot hear you.

Can you hear me now?

Uh, yes.

No.

It's a no, it's not constant. You have some problem.

No.

Norfolk.

I mean, we move on the front. Maybe you can fix the problem.

OK. OK, [omitted]. Yes, um.

Yeah. So first of all, there's just technical thing about

calling these four research questions. I somehow would say

they are not research questions. These are a kind of uh, yeah,

but yeah. So my experience with recommender systems comes from

mostly from modeling assistants and model repair software.

Uh, which I think all of you could relate to. You know what

it is, but just shortly introducing it is more about

statistically say seeing what are the next steps uh developer

would take in the next step when they are doing some modeling,

right? And then if some errors happen so.

What's the nearest solution for that? For that error or for that

problem which appeared in the in the in the modeling process?

And also we've been working a little bit with uh multi models.

So you have multiple models which are corresponding to with

each other. If you have multiple companies or in one company

where you have different systems, they don't usually talk

with each other. And the idea in that project is to to recommend

the users OK, how which parts of the data should be made related

with each other based on analyzing the.

Contents of the data. Um, but my experience with.

Tools that support the design of recommender systems is that we handcraft things from scratch. It's not good if if if I get the question right, it's like.

I don't know if we managed to find any, um, reusable pieces of software or something which is some framework which you say, OK, take this and I feed it with my data and it tells me what's recommended for the next step on these things, OK.

Um, yes, modeling complex systems is also related again to this model management activity with correspondence, uh, definition between different uh.

Systems and and.

And yes so it's it's really come back to you know as you know of course in in research projects you want to simplify the, the complexity and you want to sell it in the papers as easy as possible, right. So we we leave the complexity to the developers who are on board on the project Research project, right.

So and the one question, I mean for instance in the case of [omitted], he said that they start with the kind of proof of concept implementation that consists of a Python scripts and then once they mean stabilize the things they move towards you know C# or C++ or Java.

I mean they created the product uh do you also have the same approach? I mean no, I mean I mean in in case of [omitted] there are, I mean in a company then you have like products and you have to deliver on these things. So so research, research projects usually we do the proof of concepts and which language is used is, I mean in many cases Python is a nearest one you you pick uh and then it's depending on the students or the the researchers involved. So sometimes they.

They migrate quite quickly to to Java uh and sometimes they they keep it simple in in Python And just have a proof of concept and the moment it goes out to be produced a software then there are software professionals who are doing the the development and then they have their own.

OK in environment to do these things. So I think, I think Java is a is A1 which is usually they pick to to do these things, yeah.

OK, good. Thank you very much.

Any questions for [omitted]?

No.

UH-1.

Uh, right um.

Yeah. Uh, well, uh, uh.

Yeah. First question is what's your experience with recommender

systems? Yeah, well, um.

I would like to mention 33 relevant systems maybe not sure

if they can all be called, uh, recommender systems. So the

first one is called extreme.

Um, which is uh, which is a modeling assistance. So it's

basically a tool. I will put your reference here.

Which gathers heterogeneous um data, um ontologies, models, XML

documents, you name it.

And then you can, while you are modeling, you can, uh, query uh

in a uniform way those data sources so that you get

recommendations. So it's not.

It's not. Uh, yeah, you you need to to build a query, so to say.

Uh, the second system is a kind of a system, but it's a little

bit different because it's based on assistance using a natural

language. So here the modeler.

Talks uh and describes. So it's an assistant for conceptual,

conceptual modeling. So here the system, I mean the user may say

universe, I mean Professor works at universities. And then we

have a natural language component that parses this and creates a conceptual model for you. So the system provides

recommendations of.

Of of, of the, the, the models, no. And then the user may change

it or or or not but. But it's a kind of natural language based

approach.

And the other one you you know it you you have mentioned in in

the at the beginning is is Droid which is a system to to support

provide automation for all the faces in in building

recommender. But this recommender should be for for a

modeling language for a DSL. So we support all the faces from

data gathering. So the data should be models because you are

going to.

To build a recommender for a DSL, so you should feed many as

many models as possible.

Then also support preprocessing. No, because maybe you want to,

let's say yeah, fix a strings or remove, let's say items that are

not very popular. Things like that.

Then, uh, an important thing, and I would say that's for us is

the most relevant, is to support the evaluation. I mean, you

configure several possibilities and you choose different

recommendation methods, content based, item based, collaborative filtering, whatever, each method may have parameters. So you need

to choose a range of parameters, a range of methods.

And and then you evaluate which is the best, the best.

And then you deploy and then you integrate with the modeling

tool. That's what we would like to automate. OK.

And yeah, the last question is experience in modeling complex

systems, um, worth, I mean our experience is from an academic

perspective, uh, kind of modeling complex modeling

systems that we.

The complex systems that we wanted. Yes. Yeah. Yeah. I mean

this last question, I mean makes more sense when you know the

group easier to genius, not only modelers or modeling experts.

Yeah. Thanks. Thanks, [omitted].

Uh, [omitted].

Yes, I hope you can hear me. Yes, OK.

Um, yes. So, um, the question. So I think the first one was

about the experience. Yeah, in recommender systems. So maybe I

I can explain a bit.

What we did in the past was for instance to develop a framework

for web-based recommendation systems. So for instance here

was the idea. You have already a website.

OK, uh, maybe this is a website that is not even developed by you and for for adding now recommend or support we used web augmentation, so we focused a lot. How can you adapt the website on the client side in the browser to get a recommendation service. For instance, you have a book shop and the book shop has no recommendation service, but it's a nice book shop and you would like to see what other books. Could be nice for you. So therefore what we did is was a bit about.

Developing tools support in the end having a generic recommendation server.

That was based on collaborative filtering.

Um, that was mostly based on rating certain items. So we had a generic model for this recommendation server and then you could develop different recommender systems that are running in the client side of the websites by the apartment station where you can even use this recommender services from the server. And this was one of our first approaches we we did with with recommend.

Um systems um.

Then what we also did, and maybe this is a bit different um to this web-based recommendation services because here we had a

very fixed.

Model for the recommender service. It was based on rating

certain items no and then using collaborative filtering

approaches.

For the other work we did for recommender systems was about

model repair. So here what we tried to do is to get the

recommender system for textual DSLS for fixing inconsistencies.

And here we did not use a databases.

But we used search based approaches, so actually here the

recommendation.

