 A black letter with purple spirals around it

Description automatically generated

Apollo Tyres

LIMS Solution

An overview of outstanding functionality and open questions

|  |  |  |
| --- | --- | --- |
| Authors | Version | Date |
| CONTACT Software  (Patrick Simon and Fabien Visentin) | 1.0 | 20 Oct 2023 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1. Specification Management 5](#_Toc148694173)

[1.1. BOMs 5](#_Toc148694174)

[1.1.1. Calendared material calculations 5](#_Toc148694175)

[1.1.2. PHR / Weight calculations 5](#_Toc148694176)

[1.1.3. Other BOM calculations 6](#_Toc148694177)

[1.1.4. Alternative BOMs 6](#_Toc148694178)

[1.2. Raw Materials 6](#_Toc148694179)

[1.3. Compounds 6](#_Toc148694180)

[1.4. Semi-finished goods 7](#_Toc148694181)

[1.5. Green tyres and finished goods 7](#_Toc148694182)

[1.6. Workflows 7](#_Toc148694183)

[1.6.1. Release workflow 7](#_Toc148694184)

[1.6.2. Between QRs 7](#_Toc148694185)

[1.7. Roles and Rights 8](#_Toc148694186)

[1.8. Others 9](#_Toc148694187)

[1.8.1. Work centers 9](#_Toc148694188)

[1.8.2. Functional description of BOM Items 9](#_Toc148694189)

[1.8.3. Special changes between frame revisions 9](#_Toc148694190)

[1.8.4. Supplier quality rating 9](#_Toc148694191)

[2. Test data management 10](#_Toc148694192)

[2.1. Processes 10](#_Toc148694193)

[2.1.1. Creating a new Test Request 10](#_Toc148694194)

[2.1.2. Planning/scheduling a Test Request 11](#_Toc148694195)

[2.1.3. Generation of Worksheets and Subsamples 12](#_Toc148694196)

[2.1.4. Performing the tests and entering test results 13](#_Toc148694197)

[2.1.5. Controlplans / quality of production checks 14](#_Toc148694198)

[2.1.6. Creating a simulation Test Request 15](#_Toc148694199)

[2.1.7. Viewing and comparing test results 15](#_Toc148694200)

[2.1.8. Others? 15](#_Toc148694201)

[2.2. Business Objects 16](#_Toc148694202)

[2.2.1. Test Request Types 16](#_Toc148694203)

[2.2.2. Test Requests 16](#_Toc148694204)

[2.2.3. Sample Types 17](#_Toc148694205)

[2.2.4. Samples 18](#_Toc148694206)

[2.2.5. Test Plans 18](#_Toc148694207)

[2.2.6. Parameter Groups 19](#_Toc148694208)

[2.2.7. Parameters 20](#_Toc148694209)

[2.2.8. Test Methods 20](#_Toc148694210)

[2.2.9. Test Method Cells 21](#_Toc148694211)

[2.2.10. Info Cards 22](#_Toc148694212)

[2.2.11. Worksheets 22](#_Toc148694213)

[2.2.12. Equipments 23](#_Toc148694214)

[2.2.13. Others? 23](#_Toc148694215)

[2.3. Others 24](#_Toc148694216)

[2.3.1. Roles and access rights 24](#_Toc148694217)

[2.3.2. Role-based dashboards / default searches 24](#_Toc148694218)

[2.3.3. Notifications 24](#_Toc148694219)

[2.3.4. Communication with test machines 24](#_Toc148694220)

[2.3.5. Connection to specification management 26](#_Toc148694221)

[2.3.6. Others? 26](#_Toc148694222)

[3. Engineering process 27](#_Toc148694223)

[3.1. Product creation / size extension 27](#_Toc148694224)

[3.1.1. QR0: Marketing specification 27](#_Toc148694225)

[3.1.2. QR1: Mould information 28](#_Toc148694226)

[3.1.3. QR2: Specification enhancement 28](#_Toc148694227)

[3.1.4. QR2: Extrudate Design (SPPL) 30](#_Toc148694228)

[3.1.5. QR2: Die creation (SPPL) 31](#_Toc148694229)

[3.1.6. QR2: Profile validation 32](#_Toc148694230)

[3.1.7. QR3: Prototyping 33](#_Toc148694231)

[3.1.8. QR3: Specification release 34](#_Toc148694232)

[3.2. Size extension specific process 35](#_Toc148694233)

[3.3. Design possibilities 35](#_Toc148694234)

[3.3.1. All CAD jobs 35](#_Toc148694235)

[3.3.2. Calculations 36](#_Toc148694236)

[4. Interfaces to other systems 37](#_Toc148694237)

[4.1. General information 37](#_Toc148694238)

[4.1.1. Required data 37](#_Toc148694239)

[4.1.2. Interfaces through SAP PO 37](#_Toc148694240)

[4.1.3. Test machines interfaces possibilities 37](#_Toc148694241)

[4.2. Interface to SAP 38](#_Toc148694242)

[4.2.1. Part number generation 38](#_Toc148694243)

[4.2.2. Synchronization of specifications 38](#_Toc148694244)

[4.2.3. Synchronization of catalogues 39](#_Toc148694245)

[4.2.4. Synchronization of pricing 39](#_Toc148694246)

[4.2.5. Synchronization of projects 39](#_Toc148694247)

[4.3. Interface to WMS 40](#_Toc148694248)

[4.4. Interface to Test Machines / Tools 40](#_Toc148694249)

[4.4.1. Outgoing calls & generation of control files 40](#_Toc148694250)

[4.4.2. Incoming calls 40](#_Toc148694251)