Benefits was you don't have to think about how you resolve, in

which order, the different uh inconsistencies in which way. So

this was recommended somehow. What the short paths, how do you

get the good quality of your model? Again by using search

based techniques and we integrated this in Eclipse with

xtext. So if you had an xtext based DSL you can use our tool

in the end.

Having OCL constraints on and so on and then you get this

recommended tool for your DSL for fixing inconsistencies. So

maybe this is a bit related to what?

[omitted] has explained before, yeah, but I I think you are

using reinforcement learning, right?

Uh, good then uh, yeah, I worked with [omitted] uh, on Extremo. So

this was a nice experience. But here [omitted] already explained what

what has been done.

And what we are currently working on is uh in in the

context of locomote project is predicting the next modeling

operation in in in the modeling tools by using process mining.

So here we also try to have a reusable recommender that is

utilizing the execution logs you have in your modeling editors

and based on process mining you would like to derive.

Process models that are explaining OK, what are?

Common.

Interaction.

Histories.

And based on this you should get.

Yeah, also recommendations what what you should do next in your

modeling ID when you are modeling. So it's more an

operation based viewpoint.

So this is my experience and this is also my experience with

the tools that we have developed so.

Concerning the experience in modeling complex software

systems, yeah, so.

Yeah. Not, not, not, not sure. We have done a lot here. Um,

maybe also concerning, uh, the the recommenders system. So I

think it's quite important to find out what is a good

reference architecture.

For a certain recommendation approach.

And if you have this kind of reference architecture, I think

then the next step is to automate by using modeling

approaches to synthesize these architectures.

Yeah so so this worked quite well in the past for the

different approaches I have explained before.

Good, thank you. [omitted], you have a questions I see your I

would have a question to [omitted] in this bookshop application.

He said that this is based on book ratings or the recommender

system would, as inputs, take a ratings of books. Where is this

data coming from? Is this business intelligence data

coming from the bookstore itself or is it just taken from Amazon

or from somewhere else? And if it's taking if, it's if it's

business intelligence from the bookstore?

Is the bookstore usually willing to share the data or is there a way that it works without sharing the data?

Yeah, yeah. So here somehow we need the data. So if you

integrate it by calling a service that gets the data from

an external website or web service.

Or maybe the users have to rate their items?

This is uh up to how you implement this rating

functionality. OK so somehow you you have to enhance your web

store whatever you sell, whatever you or whatever you

present could be. Also not non commercial application like we

had a community website with cocktail recipes and so on as

one example. So somehow you need the ratings and you you could

think about it now.

Doing some external calls.

To other websites or services to collect the data or you have to

enhance your website with a kind of rating.

Functionality. OK and then this data has to be sent to your

recommendation server.

And and then you can can use the recommender. So as maybe this

also relates to what David I said before so this this data

integration aspect or getting the data in the format.

That you need for your recommender and even getting the data from your users. This is a huge challenge, yes.

Exactly, yeah.

OK, yeah, what? We also developed recommenders like the

search based approaches.

Where you don't need data.

Umm.

OK. Thank you. Thank you very much. Umm, OK. So I would since

we are, I mean when I worked there one hour, I think that

they would move to the next part of of this group. So Claudia,

you want to share the screen? Yes, sure, I'm going to do that.

OK.

Uh, OK, let's see. You can see my screen, yes. Uh, OK. So, um,

to cope with the challenges that we that we presented, we propose

this tool that is called foreca. Let's say a local the prototype,

design, customize and actually deploy. The actual system that

they use are as specified.

The treatment point that we we would, we would to cover is

first selecting the.

The crucial feature of the one the the customer contact system

that the user want to, I mean to to deploy at the very end. And

to do this we set up a set of logical constraint among the chosen feature. Of course this feature are based on recommender system for software engineering, because I mean we can cross miner. We developed, we developed a system.

More developers essentially. So not based on our let's say knowledge and of course we reviewed, we reviewed also notable works in the domain and the final step is the generation of the source code for now in Python. But we'll see how we gonna stand. OK, when we conceived, I mean that the left for rec approach methodology.

Say, uh, we we we identify first a set of challenges that these are, let's say an extended version of the open challenges that we presented before. So the first one is how to specify a common pipeline of recommender system? Because I mean not about recommendation. There are there are a lot of approaches in the domain, not only of course in the engineering domains of development, but.

Umm, and so the the one problem is to identify the common similarity, the common, the common components and the similarities among them. Then that would be a very very problem is the collection of the data, because their system are basically all information retrieval tools or.

Presently we we see that the proliferation of machine

learning based approach, so the data is crucial task and they

usually recommend the system needs needs a lot of data to

perform proper.

To retrieve relevant items then.

Uh, then we have, of course, the the development of.

Scored components that are mapped to them, modeling one and

at the very end to evaluate the system. Because there are a set

of metrics and a lot of evaluation strategies, but not

all direct niques are suitable for all for all.

Different type of uh, recommender system because for

instance.

Information, uh, collaborative filtering methods have his own

metrics, for instance catalog coverage.

The sales diversity and so on. Uh is that uh, for instance, um,

machine learning domain are usually evaluated with

precision, recall and.

Uh metrics and so in in the in this field we have to cope with

the progenity of this kind of things. So this is the

architecture, the overall architecture, how we address

this heterogeneity by means of feature model. Feature models are used especially in software, software product line to I mean

to set the general feature, the crucial feature law system.

Uh and uh. Of course this kind of model supports also logical

constraints between amalgam and the different components, but

for instance, if if I'm, I mean if we have some requirement of

the data we we have to take into account when we we have to

choose the corresponding algorithm for instance.

We have a particular, for instance a collaborative

filtering. In collaborative filtering we have as a data

structure usually metrics, user item metrics with the ratings.

So when we design a collaborative filtering.

We want to use our collaborative filtering algorithm. We have to.

I mean we we have to encode data in some way and in particular in

a matrix format for instance. This is just an example of a

constraint between data and then we have also intermediately.

The outputs of each component. So we have carefully designed

the constrained and among the components. This is the first

phase and we have a second, second refinement. Let's say

when we selected the feature, we can we can optimize or customize

the algorithm that algorithm each component actually that.

We we selected the, the, the, the. I mean in this case they the designer selected in this case. So in the, I mean in the,

in the next step of in this announcement of the system we

can select for instance.

Yeah, but parameter of machine learning model or for or for

instance if the collaborative filtering is user based or item

based and the mini many other aspects of for instance

evaluation methodology on in which way we can test this

system and so on that are not in the final step. But as I said

there is a generation of source code.

Manager of the developed the development, we we generate the

Python code for the system and in three, let's say different.

Three different using the three different formats.

That's the notebook, because it is more readable. Let's say for

the I mean another user. The flask web interface always

written.

Also written in Python And playing the Python code to test

on your machine. So this is the future model the dimension, of

course, yeah, for space, I mean for the sake of the presentation

we collapsed some feature, but I can show you the extended model

in the demo. As you can see we have elicited the components of

genetic, genetic, potentially genetic recommender system.

Uh, so the algorithm you see the algorithm component that the

data set component in which we specify the different data set.

For instance if it is a supervised or unsupervised does

it for machine learning problem. Then we have different classes

of algorithms like for instance availability, filtering, content

based algorithm, classification, neural networks and so on. And

also also an aspect that actually I don't, I don't

mention.

Indeed, the management of the presentation layer, let's say of

their communities.

In a word, how how we we present the final recommendation to the

final user? Because also I mean the interaction of user is a

crucial component. And then of course we have this list of

course we I mean it is our ambition to implement all all

the the components but.

Uh, I have a request. Maybe? Yeah, accepted the request. OK,

OK, perfect.

OK, and and of course ER, we implement a subset of this

measure for this demo, but let's see after this and they they

constrain the logical constraint and dimension. And is a subset

is represented by by this table. For instance, as I said before, if we select an unsupervised algorithm, we cannot we cannot

select supervised the data that because the data.

But not in the right format for this kind of algorithm. And then

we have for instance in if we want to use a another language

preparation, since we cannot do it if the data are not included

in a textual format. So.

And then of course we have to take into account all these

aspects when we have, I mean opposing and building our

recommender system.

OK, we will do in the second phase we use the the concept

coming from a model driven engineering that is meta model

actually model is a model in general is a presentation of

real world entities and the middle model, middle models give

us the concept to to represent a model of the system. So you see

that here we have different we have.

Different.

Concepts. Let's say abstract concepts that represent the real

world entity. In this case we have the the concept of data

set, the concept of algorithm, recommender, system itself and

so on. You see also that we conceptualize also the Python

libraries that we use. And of course also here we we actually implement the two of.

Of this library, but we can we can discuss it later. And in

this case, just to end, the meta model is used to in the second

phase of the process, so to to enhance the initial

specification that is, I mean the initial feature that the

users selected in the first place.

Uh, in this phase? Uh, so.

Umm then that space is driven by acceleo. That is a model driven

also, yeah, a model driven tool that is capable of generating

source code by using the templates. So here we have an

example of accelerator template and as you can see we can

specify not only Python. In this case we have Python code but but

we we can specify we can generate.

Source code for every kind of language because it's a model

driven tool. So depending on the.

Affected feature that the user as a specified before we can

generate the corresponding code. For instance, here we have a set

of, let's say condition and.

Uh, we use model concept to generate the source code and of

course to tune the the recommender system in the end.

So to evaluate, I mean the capability of left or right to mimic the.

Existing system we we reuse as a use case to different approaches

but we developed the indecorous manner Vortex one is cross

record that is our collaborative filtering based recommender

system for third party libraries and basically we have matrix

stuff structure in which the recommender system takes the

request from the user by using the by by using the the GitHub

project.

And that is the abandoned. This actually and recommend uninstall

by items. So to set I mean to mimic the prospect

configuration. Of course we have to select.

You have to select the filtering algorithm. You see that the ER

we have the representation of the Fisher model as a set of

checkboxes, so the user can easily interact with the the

corresponding feature and you it. It is worth noting that the

number of manually, I mean the feature that user the user have

to select finally is very immediately because thanks to

the.

Vertical constraint that we define. We define it that we we

can. I mean this is then we can automatically select the

remaining treasure for based of course on some choice of the user. And then we have the second phase represented in the

right side of the slide which we have. We can fine tune the

system. For instance, here we can set the library user for the

generation the.

Been the dipole evaluation, the number of recommendation,

recommended item, number of folders that they devaluation

for instance. And you see that of course we can customize every

aspect of the system, not only in this evaluation. And this is

the final generated code for CROSSTREK in Python. So you see

that we have a complete implementation in the actually

working implementation.

The second I mean the second use case that we use, it is a horror

right that these are machine learning based the classifier

for meta models. So we selected this this second approach to to

show that the that the Lefrak is capable of managing the

different algorithm and strategy because.

Crosstrek is based on availability filtering, while

here we are using a neural network to. I mean to classify

media models in this case but can be.

Also other other different areas of the artifacts and also

similarly similar to Prozac, here we have the two phases. One is the feature selection, yeah, you see that we go between

deeper by selecting for instance in this in this case a

supervised feedforward neural network and the in the second

phase we again set the other parameters of the model.

Also here we have the generated code and to evaluate I mean we

all we all success. I mean the the quantitative, quantitative

evaluation.

See that the aspect of refractor by running a personal validation

in which we demonstrated that refractor resemble in some way

the the original results of the tools. Because we have, I mean

more or less the same.

Uh, the same results on average. So OK, I guess which? Directly

to the plug in, no?

I have, though of course I have a demo that we recorded for

Rex's conference.

And you see, yeah, we what we developed is an Eclipse plugin.

And you see here that we have.

See the complete feature model which the user can navigate.

And.

And here the user can can specify a new configuration for

instance here we want to mimic across direct for instance and you see that there is that produce this kind of file that

is based on visual model. So the user can select what were they

the algorithm and each component. Usually that also

there is some show they go strains so the user.

And are driven in some way in this election.

And though of course they are OK, we can select also the

presentation layer, component data, also the data set, the

pressing data set library also written in Python And also I

mean the web library for developing.

This is uh and so on. Um.

OK, so we can skip a beat that. OK here from the from that we we

automatically generate a model that is the second phase. So

this model is compliant with the meta model that I showed. So you

see that error.

This is some generate an initial specification of the system and

the user can refine it. By using I mean the this modeling

environment in this case. So you see that in cross fold

validation we can set.

We can set for instance the number of folder that is used at

the.

In the evaluation uh settings and uh the the final recommendation. Also the the final recommendation algorithm

and the CER you can customize every aspect of, in this case

the collaborative filtering system. We can select the number

of neighborhood and so on.

So OK, OK we can skip to the the generation part and from this

model we generated the Python code as we can see from the

video. So here this is a running code to I mean to deploy capital

system actually.

So, and this is the running system in the Jupiter notebook

environment. You see that the code is.