[4.5. Interface to CATIA 41](#_Toc148694252)

[5. Reports 42](#_Toc148694253)

[5.1. General overview of reports 42](#_Toc148694254)

[5.1.1. Required Data 42](#_Toc148694255)

[5.1.2. Report types 42](#_Toc148694256)

[5.1.3. Others 43](#_Toc148694257)

[5.2. Specification management Reports 43](#_Toc148694258)

[5.2.1. PDF Export based on Specification filters 43](#_Toc148694259)

[5.2.2. Full Specification export 43](#_Toc148694260)

[5.2.3. First Product Check Card 43](#_Toc148694261)

[5.2.4. Multi-level Targeted Specification 44](#_Toc148694262)

[5.2.5. Targeted Specification 45](#_Toc148694263)

[5.2.6. Interspec analysis / Specification overview reports 45](#_Toc148694264)

[5.2.7. Tyre Composition & Weight 46](#_Toc148694265)

[5.2.8. Compound and mixing reports 46](#_Toc148694266)

[5.3. Test data management reports 47](#_Toc148694267)

[5.3.1. Test overview report 47](#_Toc148694268)

[5.3.2. Test planning report 47](#_Toc148694269)

[5.3.3. Test requester overview 48](#_Toc148694270)

[5.3.4. Test preparation overview 48](#_Toc148694271)

[5.3.5. Tyre evalution sheet (TES) report 48](#_Toc148694272)

[5.3.6. Conformity of Production 48](#_Toc148694273)

[5.4. Other reports 49](#_Toc148694274)

[5.4.1. Size extension report 49](#_Toc148694275)

[5.4.2. Other reports 49](#_Toc148694276)

[6. Data migration 50](#_Toc148694277)

[6.1. SAP harmonisation 50](#_Toc148694278)

[6.2. Specification management 50](#_Toc148694279)

[6.2.1. Frames 50](#_Toc148694280)

[6.2.2. EU Specifications from Interspec 50](#_Toc148694281)

[6.2.3. APMEA Specifications from Excel / SAP 51](#_Toc148694282)

[6.2.4. Specification documents 51](#_Toc148694283)

[6.2.5. Keywords 51](#_Toc148694284)

[6.2.6. Display formats 52](#_Toc148694285)

[6.2.7. BOM functional description 52](#_Toc148694286)

[6.2.8. Work Centers 52](#_Toc148694287)

[6.3. Test Data Management 53](#_Toc148694288)

[6.3.1. Relevant Unilab objects 53](#_Toc148694289)

[6.3.2. APMEA test management data 54](#_Toc148694290)

[6.3.3. Documents related to test management 54](#_Toc148694291)

# Specification Management

In this section we try to give brief overview of *only the outstanding* topics related to Specification management, so leaving aside anything that was already implemented and deployed on either the Production or the Test environment.

Furthermore, this section deals mainly with the basic functionality around viewing, creating and maintaining Specifications. For the full overview of *all* remaining topics that will influence the Specification management side of the LIMS solution, one should also consider sections 3 (replacement for the engineering tools XPERT, Spectrac & SPPL), 4 (interfaces to other systems, such as SAP), 5.2 (reports for Specification management) and 6.2 (data migration for Specification management).

Any remaining questions with need for additional specification and/or input from Apollo are **highlighted** in bold type.

## BOMs

### Calendared material calculations

Nico and Venkatesh provided a specification for calendered material that explains calculation of the required BOM quantities based on chemical and physical properties of the used compound and the fabric/steel cord RM. A first prototype of functionality was built, using these calculations for calendered materials as an example, which will enable Apollo to set up and maintain these kinds of calculations on their own.

The prototype solution must still be properly demonstrated to and tested by Apollo and will likely need some extension based on the feedback from this. At the moment it can be used to create a new BOM with certain calculated BOM quantities, but it cannot for example edit the specification properties of the calendered material.

### PHR / Weight calculations

The PHR / Kg calculation in BOM was already delivered in the context of compounds and calendered materials.

Additional feature requested: Proper weight calculation for semi-finished goods, green tyres, and finished goods. The calculation should be carried out by traversing the entire BOM down to the level of the raw materials and summing up all individual weights.

**Question: What about situations where the specification properties of higher-level BOM Items (compounds, SFGs, …) also contain a weight value, which might slightly differ from the weight which is dynamically calculated from the raw materials? Should these specification property values be taken into account? Should they be overwritten at some point?**

### Other BOM calculations

The prototype BOM calculation for calendered materials was developed in a generic manner, so that the same approach will also be usable for calculations for other types of frames.

However, if any upcoming calculations would require more advanced methods then the ones described for calendered materials, this has not yet been shared and will likely lead to additional implementation efforts.

### Alternative BOMs

Alternative BOMs / bag management was implemented with a basic machine management.

**It was not yet defined how alternate recipes should behave after revisioning the original PHR recipes.**

**Furthermore, it is unclear if there should be any automation (and if so, to which degree) for how changes in the PHR recipe should be reflected in all of the recipe’s alternative BOMs.**

Venkatesh stressed that some kind of automation here would be very useful, but so far, the exact mechanism for how this should work has not been clearly specified and would likely be potentially difficult to implement (since some alternative BOMs can differ significantly from the PHR BOM, it would not always be clear how a change in the PHR BOM should carry over to the alternative BOM).

## Raw Materials

Almost all functionality related to Raw Materials was already developed, deployed to both Test and Production environments, tested and signed off on by Apollo.

There are a limited number of outstanding tasks related to the Active Index view and the Global Approved Supplier List, which were already discussed with Apollo.

**For some of these topics it was agreed to have an additional session for clarifications, which has not yet taken place**.

It was agreed that these issues should be worked on with high priority, but only after the initial Raw Material Go-Live.

**Due to the temporary stoppage of all project activities, these tasks are still outstanding.**

## Compounds

The fundamental functionality to work with compounds frames, specifications and BOMs was already delivered to the test environment.

**As Apollo has not yet finished full tests of all functionalities related to compounds, additional feedback and further required features are expected to lead to additional efforts and must therefore be specified.**

## Semi-finished goods

Excluding features which will come from the Engineering tools, we are not aware of additional features needed on top of the ones already developed for compounds.

## Green tyres and finished goods

Excluding features which will come from the Engineering tools, we are not aware of additional features needed on top of the ones already developed for compounds.

## Workflows

### Release workflow

Release / approval workflow was created for RMs. It was discussed and agreed that the solution for the non-RM workflows and lifecycles would differ from the Interspec implementation, where there are many different statuses with similar names (CURRENT QR2, CURRENT QR3, …), and that the QR-phase (QR1, QR2, …) would be decoupled from the lifecycle status (Draft, Review, Approved, Released, Obsolete). Therefore, it is expected that most approval workflows will look quite similar, with the main difference being different roles for which user groups should have rights to do certain tasks or be informed about certain things.

**If any of the review/approval workflows for non-RMs need additional functionality, it must be specified.**

**Furthermore, we must still agree on how to handle the QR information, a suggestion was to have a mandatory field on the status change dialog (from Draft -> Review), where the user must select the QR phase in addition to the target status.**

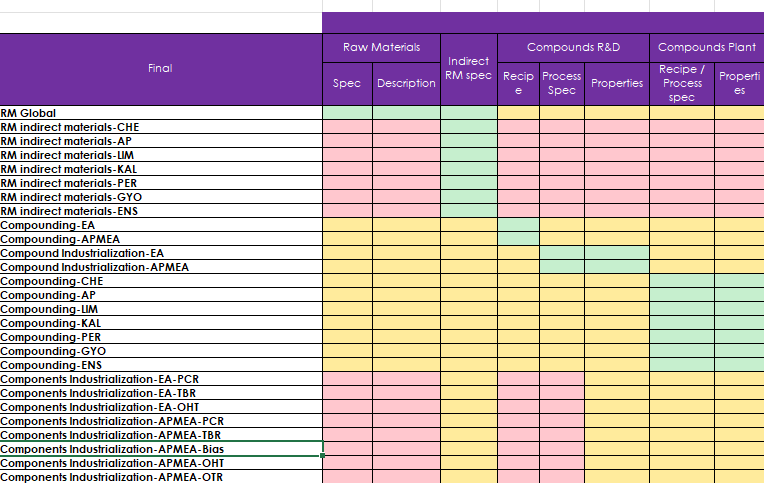
**Finally, for RMs it was implemented that the Approval workflow automatically results in the generation of the Specification Change Note (SCN). If any other similar documents must be automatically generated for Compounds, SFGs and above, it is not currently known. (See also the Reporting section 5.1.2)**

### Between QRs

**The workflows to pass from QR to QR needs to be specified (see also questions in section 3.1).**

## Roles and Rights

An overview of 225 roles and which parts of the Specification management solution they should have access to was shared by Apollo in 03/2023:



Several solution approaches were estimated and it was agreed to go with an intermediate approach where access would be limited by Specification category (RM, Compounds, SFGs, …), plant/region (Chennai, Enschede, APMEA, …) and product category (PCR, TBR, …), but that there wouldn’t be additional access restrictions which allowed only partial access to certain Sections/Subsections within the full Specification, such as having different access on the Properties and Processing sections (hiding certain irrelevant information for some user groups can be achieved using the already developed Specification filter functionality, but in its current form this will not guarantee 100% data confidentiality).

**Apollo should review the shared document and update it, if necessary. Additionally, it may be necessary to define roles which are responsible for certain workflow tasks (similar to the already established role “Approval Raw Materials”).**

## Others

### Work centers

A rudimentary machine management was implemented so far thanks to the information that was available in the specifications (so for every frame which we migrated, we manually created Work Center objects corresponding to the subsections in the Processing XYZ sections. Ideally it would be better for Apollo to provide a full overview of all Work Centers for all Apollo plants (see also the Migration section 6.2.8).

**If there is need for additional functionality coupled to the Work Centers (example: limiting that certain Work Centers may only be used in connection with certain Frames), it must be specified.**

### Functional description of BOM Items

We know that BOM items can be assigned a functional description, but we do not have a good overview which other functionality actually depends on these functional descriptions. For example, we know that some of the Athena reports described in section 5.2 display the function of BOM items and group information by that function, but we don’t know where else it may be important.

**We need to have an overview over all functional descriptions (also see Migration section 6.2.7), as well as explanations for how they are used. For example, should there be functionality where a Greentyre specification cannot be released if it doesn’t have entries for Tread/Sidewalls/… in the BOM, or is this checked manually by the responsible people? Should there be limitations where certain functional descriptions can/must be used a certain number of times for BOMs in certain frames?**

**If there is any other important metadata that must be assigned on the BOM item level, like the function, it must also be specified.**

### Special changes between frame revisions

The basic functionality to update Specifications made from older frame revisions to the newest frame revision has been developed and works. However, it was mentioned that in rare cases there can be special change operations from one frame revision to the next, which would not be automatically recognized, but may still need to be carried out to ensure data is not lost (for example if one Property is moving to an entirely different Property Group or Section).