Uh, it's actually the lawyer that.

On Amazon, OK. So Umm the same goes, I mean the same process

has been done for Aurora. What is different I mean is that the

as you can see the, the, the, the future about the I mean they

they chose an algorithm so we select a feed forward network is

different to the library that we use it because we use to learn

to generate it. But the, I mean the.

Should see is the same and uh we uh we also have uh, let's say.

In this case, we have Bob's Plain Python code to test in our

ID for, for instance.

So, OK, coming back to the presentation.

Um, OK, OK, we developed this Eclipse plugin, but what we want to do is to, I mean make Laforet more platform independent in the sense that.

We want to. I mean to.

I mean they they improve their usability also by means of for instance a cloud based architecture or for instance a standalone version at his daughter. So he's in this way the system is not bounded by let's say a modeling a specific model in a framework like clips. And actually what we did already is to generate a web editor in which the user can select.

Um.

And again specify this.

The the the system, the entire system using the uh.

Well, basically generated from the meta model. So you can see here that also in this I mean in this web auditor that we generated starting from the feature model and the middle model we have also the context assistant, how to complete the complaints from the issues and so on. All I mean all all features that can help the user, I mean to better to better design.

And the best the, the, the, the system uh, so just to summarize, so we developed the model driven based environment let's say

following the logo to I mean to design possibly every.

Any class of recommendation, recommendation, recommendation

system and of course.

Uh, even though I just show the uh, the red part, generally the

coding Python, thanks to the acceleo and techniques

techniques we can generate tools. So Java implementation or

C or whatever and.

A11 point that we we would mention is their extensibility

because if I wanted to decide to I mean to introduce a new

component for recently in the data set, the preprocessing or

in the.

Uh, algorithm a new algorithm for instance I I have to just

two to extend the two models. So first if we should if we should

model and the the meta model that can drive in this.

Uh, in drive the user during the specification. And of course

what we want to do also is to develop actual, actually a local

the platform in the sense that we can enhance the, the UI, the

user interface with.

The classical uh breaks of local so blocked.

And that.

We we can uh, we can also.

Not an agnostic environment to compose with the tracking drop

utilities the the pipeline and of course we can support, as I

said, the different systems and libraries by introducing by just

extending the model. OK.

Uh, we are, and we can discuss this.

In this kind of research question about more now of

Lefrak.

Yeah. Before doing that. So I would like to thank Vasta for

joining us and we'll ask him to, yeah, to do just a short

introduction about yeah. But if you can say a few words about,

yeah, what you're doing, where are you from?

Yeah, sorry for being late. I'm [omitted] and Emily, and I'm from

the Polytechnic University of Bari and I'm a researcher there

and I usually work on recommended systems and

knowledge representation. So in the last, let's say 5 to 10

years, I worked on it and.

We barbish the AT Rexis mostly and international semantic web

conferences, so we basically work in this.

Great space in the middle between knowledge representation

and semantics and recommendation systems.

Well, I tried to to make it as, uh, yeah, as compact as

possible. I'm not one of the.

Although the recommendation frameworks, yeah that was

presented before and for what regards the recommender systems,

I mostly worked on factorization methods, deep learning methods

and evaluation of beyond accuracy dimensions for

recommender systems.

OK, thanks for the digging introduction and because we we,

I mean we this this.

Meeting here was, I mean essentially consists of two

parts. So the the first part that unfortunately you missed

was about, yeah, presenting what I mean the other challenges

about now development of of recommender systems, existing

tools. So of course this part is, I mean you are aware of all

of this. But at the end of this first part essentially we had a

quick discussion where each of us essentially answered a number

of questions about what is your experience with recommender

systems in general and actually you already.

And you have done this during this presentation that you have

just done. And also we had another questions about was this

your experience with the tools that supported the design of recommended system. So during this first part of the meeting

we also mentioned the Elliott. So the, the and the tool that

your group I mean you are you guys have been working on to

support the evaluation of recommender systems now if I'm

not wrong, so if you can say.

Also, a few words about, uh, yeah, the experience for the

supporting tools. Yeah, yeah, exactly.

Yeah, yeah. I'm not I I'm buying. No means an expert of

tools used to create recommended systems since we usually.

Write code from scratch for any recommended system. We use and

then we integrated in the framework in the last few years

and before we just.

Write a code, check the reproducibility of the

experiments and then use it.

Uh, our only experience with the these tools are the the

connection with you guys that are devoting a lot of effort on

creating these distances.

Yeah. And then then I think that yeah we can continue with the we

do about answering these questions that we have here. So

what do you mean what you can say about given ability or I

mean do you think that it is useful tool like for left for rack in supporting development of a customer recommender

systems, you can say some comment about this. Yeah, I

think he is because when when we try to.

Let's say introduce recommender systems to the to the students

or to the stakeholders when we have project with companies.

Let's say there is like a wall.

Between what they think I recommend the system means and

are they actually work so and what they need what they

recommended system need to to work properly.

So these kind of tools, I think that they are.

Let's say uh very very important to to lower the the barrier and

to make also these small companies, let's say some

startups or.

We had the opportunity to to work, be it with the BBC team.

Let's work on uh on the recommendation engine and uh.

And they had the same problem. So these kind of tools are, I

think they are really, really helpful for them and obviously

on the one side we should have somebody with really in the

development of recommended systems and the structure and

framework all these.

Much later models mainly and on the other side obviously, and another kind of subject.

Or audience with the different skills? Or or just The Who wants

to run experiments without going too much into the details?

OK. So, yeah, very useful. Thank you. So this is essentially, so

you answered about this usefulness of tools like Forex

for you know, guiding and supporting the development of

recommender systems. Now you all followed, I mean you enter the

just in time while when Claudia was presenting Laforet, maybe

you missed a couple of minutes since the beginning. So

according to what what you have seen?

Uh, can you comment about. Yeah, about what you have seen about

the tools, the tools. So you see that the tool now rely on the

kind of reference architecture. So this was what manner was

mentioning at the beginning. So essentially we rely on the

availability or we design the reference architecture

recommender systems and we try to automate a different process,

no. So can you comment about what you've seen, how the left

for like you know?

Support, uh all the different phases and by by considering I

mean what you have in mind in terms of what would be an ideal

tool for supporting the development of recommender system. So can you comment about what you've seen and what you

would expect or what to improve or extend the tool that you have

seen in order to meet your expectation?