**CONTACT needs a description of all such special scenarios that could occur, so that a solution approach and effort estimation can be made.**

### Supplier quality rating

**This hasn’t been discussed anymore, specification is needed.**

# Test data management

In this section we try to outline – roughly separated by typical processes (or user interactions), business objects, and other aspects – our understanding of Apollo’s current way of working with Unilab and the general requirements for a test management solution, based on the multiple user story sessions and the discussions during the joint workshops. Any remaining questions are **highlighted** in bold type.

Further topics related to the test data management side of the LIMS solution can be found in sections 4 (interfaces to external systems like the WMS tool), 5.3 (reports about test data management) and 6.3 (data migration for test data management).

## Processes

### Creating a new Test Request

High-level requirement: Test requesters should be able to request a set of tests to be carried out on one or multiple test Samples, which are to be validated against a range of expected values. The system should aid them as much as possible by pre-filtering and suggesting only relevant types of Samples and Test Methods.

Unilab process: The general procedure for creating a new Test Request is very similar across all departments (Lab, Indoor, Outdoor): a test requester created a new Test Request based on the template of one of the available Request Types. Depending on the Request Type, the requester can (or must) provide different kinds of additional metadata, such as assigning the Request to a certain project, asking for the test to be carried out by a certain date, or providing additional test operation instructions to the testing personnel. Project assignment is usually mandatory since this decides where the budget for a test is coming from. Projects are synced with SAP. As part of the Test Request creation, the requester also must assign one or more Samples, instantiated from several available Sample Type templates (which Sample Types are available depends on the chosen Request Type), each of which is explicitly connected to the Part No. of a Specification. One Test Request can contain Samples with different Part Nos. and with different Sample Types. Additional metadata may be added for these samples (such as whether to keep or scrap tyre samples after testing), but this is not always mandatory. After the samples are created, a Test Plan must be assigned to each sample (the same test plan can be assigned to multiple samples at once). Which tests are available as part of the full Test Plan depends on multiple factors, but usually the Sample Type defines which Parameter Groups and Parameters (aka AV Test Methods) are available, and in some cases additional logic selects or hides certain alternative Test Methods based on input values from the connected Specification of the Sample. For many Sample Types the full Test Plan will have many possible Parameters, from which only a small selection is manually chosen to be included in any given Test Request. After all Samples have their proper Test Plan assigned, the status of the Test Request is changed to “Submit” and it will be processed by the users responsible for test planning in the respective labs.

### Planning/scheduling a Test Request

High-level requirement: Test planners should be able to view, validate and schedule (or reject) submitted Test Requests which are relevant for their area of responsibility. All involved parties should be informed about relevant activities and outstanding tasks.

Unilab process: The users responsible for planning and scheduling Tests will typically search for newly submitted Test Requests within their area of responsibility and open them for validation. Then they will check the Info Card of the Request itself (to see if all required metadata is there and looks correct), followed by the Samples (to see for example if the selected Part Nos. fit the chosen Request Type and Sample Types) and the selected Parameters from the Test Plan (sometimes requesters may ask for certain tests or combinations of tests that do not make sense for the given Sample). Depending on these validations, the Test Request is either rejected or accepted. When it is rejected, the test requester should be informed about any inconsistencies, so that the Request can be amended and re-submitted. When the Request is accepted, its status is either changed to “Planned” (a slight misnomer, as it means that the Request is valid, but not yet ready to be performed) or “Available” (meaning that the Request is valid and ready to be performed because it was verified that the necessary testing materials are present in the lab). Some departments may skip the “Planned” status and always directly set a verified Test Request to “Available”. In some cases (and once the Indian plants are integrated probably in many more cases) certain Test Methods can be carried out in multiple locations, one of which must be chosen during this stage, preferably with some (overridable) automation to avoid unneccessary user interactions.

Questions:

* **In the user story session, it sounded like these users currently regularly have to use an Athena report to get an overview of open Test Request which they then manually search for by code in Unilab to work on. Is this accurate, and if so, should it not be improved so that they have a useful dashboard (or something similar) where they see these requests, and to have a kind of automated notification that is sent to the test planners whenever a new Test Request for their lab has been submitted?**
* **Are the same users which check the validity of submitted Test Requests responsible for assigning the Tests to certain locations and weeks? Or are these separate steps (especially since one Test Requests will typically contain many Test Methods which may be carried out at different dates in different places)?**

### Generation of Worksheets and Subsamples

High-level requirement: Test requests comprise many different tests in a number of locations and concerning many Samples or sets of Samples. The system should aid its users by making it possible to view and schedule separate groups of tests coming from these larger Test Requests and by making sure that as many physical Samples are created and stocked as are needed.

Unilab process: Worksheets were demonstrated during the user story session for outdoor tests. After a Test Request has been accepted, Worksheets are generated for the various Tests belong to the Request. A Worksheet corresponds to the task of carrying out a certain kind of Test in a specific location on one or multiple Samples contained within the Test Request. For outdoor testing there is additional logic connected to these Worksheets, because outdoor tests use Samples which represent a set of tyres (either a “normal set” with four tyres of the same type, i.e. the same Specification and Part No., or a “mixed set” with two different Part Nos. for the front and the rear tyres, in very rare cases each of the four tyres might be selected individually). When a Worksheet for carrying out a specific Test Method on one (or more) of these Sample sets is set to the “Planned” status, an additional preparation Worksheet is created and a number of Subsamples are created and added to the Test Request and that Worksheet, typically with a small pre-defined Test Plan that includes a Test Method for mounting the tyres in preparation for the outdoor test. Each subsample represents a different physical tyre that has to be produced/ordered, and the total number of Subsamples depends on what kind of Sample sets and what kind of Test Methods were requested. This way, a single Sample set in the initial Test Request may lead to dozens of physical tyres that have to be stocked (because tests could be carried out in different locations or may deteriorate the tyre quality so that they should not be reused for other tests within the same Test Request). When the newly created preparation Worksheets are set to the “Available” status, communication to the external WMS tool is triggered, so that all Subsamples can be stocked and prepared. Once the requested tyres are available in the warehouse, the preparation Test Methods are filled out with the relevant information and marked as completed, so that all information is available in the Worksheets and the actual tests may be carried out.

Questions:

* **Is the creation of Worksheets coupled to the status change from “Submit” to “Planned” (or “Available”)? Or are they already generated once a Test Request has been submitted?**
* **Are Worksheets also used by other departments (Lab, Indoor) to plan (or view) sets of tests for a given Test Request, or is this strictly limited to outdoor testing? If yes, how do users typically work with worksheets and is there any additional functionality or logic that was not mentioned above?**
* **Are Sample sets and Subsampling used by other departments (Lab, Indoor) to plan (or view) sets of tests for a given Test Request, or is this strictly limited to outdoor testing?**
* **In the user story session it seemed like information from the warehouse about tyre codes, weights, etc. is manually entered in the Worksheets in Unilab? Is this true, or does all of that also happen automatically through the Unilink interface to the external WMS tool?**

### Performing the tests and entering test results

High-level requirement: Testers must have a way of knowing which tests to carry out when, where and on which Sample, and all relevant test results must be stored in the correct place, with as little room for human error as possible. Follow-up actions based on successful/unsuccessful test results must be clearly defined and initiated automatically.

Unilab process: Once a Test Request has been fully processed and scheduled, the various Test Methods are ready to be carried out. Usually not all Test Methods are available to be carried out at the same time, instead there may for example be some logic that dictates that each Test Method for a single Parameter can only be started once its predecessor has been set to “Completed”. The tester responsible for carrying out a certain Test Method has to get an overview over all tests that should be carried out on a certain testing Equipment (or a test track, etc.) in the current period of time. How exactly this happens in each case is unknown, but in some cases, they will open Unilab on a given test Equipment and have an overview of open Tests right there, in other cases the Equipment might not be directly connected and the overview has to be obtained on a different Equipment. In some cases, Unilab can directly generate control files for test machines and store them on the machine’s network drive. The tester must also make sure that the correct sample will be used to the test. After carrying out the test, the results are recorded, processed, and stored in the correct place in Unilab (i.e., assigned to the correct Test Request, Sample and Test Method). In some cases, this might require manual interaction with Unilab, i.e., entering results and performing status changes by hand, but in the majority of cases the raw result data is processed by external tools, the relevant result values are extracted and transformed into a Unilink-format text file, which is then placed into an exchange folder and automatically interpreted by Unilab. Some of these steps may be handled by the test Equipments themselves, others are handled by the external LIMS client system. Entered results for Parameters may be validated against the target and upper/lower limit values coming from the Sample’s Specification, resulting in further actions (the range of possible scenarios is unclear). For manual data entry, there must be an option of “blind testing”, where the tester responsible for entering the results does not have access to the target value or the validation results.

Questions:

* **Will testers always interact with Unilab in some way in order to see what tests must be carried out, or are there fully automated setups where they will only deal with the external control software of their testing Equipment, and all open tests will be shown and handled there?**
* **What exactly will they typically search for? All Samples on which a test must be carried out on their testing Equipment? Or all related Worksheets? Something else?**
* **How common is it that Unilab creates and sends control files to the testing Equipment? Is there different logic implemented on Unilab-side for each such scenario, or will it generally send full information about Test Request, Sample and Test Method and leave it to the external software to extract the relevant properties for execution of the Tests? Alternatively, how common is it for testers to open Test Methods in Unilab manually in order to look at the requested test settings there? (Probably more common for outdoor tests, although here it was also mentioned that certain Athena reports are used to generate Excel files which are used by MATLAB tools?)**
* **How can testers make sure that the results of their tests are always connected to the correct Sample? Is this a manual process where they have to enter/confirm the code of the sample on the testing Equipment every time? Is there some process for barcode generation in use which we have not seen but which must be implemented in the new LIMS system?**
* **What are the scenarios for test result validation? It has been mentioned that certain out-of-spec measurements may result in re-measurements, but at which point is this decided, and by whom? Is there a need for integration within a workflow, where failed tests create a task for someone to give their ok/not-ok for retesting, potentially leading to the planning of additional tests? Are follow-up Test Methods still carried out if a preceding measurement was out of spec?**

### Controlplans / quality of production checks

High-level requirement: To ensure the continued quality of all materials which are used in the tyre manufacturing process, regular automated quality checks must be scheduled and performed without the need to rely on manual Test Request creation.

Unilab process: While there was no proper session on this topic, it was brought up multiple times that in certain scenarios tests are automatically scheduled by the MES tool based on the controlplan in the Specifications of various materials, compounds, etc. In these scenarios, there is no creation of a Test Request by a user, instead Samples with specific Test Plans are automatically created in Unilab. For the testers it should not make a difference whether a certain requested test is coming from one of these automated quality checks or from a manually generated Test Request. It was brought up that this setup currently only works well for lab and indoor tyre tests, but not for outdoor tests (likely because the subsampling mechanism does not work correctly without a test request), but that ideally it should work for outdoor tests as well.