Uh, yeah, yeah. I think that left Foreca is is very, very

useful and it is general enough to cope with the different

recommendation frameworks and we know it. And I think that, OK,

it is evident that a lot of effort have been put in making

it so general.

Uh.

I think that different groups that want to use Lab 4 maybe

some.

Some, some bees, some part of the audience obviously would

like to.

To just use the tool as it is with the different interfaces

and.

And the field the phones up and and then uh, run uh, run the

experiments and everything.

But I think that other.

And the group's maybe uh could use Lefrak as a I am by no means

an expert all software engineering, but to use lab

forak as a.

Low level layer.

And developed their own interfaces to do that, like a

mobile app or a specific web app to use.

In their company.

With the specific configuration for some models and imagine that

in this case, in this case they would like.

Call the specific API's of these.

Low, low level, uh, leverick, so D engine, let's say of love for

work and OK, I think that I answered a couple of questions.

I yeah, the others, yeah, I mean, I mean you said there

actually essentially what you're saying is that maybe the

graphical interface or what you're proposing as a front end

for the user might not be, I mean satisfy some users. So what

you're saying or what you're proposing is that keep the core.

As they are and may be exposed to different ways of selecting

the features or customizing the models that you that we

generate, let me let me make an example.

The BC group was working on mostly.

On content based and ibry, Dirk condition models. OK so

basically.

They were.

Mostly not interested at all in a lot of different collaborative

filtering methods, but for what regards the content based and

the IBRD models, maybe they would like to have a specific

preprocessing steps.

Before a feeding the recommendation systems so they

they would integrate. These are these steps within.

A specific proprietary framework or interface through that and

then run under the hood.

A lab for act to to prepare the code for the experiments.

OK. Yeah. Thank you very much. Thank you very much.

Um, yeah, I would I would pass to [omitted]. [omitted], can you say

also your your thought I mean share with with us what do you

think?

Yes. So sure. So it if it's about these uh these questions I

think it's if from my perspective I would say the

short part of the demo which I the short part of this work

which I've seen now looks very promising and very nice. The

only thing is that I would say I would need much more time to

answer all your questions. Yeah. Yeah. Yeah. Yeah because as you

first noticed you you said it yourself I've I've not been

involved in the in the local Mode Project Alliance this is.

To me this is the first time I'm seeing this and the only thing

which could come to my my mind is that usually when you have a

long long list of configuration things which you show cloud you

showed in the in the demo.

It might only question is whether there was any default

variant. So if you say if I don't know anything, I just go

for the default, is it possible?

Yes, it is not. I mean, I mean this is not done now, but you

can of course specify default to the feature model of course.

Yes, for sure, yes you can.

Uh, yeah, also for instance in the in the web-based data that I

showed. I mean it's still a prototype, but we can drive, I

mean the user with also I mean some for instance for

information about the different algorithm, for instance

collaborative filtering is suitable for this. So maybe for

your problem is.

More suitable than another, uh, recommender system for sure. For

sure we can add this. Thanks for. Yeah, I mean that that's

all because I mean in a point of time I think I saw it was like

300 or 400 different configurations which was

possible. So, so for if it's a.

I mean to make it easier to use and user friendly, I think it's nice to to have these custom uh, this default things. Hmm.

Otherwise I think I I would I'm looking very much forward to to see more about uh left for rack because I think it's something which we could use in different projects. But for now I don't know so much about it to say anything about.

About it. About answering these questions.

Yeah, for instance, what do UM.

It is possible for instance to to extend because currently we we don't have the support for enforcement learning now.

Because I mean the idea of using or extend your tool is that. I mean currently we support a number of techniques and for each technique we have the off to the corresponding implementation, meaning that we have a library for supporting those. So extending the tool means that we introduce a new feature. So the feature can be reinforcement learning. Then we have and this was for the feature site, then for the metamodel.

Side of course we needed to introduce the meter passes to consisting of the parameters that you might want to speak. I mean to to specify in order to customize no and you to completely enable the the usage of the forcement learning. And

then you generate of course the code that according to template

that of course you have to define. But that's the way of

this is again from the developer of the site, but the user from

from inside is just to take and pick.

The reinforcement learning specified that the the the the

parameters and then generated the code and use it. So that's

the way.

Then it sounds sounds really nice and from the user

perspective could be.

I mean with with these simplifications could be

realized and I mean it's, I mean with extent, I mean how

difficult would it be for a newcomer. I mean these kind of

things I would say it's it's more suitable to to to to to put

it in I'll say this SUS I don't know what was signing for system

under study scale something like that. I think it's called

because they are like a list of I think it's 13.

Um, predefined questions which you can somehow modify a little

bit and then send it broadly to users of the system, right?

I think that that could be an idea to get answers on this some

estimates.

OK good [omitted], you want you have a question. OK, so at the end you are proposing to have a kind of meta recommender that is able to set the configuration for a specific problem. So what that means you have a specific problem and you have to choose if you would like to use machine learning or collaborative filtering technique. And at the end you are proposing to have made a recommender system that is able to define the configuration.

So the problem is that a default configuration is.

We can we can define that default configuration, but the problem is that the the configuration really depends on what are you purposes or what the recommendation should do.

So for this reason maybe we have to implement some things that task to the, the, the, the, the. The purpose is to the to the user and then depending on the purpose is to recommend the possible configuration.

Yeah. I don't say it's easy. I'm just saying yeah, yeah, yeah, yeah. But it is related also to what?

Thing so essentially we are discussing about an alternative way of making the tool, I mean of using the tool. So essentially we can use a natural language interface and sensory use. Specify in another language what the recommender system

should do and maybe my answering a number of questions or can be

by means of web-based or DSL. So I see this comment in line with

what [omitted] was saying about this point.

Thank you. Thank you.

[omitted].

Yes.

Umm.

I think it was a very nice presentation. It difficult to

answer those questions also because I I'm not a software

engineer and I speak a little bit of different language. So as

I said, we have difficult personas within Siemens who get

in touch with recommender systems. So there are people in

predevelopment who look into the technology, technology look what

has to be done to bring this out to the customer.

So for example define the technology and on ontology

estimate how much effort this is to bring this into the to the

product. Then there are product developers who really implement

this properly into the product. But you have to understand these

are products which costs.

10 thousands of EUR. So the the, the quality has to be really,

really high. So they probably have are hesitant to adopt to.

Two novel technologies.

Unless customer usage is already shown. Therefore there is a third persona, namely our engineering services and customer support, and there I think this can be really useful.

Because they are.