Questions:

* **Are the rules for how these automated tests are created identical for all kinds of Specifications, i.e., does it work the same way for compounds and for finished tyres?**
* **Is it a hard requirement that these automated tests only generate Samples, but not Test Requests? In the new system, it might be easier to always start from a Test Request, which in these cases then would automatically be approved and scheduled without the need for user interaction.**
* **What are the exact circumstances of the current issues when it comes to outdoor testing, are there any additional difficulties to which a solution must be found?**
* **Are there any special processes in place for validating the results and dealing with unsuccessful tests for these kind of tests, or are they handled exactly like the results for manually requested tests?**

### Creating a simulation Test Request

High-level requirement: Requesting simulation runs instead of physically performed tests, and storing their results, should fundamentally behave in the exact same way. All processes should be able to be automated and externally controlled as much as possible.

Unilab process: The handling of simulation Test Requests is currently done by the external Desktop Tool (DTT), which communicates with Unilab in the Unilink format. There are certain Request Types and Sample Types in Unilab which correspond to simulation tests, but the basic functionality is the same (with the main difference being that a Sample for these Test Requests does not correspond to a physical tyre that has to be stocked somewhere, but instead represents a distinct set of input/output data for one or multiple simulation test runs). Unilab is responsible for creating control files for the external simulation tools (there are multiple external tools working in sequence, which also query certain parameters which are stored in Interspec), at the end of the external simulation runs the processed results (but not the large raw result files) are stored back into Unilab automatically. If any errors occur during the simulation, the relevant people are automatically informed via e-mail so that necessary steps may be taken.

Questions:

* **Do users who request simulations and view results currently interact with Unilab at all, or are all steps handled from within the external DTT (and maybe Athena for certain reports)?**
* **How are the control files for the simulation runs generated from Unilab? Is this identical to the process where control files for physical test Equipment can be generated? Are there different settings for different Test Methods, or does Unilab always transmit the same kind of information and it is up to the external tools to extract the relevant properties?**
* **Is the error handling and notification of users currently handled completely by the external tool? Should this part be brough back into the future LIMS system, so that such notifications and follow-up tasks become part of the regular workflows as used elsewhere in the system? Or do users expect to continue handling everything from within the DTT without any need of accessing the LIMS system directly?**

### Viewing and comparing test results

*To be found in section 5.3, “Test data management reports”*

### Others?

Questions:

* **Are there any other typical processes in which users currently interact with the LIMS system, but which are not covered by the preceding subsections (likely because they were not brought up in the user story sessions or the workshops)?**
* **Are there any other processes related to test management which are not currently handled by Unilab (or not in any meaningful way), but which should be covered by the new LIMS solution?**

## Business Objects

This section mainly describes the different kind of business objects and their functionality in the current Unilab setup, with some questions about unclear elements. The data structures in the new LIMS solution will likely not be a 1:1 adaptation, but a full proposal based on the CONTACT standard can only be given once a full common understanding on the requirements and the current way of working is reached.

### Test Request Types

These are templates for different types of Test Requests that can be made by users. They are necessary to avoid overwhelming users with too many input fields and possibilities for test methods. Each Request Type defines, using connected Info Cards (see below in subsection 2.2.10), which metadata can and must be filled out by users, and which Sample Types will be available to be chosen. For some Request Types additional logic is set up that may influence the behavior of the assignment of Samples and Test Plans. For example, for compound test requests the users will be able to choose between multiple mixers while assigning Samples, leading to the automatic generation of an initial Test Method for mixing on each Sample’s Test Plan.

Questions:

* **Where is the logic maintained that defines whether a Request Type has additional behavior, such as outlined in the compound example above?**
* **How many Request Types currently have special behavior such as this? In the user story session, only this example was explicitly named.**
* **Can the Request Type already determine that certain Parameter Groups and Parameters are added to the Test Plan for all Samples, or is this only maintained on Sample Type level?**
* **Is there any additional business logic or important metadata that is configured on the Request Type level?**
* **Do Request Types have their own object lifecycle? Can they be revisioned? If yes, is there any special behavior connected to that, or is it mainly used to keep a record of previous versions?**

### Test Requests

Test Requests are manually instantiated from the Request Type template and represent a distinct set of connected tests which are to be carried out on one or multiple Samples. They are identified by a unique code, and they have their own object lifecycle which is coupled to the various test management processes outlined in the previous section. Typical metadata for any test request includes the expected date by which the request should be carried out, the project to which the test request belongs, a week in which it is scheduled, a priority or a specific testing site (although these things may also be maintained on the level of the individual Samples and Test Methods).

Questions:

* **The recordings from the user story sessions show that Test Requests can be assigned to a “Work Order”, but it was never explicitly mentioned what that is or whether it is used. Is this relevant for the new LIMS solution?**
* **How common is it to enter locations and weeks on the Test Request level, if different Test Methods may actually be performed at different dates and in different locations? Is there (or should there be) any business logic connected to this metadata, so that for example requesting a test at R&D Europe will pre-filter the available test Equipments for the various Test Methods to only show machines in the European labs?**
* **It was not explicitly brought up, and probably does not make sense (unless you want to keep a copy of rejected or failed Requests), but can Test Requests be revisioned?**

### Sample Types

Sample Types are templates for the kinds of Samples that may be attached to Test Requests based upon a certain Request Type, and also for the kinds of tests which might be performed on these types of Samples. For each Request Type, only a limited subset of Sample Types is available. It is not completely clear, but likely, that Info Cards can be assigned to Sample Types, similar to Request Types, to define additional metadata that users can or should fill out when instantiating a Sample of the chosen type. Sample Types also define which Parameter Groups (also single Parameters?) will be available in the Samples’ full Test Plan, and possibly additional logic for only assigning certain Parameter Groups based on input properties of the chosen Sample Part Nos. or the Test Request metadata. Sample Types likely also define which Part No. prefix will be pre-filled in order to help users choose appropriate Specifications for the Samples.

In addition to the Sample Types which are defined for certain Request Types, there is also a distinct Sample Type for each Specification that exists in Interspec, although the concrete use cases for that were not discussed.

Questions:

* **Can Info Cards be assigned to Sample Types?**
* **What are the possibilities of configuring which Parameter Groups, Parameters or Test Methods appear in the Test Plan of a given Sample Type?**
* **Is there any additional business logic or important metadata that is configured on the Sample Type level?**
* **Do Sample Types have their own object lifecycle? Can they be revisioned? If yes, is there any special behavior connected to that, or is it mainly used to keep a record of previous versions?**
* **What is the functionality and necessity behind the Sample Types for individual Specifications? Are these mainly used as a workaround solution so that the information about which properties (and which property values) are assigned to a Specification in Interspec is also available in Unilab? Is it correct to assume that this will no longer be required in the new LIMS solution?**

### Samples

Samples are manually instantiated from a Sample Type template. Depending on the testing department within Apollo, a Sample can represent slightly different things. In lab testing or indoor tyre testing, one Sample typically corresponds to one physical object (e.g., a piece of a rubber Compound or a full tyre) and it is directly associated with a single Interspec Part No. and the corresponding Specification. In simulation testing, the Sample also has a single Part No., but does not correspond to a physical object (since it is only used for virtual simulation runs). In outdoor testing, one Sample can stand for a full set of tyres, and may have multiple Part Nos. attached, if different tyres are to be used on the front and rear axles. In the process described in the previous section, sub-samples, which represent the actual physical tyres which are to be used in the various Test Methods, are then automatically generated based on the chosen Sample sets and Test Methods. To make the selection of the desired Part No. for a Sample more convenient, the range of all available Part Nos. can be prefiltered, for example by the Part No. prefix defined in the Sample Type, the user ID of the user who created the Interspec Specification, or the date range in which the Specification was created.

Samples are identified by a unique code, which is typically made up of the Test Request code and a counter (and potentially a Group ID for automatically generated Subsamples). Samples have their own object lifecycle which is coupled to the various test management processes outlined in the previous section. The status of Samples can automatically be affected by status changes of the parent Test Request or completion of the assigned Test Plan. Additional metadata for Samples can for example determine whether a tyre sample should be scrapped or kept for re-testing after the Test Plan has been carried out.

Questions:

* **In Unilab Samples are first created without a specific Part No., which is then entered in a second step. Is it acceptable to force users to select Part Nos. directly during the Sample creation, or are there scenarios where it must be possible to leave the Part No. blank or to assign a Test Plan before a Part No. is chosen?**
* **Are there specific procedures for what should happen if a Sample is damaged/destroyed/lost before all planned Test Methods for it were performed? Can additional Samples be added to a Test Request on the fly?**

### Test Plans

First of all, it is not quite clear whether a Test Plan is actually a distinct business object in Unilab, or whether it is just a compiled view of all possible Parameter Groups, Parameters and Test Methods which can be assigned to a given Sample based on its Sample Type (and its Request Type?). Generally, a Test Plan can be assigned to either one or multiple Samples, which opens a dialog with the overcomplete list of all available Test Methods. These can be selected either in bulk, by adding/removing a certain Parameter (Group), or individually. During the selection in can be specified whether certain parts of the Test Plan should be measured multiple times (Parameters or Test Methods), the result of this was however not discussed in full detail. Once a (reduced) Test Plan has been assigned and the Test Request has been submitted, it is possible to view only the actual (and not the overcomplete) Test Plan for each Sample. From this view the users can further navigate to the test results of certain Test Methods.

Questions:

* **Is there actually a Test Plan object on database level, or is it only a view of the available (or chosen) Parameter Groups which are assigned to a certain Sample Type (or Sample)?**
* **Where do the displayed Parameter Groups in the overcomplete Test Plan come from? Are they all configured on the Sample Type level, or can certain Parameter Groups also be assigned based on the chosen Request Type?**
* **At which level does additional logic come in, for example where certain alternative Test Methods are assigned based on property values from the Specifications of the chosen Sample? Does this logic already pre-filter the overcomplete Test Plan which is displayed during the Test Request creation, or does this only come into play when the actual tests are being carried out? Can test results from one Test Method influence which of multiple possible follow-up Test Methods is carried out?**
* **What is the result of specifying that certain Test Methods or Parameters should be measured multiple times? Should this create copies of Test Methods or tasks? How does the system deal with the results of multiple measurements?**
* **Since Test Plans can be assigned to multiple Samples at once: is it possible to assign the same Test Plan to Sample from different Sample Types? Probably not, right?**

### Parameter Groups

Parameter Groups represent the upper hierarchical level of a Test Plan. They typically have an identifying name and a description, and they group together one or multiple Parameters. Sometimes they are also referred to as Parameter Profiles. Parameter Groups can be configured to belong to certain Sample Types, so that they automatically become part of the Test Plan for any Sample instantiated from this Sample Type. Parameter Groups do not have a test result, but they have their own object lifecycle. Based on status changes of the connected Sample, the status of the Parameter Group may be changed automatically. Similarly, after the Parameters inside of the Group are tested and validated, the status of the entire group will be set automatically.