There are also innovation topics like recommender systems where people we get in touch with the customers understand their their need, have their hand on the data.

And quickly have to build something up which would would solve their their customer pain.

And and for those things or in for for customization of the of the recommender systems I think that's that's useful.

For the sorry if I can comment what you're saying, just sorry for interrupting you. So I I I mean I go back quickly on what you said at the beginning. So in the sense that you first develop a proof of concept, typically Python scripts and then you actually develop the real product once you are satisfied with the proof of concept. So does it make sense for you that we can say left for rank and use for the first part for the development of the proof of proof of concept. So essentially we we use left for rack for supporting the development of

the proof of concept.

And if you are satisfied with what you get out of this, then

you can, you know, translate or move towards the the real

product. Does it make sense? That makes sense, yeah. You can

quote me on this and so for.

Setting up the proof of concept, um this can be useful, but it

can be also useful for the engineering services when they

apply this for different customers because the data

situation might be very different and therefore the

model might be a different that you want to or the whole

recommender system backbone might be different.

And they are. Having this vast amount of options available can

be helpful. Although I also agree on having that would be

helpful to have default settings which help our engineering

services then to quickly choose them.

Good. Thank you.

Uh, [omitted].

I have to. Yeah. So.

Uh, yeah, thanks. Thanks for the presentation. I was quite

interesting to see what you have developed in general. I really like your approach going from the feature model to the EMF

model, so to have this different configuration stages and I think

it's also meaningful the languages you have selected now

first the feature.

Modeling approach because here it seems you have a very fixed

configuration space and then you have the modeling approach that

allows you a bit more input from from the user side.

So it it seems to need to be very useful, um, what I was

thinking, could you, could you go back to the two examples

quickly you have shown, yes.

The example you may need this, yeah from the configuration I

guess. OK yeah case case studies. Yeah exactly. OK so you

you had so.

I was thinking.

Uh, because the first question is about the the usability.

Or could you use costs? Uh, uh, your your porch. Could you use

your approach for to apply it directly to your approach?

OK, yes, OK meta live for rec. Let's say yes, because it it

seems like for many steps you have to do like also what [omitted]

said before, not the default values, but it would be

interesting, for instance now if you select no certain features, you could explore someone else before already also selected

this features, selected other features.

And especially if you go from the feature model to the system

fine tuning on the EMF model site here this this could be

very helpful. Maybe this could be a very nice case study in the

end.

Yeah, yeah for sure we can improve. I mean the the

graphical interface, so we can provide the as we said the user

with the default configuration based on the problem on the

recommender system that the user want. But definitely yes is.

We can, uh, we can apply, I mean this this sort of automation

also during the feature selection, I mean to improve the

automation during the feature selection for sure. And also I

mean come back to that and then and then for the next phases of

modeling, let's say, you know if I can add something related to

this when we are selecting the feature or for sure the feature

model and also the.

Modeler in this election because when we are selecting some

things, some feature will be the the the selected some feature

will be unavailable. So depending on the configuration

that you are specifying with the feature model, the other parts will be adapted on the selection of specific features. For

instance if you are using the user based. Sorry if I'm using

unsupervised data set, I cannot use supervised approach. So I

kind of guidance.

Or did the user is provided by by the, the, the the constraints

that the cloud you showed during the presentation?

Yeah, I think the future model makes perfect sense, right. So

this is very helpful you know, because it really helps you to

make mistakes and and you also what you have shown now you get

this automated selection and so on. So I think this is really

the appropriate formalism.

Then there was just thinking maybe on on the EMF model side.

Yeah, maybe to find some features, standard feature

values or maybe maybe feature values would have been used

before. Could be interesting.

So I I would say the usefulness seems very good to me.

Uh, let me quickly check the other questions that you have.

Yeah, yeah, yeah. They are quite detailed, no question 2.3 the

the extensive.

So it seems automation level is quite good in so here I have.

I would say no further comments. The only thing was, uh, maybe I missed it, but if you now deploy your recommender system.

Could you even uh generate different variants and then see

what maybe works better?

Uh, yeah, actually we can do it, but not in parallel because

actually, but I mean I guess we can use acceleo to deploy

different as you say the different version of the same

system and test the better I guess. I mean there is the exact

work that one has has done with the Droid because Droid have

this picture actually to test.

So that's the barrella differenter Quander system. No,

definitely we cannot do this this special and I mean it's a

very good suggestion, because now yeah actually we deployed

just the system and then the user can refine after, etc. The

results. But we if we can support the default

configuration, we can also generate in parallel different

specification of I mean the same system with some.

Clients, as you said. Yeah, yeah, I also in in this respect

I liked before very much the comment about the API approach,

because now if you generate such a recommender system and then

you would like to do changes.

Some fine tuning. How? How is it currently done? Is it done on again on the model level and yeah and then you reach enerate

again? Yeah. Can you update the recommender system also? Exactly

yes, yes, yes actually yes, yes. So DevOps or Lifecycle

management is not supported yet. I mean, and I think here the API

could be a handy tool because maybe then you have the specific

API where you can do still some.

Some configurations or changes during runtime in in your

system. So I think this this could be a very.

OK yeah, yeah he's, he's.

Uh, useful. Such as some. Yeah, you know.

The the last question I did not fully understand about the

impact of the meta model evolution. What what do you mean

here?

Ah, sorry, uh.

Is actually a mistake the impact of.

Especially the recommender system, let's say.

Yeah, no, this question is to I mean to assess something the

usability of the use of the interface for unknown expert

user of.

Uh, recommended system, let's say, because uh, you know, for

instance.

Umm, a user uh, that is not unexpected domain, maybe to certain point could be lost among the possible choices and then we I mean with the feature model we we tried to to drive him to this, but I mean is that is more on the on the usability considering an expert user.

Activity in in the end. So here you talk about how how easy it is to learn the tool, right? Yeah exactly yeah. And here I would say maybe it makes a difference now if a newcomer to recommender systems at all or if you're already an expert in recommender system and you use the tool so you're only a newcomer to your tool. I think if you're already you have experience in recommender systems, then of course I think your tool based on the feature model. This should be for me.

Easy to use, but yes of course. Maybe you.

Doing some studies in your future could be interesting here, but if you are completely newcomer to recommender systems then maybe it would be a good idea to have a come some kind of explanations about the feature models and guidance.

So yeah, OK. The really the newcomers to recommend US systems, um can can, yeah. It's, I would say more about learnability now that you have some documentation and this kind of thing, OK.

Or maybe examples, tutorials and these kind of things. Hmm.

OK. OK, good. Yeah, thanks a lot for the presentation. I really

like your approach.

Thank you. Thank you. Thank you, [omitted]. Before we move to one,

sorry and I don't want to skip the line, I just have a question

maybe for, for, for afterwards then.

OK, just finish then the, I mean this round and then we can have

another one, uh, one.

Alright. Yeah, yeah. Again thanks for the presentation. I

also like the, the, the approach. I like the the idea of

starting with the.

Rough, rough configuration. Uh, the features. And then you fill

in the details, like, uh, numbers, neighborhood sizes and

all the details. I think that that's I I like. I like that.

No, not enjoyed. It's a little bit different. We use a DSL from

the beginning. But I also agree with the comments of [omitted] and

[omitted], no. So this is going to be used by newcomers.

Then maybe already seen so many features. Uh, it may be a little

bit scary, you know. So maybe you have a you may have a kind

of with art that goes step by step and then maybe it produces a like a configuration that may be suitable and then later you can refine. Maybe that would would be would help. I don't know. And I also agree with the comment of [omitted] yeah, I I I like the approach that you generate.

There's something intermediate EMF models in this case that the user can fine tune. I I like this approach, but also it's a little bit risky, right? Because then if you use the three editors you can remove objects and objects, um, do things that are not meaningful and then you generate the code, maybe it's not working anymore.

So, but I guess that depends on the interface, no? So you can provide another interface, yes. Exactly, yeah.

So that's why we are moving forward, let's say more web-based, cloud based architecture. So I mean the user cannot, I mean in the sense that we can drive better the user also in the modeling, I mean in the second phase let's say and also in the future as you say. Yeah, yeah, for sure. Maybe we can also constrain the editor in order to just modify the parameters that is not specifying the feature model.

So in this way we can make sure that the the model is still valid after uh, specifying the the the parameters.

Yeah, yeah well probably if the user is is a newcomer, probably something else than than three editor would be, yeah. Because otherwise, you know, if you are not familiar with the MP3 editor, it's a little bit tricky, right? Yes, yes.

And then yeah, so the level of automation seems uh, I mean it's automatic if you have the right components. I think that to combine and you have the right code generators and you have all the components. I think no, it's you provide full automation. So I think it's like here my question is I didn't fully get,

I think David was mentioning that but I didn't fully get that. So if I want let's say I want to have a recommender let's say for.

The programming language rust or something like that.

Then what do I need to provide? So I need to provide the kind of encoding of my face. I need to to find a repository of Ras programs and and and and embedding of these programs into some vector representation.

And then I come from existing components or what can I do?

Wait, uh um.

If you're saying that you want to implement your recommender systems in the in the in the new language for, currently our target language is Python, right? So no, no no no no. I I

don't know if I want a recommender for rust programmer.

So a programmer is building programming and I wanna

recommended for rust.

No, that's OK for me.

Yeah, you what you wanted to watch. OK. So for sure you have

to provide the vectorization of the data. Then you can apply the

preprocessing in order for instance to remove useful

columns or rows or to aggregate columns. So we are providing the

way for aggregating and manipulating the data. But at

the at the beginning we need a vector based representation of

the data set, right? And then at that point I can use all the

components that you already have now in your. OK, yes, yes.

Yeah. So sounds sounds very good. No, I mean sound sound

reasonable not that's um, yeah, let's say uh, yeah it saves

effort, no. So I think it's let's say that the the data set

I mean the generation of and I mean the data extraction is not

fully covered actually, but we are working on it on it for

instance and now we can, we can, we can handle fix it, I mean

structured.

Um, data like CSV file, Jason. But also we can add let's say a component that that may be a mine, for instance Rostrevor story to. I mean to support also the data preparation, let's say data preprocessing, preparation of genetic data set. Yeah, it's interesting piece of work for sure, yeah.

Yeah. Well then concerning. Yeah. Well it's only I think it's quite useful. No. I mean it's provides automation once you have everything in place and then extensibility here I'm, I'm also not sure because yeah to me a feature model is is is allows you to to work with a close variability, no. So all the features that you can choose are there if you want other features.

Then you need to modify the the feature model and then probably to to add the code generator or whatever. So yes, yeah, it's extensible, means you can understand something without touching.

What already exists? OK, if you need to touch the feature model then yeah so so probably an approach based, I don't know extension points or something like that where I mean because otherwise you as a design you, I mean your group needs to extend it or I need to dive in into the code for extending it.

That's the point. I mean, yeah, yeah, it is done in a way that is extensible because it's a feature model, so it's more or

less.

Know where you would need to touch and then you have this

accelerate scripts, but maybe it would be. I don't know if it

would be easier to extend if you provide a kind of possibility to

extend it externally so that you provide extension points for

your. You have a number of features at the top, no? Like

the the recommendation method this and that so that I can.

And so the new feature as an as an implementation of an

extension point and you you can let's say read those

implementation dynamically and not the features to the feature

model or something like that. No, I don't know.

These are just my my thoughts maybe, maybe doesn't make a lot

of sense.

Uh.

But in in any in any case I think it's, I think it would be

this approach is more extensible than if you would. I mean the

way you have done it done. If you you would have written it in

let's say in in Java or well maybe written in Java. But you

know you use templates and each model model so it's it's a it's

a more restricted way of extending the tool of course.

But yeah I agree that maybe having a kind of a plug in.

Because at the end of the day, of course you you have a mental

model behind.

And it's it, it it's a question about how extensible is that

metamodel, what the metamodel is, what it is.

Uh, you can extend it by adding new meta classes.

But if you want to be truly extensible from outside, maybe

you need a kind of.

Of plugin or feature of or here.

Mm-hmm. Yeah. The the other thing that is a suggestion that

I, I think someone said I think was it [omitted]?

Probably as a someone that is going to use your tool to create

a recommender, maybe I would like to try out different

configurations and and maybe yeah compare the performance or

something like that. Yeah like you have Android right? So at

the end.

Yeah well maybe I suggest that because it's what I have seen

and yeah, but sometimes it's really useful not to to yeah and

yeah how difficult it would be for a newcomer. Umm yeah I don't

know. For someone that don't like to code or is not familiar

with this Python libraries it would be very useful you generate the code.

For someone that is really really newcomer is very. Then

maybe he doesn't even know about what are the. What does the the

recommended methods mean? And probably would need something

like a wizard or or a kind of default configuration.

Uh, so it depends on what do you mean by newcomer. But but yeah,

of course this automation is very useful, no doubt.

OK.

Thanks suhan. Thanks a lot.

Then, uh, [omitted], you want to say something you wouldn't

question? Yeah, I have, I have a question, but it may may take 5

minutes or so.

Yes, OK. OK.

So I'm going to talk about one use case. Please don't quote me

on this because it's it's current development.

So we have the situation that we want to build a simulation

management base where the customer puts all the simulation

that he or she runs into it and we define a an ontology to

describe those simulation models and context like what are the

requirements for the models. Now if someone in the company wants

to do a similar simulation, we want to recommend from this database.

A simulation which was very similar.

For the for the use case given. So maybe the design has changed

a bit, but setting up the simulation tool.

Requires lots of work because you have to do many steps

meshing configuration and so on, and we want to help and

therefore retrieve.

And an entry from the simulation database.

Umm.

Which fits best to the current needs. So usually we have this

scenario. We have a complex ontology and very few data

points.

Right, so we query and then having some kind of similarity

score which which assess whether the simulation fits to the the

actual needs.

Do you think that for this data scenarios your platform will

help me?

So, complex ontology and you said you need to have the

vectorized the data in some kind of vectorized form. OK, I can

extract a CSV file, plug it in and then I want to train or.

Test the workflow.

And having them a a variety of similarities, measures and so

on. This. So these are these are my needs for this use case.

There are other ones where I have more typical machine

learning models and there I see that this.

This would help, but also for this I have a complex ontology

and few data points. Do you think that your platform helps?

So let's put in this way so the platform simplify.

The typical activities that you have to do [omitted]ly when I mean

when I refer to activities essentially I say you have to

select the the machine learning component or the technique that

you want to use to to to to you know to run your recommender

system configure this component.

I mean who told the WD blocks together? So this is what the

the the platform gives for free or support, right?

Uh.

So this means that if you can develop your scenario, the

scenario that you're seeing, meaning that you can take your

ontologies and vectorize your representatives ontology in the

way that.

Is needed by.

To collaborative filtering the component or to training specifically on the neural network. So if you can do this.

You know, [omitted]ly.

Again, I think that the the platform can be employed as long as the libraries and the components that we characterize support are enough for developing your case. I mean [omitted] listed the number of libraries that we characterize support. So that's a surprise. We for the processing we support the pandas and Numpy, so as long as.

You you you the components that the libraries that are there are enough for you then you can use the the platform as it is. If not we need to extend and then all the discussions that we had so far about how to extend the platform. So currently this extension is not supported as a in eclipse. For instance in Eclipse VS Code where we are this extension mechanism. So currently we do not have any extension mechanism.

At that level, but the system is engineered in a way that we know what are the components that need to be added that without disrupting the rest. OK, OK, so I don't know if I answered the question. No no no no it it helps. So a certain amount of pre work has to be done.

And um.

Maybe. I mean if a complex ontology is in front of it, extracting this and then in case the available packages which are implemented so far are not sufficient then one had one would have to extend this. Yeah OK yeah. But to me this is also related to the questions that one of the questions that have done at the beginning for you about the usability now. Yeah because I mean this is a pre work that you can do, you do once and if in a similar cases you can reuse the components that you have previously.

Yeah, yeah, yeah. Therefore I meant I I see the biggest value for Siemens in engineering services customer support where it's about customization the model after all these difficult pre development is pre work is done like defining the ontology, the data structure and so on.

OK. Thank you very much. Thanks. Do you have a questions about this?

Ohh yeah.

Would it be possible to?

Uh, treason over this anthology.

Let.

Me. Um, in the recommender system research field. We are wondering, since two years ago on 2-3 years ago, about the

fairness of the recommendations.

So what would it be possible to associate to these ontologist

facts related to the to the performance?

Of specific groups of users.

So based on demography or whatever, and then uh, let the

final user ask. Let's see which are the more.

Beer.

Recommendation.

Brooches.

Sorry, I lost the last, I mean sorry I lost part of your

questions. What is the most?

The most fair OK in the system with respect to a specific

portion of the population, for instance.

Maybe maybe a complex ontology that is able to handle different

recommended systems?

And the.

That in case option I could could deal with the the

experimental results.

Maybe this kind of an option is the the only formal way?

Let's say two to answer this kind of questions.

And I I think that I know that this would need an expansion of

the ontology.

In at least a couple of directions. So the directions of

the users and the demography and the direction of the of the

results.

But I don't know this could be a potential extension in in this

sense.

That would be really appreciated, because I finally

user maybe does not anything about how recommended systems

works and.

And everything. So they just need to know if a specific

recommender system could be fair for for these group of users

say. So essentially you have what you're suggesting is for

instance what I see in order to answer your question is that we

are able to extend the platform on in order to evaluate the

recommender systems farther than adopting the typical metrics

that you have about precision, recall, Ness, measure and you

know these kind of things.

Also an additional thematics about fairness.

You could you could also take from the different frameworks

all the all the measures they have. Obviously I know a lot more and it has more or less 3540 metrics already implemented

there.

But the the the idea that the there isn't in in Elliot and

there is a in in any recommendation framework is like

create these.

These are a shortcut between a final user and the results.

And why there is no? Because if the researcher 1.

To to answer this question, they they need to run experiments,

analyze experiments and everything.

But here since you have this complex ontology.

Maybe you can directly create these shortcut between final

user and.

End results.

Yeah, Umm.

And I think that there is nothing like these, uh, in the

literature so far, but I could be mistaken. But yeah.

I mean no you you know more than much more than at least yeah.

And and obviously if you can reason, this is just an example

the fairness example. But in general if you can reason on the

results.

Usually you can have a lot of semantics of automatic agents that could reason.

All all the recommended systems and results and users. So yeah.

That would be great I think, but.

Very good suggestion. I mean we take note of this. Thank you.

Thank you.

I think related to explain ability and reasoning is also of

interest for for industrial use cases indeed indeed because what

we compete with is an expert user of our software and it's

not like in on Amazon where you do shopping and then you want to

have your recommendations. But we have our recommendations have

to be better or quicker than the typical use of our software who

are already domain experts. So if there is reasoning behind

this, this helps.

To accept assistance technology.

Yes.

OK.

OK. So I mean this took a long longer than expected. So we are

really sorry if we, you know, took all this time from your

side, but we really appreciate your availability and your

comments, suggestions. Thanks a lot. And of course we'll keep

you posted about, I mean our work and we'll share with you what's going on.