Questions:

* **Can Parameter Groups also be assigned to Request Types, or only to Sample Types?**
* **Is there any additional business logic or important metadata that is configured on the Parameter Group level, such as functionality for determining that only certain Parameters in a Parameter Group are shown in a Test Plan depending on other input factors?**
* **Can Parameter Groups be revisioned? Is this used?**

### Parameters

Parameters represent the middle hierarchical level of a Test Plan. Each Parameter belongs to a Parameter Group and has one or more underlying Test Methods. Parameters are sometimes also referred to as “AV Test Methods”. They have an identifying name and a description. The identifying name is often (but not always) a Test Method code (like TP006AA) which is also used to link it to a corresponding property in Interspec Specifications. Each Parameter has a result, which can come with a unit of measurement, and it may have a target value and upper/lower spec. or warning limits (likely coming from the Specifications of the selected Sample). Parameters have their own object lifecycle, with status changes behaving similar to the ones described above for Parameter Groups. Additional metadata on the Parameter level can predefine certain things, such as the Sub Program ID (used in the subsampling mechanism of outdoor tests), whether tests should be carried out in certain default test locations, whether tests should be carried out on all four or only some tyres of a car. In the configuration of a Parameter’s Test Methods, it can be defined whether certain Test Methods should only become available after completion of a predecessor Method, or if they should or should not be assigned to a Test Plan based on the Sample Specification properties.

Questions:

* **Is there any additional business logic or important metadata that is configured on the Parameter level?**
* **Can Parameters be revisioned? Is this used?**
* **In the workshop it was mentioned that certain Test Methods sometimes can sometimes create new Parameters that were not part of the initially assigned Test Plan. How exactly does this work, and is it correct that this is mainly used as a workaround so that a single Test Method can be used to produce multiple results which are then validated against multiple Specification target values?**
* **How does it work if the Test Plan requested multiple measurements of a single Parameter?**

### Test Methods

Test Methods represent the lowest hierarchical level of a Test Plan. Each Test Method belongs to a certain Parameter. They have an identifying code and a description. The code of the Test Method is likely not always found in the Specification of the Samples (but may be in some cases). Test Methods have a single result, which can have a unit of measurement. Often the result may only be the string value “Completed”, if the Test Method does not represent a measurement but for example a preparation step that is necessary to carry out the next Test Method in the test sequence of a given Parameter. The result of one of the Test Methods for a Parameter is also used as the test result of the entire Parameter, which is validated against the Specification target values and limits (see above). A Test Method can be assigned to a specific Lab, in case that it can be carried out in multiple locations.

Each Test Method has a dialog consisting of various Test Method Cells, so while it only has a single “official” result, it may in fact store many result values, which may be used by the reporting tool Athena or for further calculations or business logic of other Test Methods. Test Methods have their own object lifecycle. They can be available to be carried out based on logic defined on the Parameter level (typically depending on the completion of a predecessor Method). They can only be completed when all mandatory Test Method Cells are filled out. Some Test Methods are not marked as completed due to manual user interaction (or through Unilink commands), but automatically after a certain time period has passed (for example to represent steps where a material must be conditioned for a certain number of hours). Certain Test Methods can be used to generate control files for the connected test Equipment.

Questions:

* **Where is it defined which Test Method’s result will serve as the result of the entire Parameter?**
* **What are the scenarios in which it becomes necessary to choose a lab for a Test Method? Is this pre-filled automatically based on certain pre-conditions?**
* **How does the generation of control files for test Equipment work? Is this defined on the Test Method level, or is there a generic approach where, for example, the values of all Test Method Cells are written into a file and stored in a location which is defined on the Equipment level? Or is the control file generation always configured outside of Unilab and only done through external tools (Athena, DTT, LIMS Client, …)?**
* **Can Test Methods be revisioned? Is there any special functionality connected to that, or is it mainly used to keep a record of previous configurations?**
* **How does it work if the Test Plan requested multiple measurements of a Test Method?**

### Test Method Cells

Test Method Cells represent fields on the detail dialog of a Test Method. They can be freely arranged in the configuration of the Test Method in terms of location on the mask and size of the Cell. A Test Method Cell can be freely editable, read-only, or even hidden. A Cell can be marked as mandatory. A Test Method Cell has a unique identifying name, by which it can be referenced in other places (for example during calculations). The displayed input value of the Cell can come from a variety of sources: it may come as direct keyboard input from the user (or from the Unilink interface), it may come from a certain column of a certain Property of the Specification which belongs to the Test Method’s parent Sample, it may possibly come from any Test Method Cell belonging to another Test Method within the same Test Plan (this is not completely clear), it may be a calculated value based either on a simple one-line calculation (using other Method Cells and simple mathematical functions) or a more complex test script written in Visual Basic, it may be the result of an SQL Query on the underlying database. Method Cells can have different data types and allow for example String or Float values. It’s not quite clear from the previous sessions, but likely, that Method Cells can also have a limited set of possible values defined by dropdown catalogs. Cells can also be two-dimensional tables, where each cell inside of the table can be addressed by its x- and y-coordinates.

Questions:

* **Are there other types of Cells? Checkboxes? Date fields? Links to specific objects?**
* **Are there differentiations between integer and float Cells?**
* **Are there Cells which are prefilled with one value, for example based on an automated calculation, but which may be overwritten by manual user input?**
* **Are there any other possible input sources (either for Cells directly, or for use in the VBA scripts that calculate Cell values) in addition to the ones named above?**
* **Can the input of Method Cells actually depend on Cells in other Test Methods of the same Test Plan? What if these Methods were not assigned? What if there are multiple measurements for these methods?**
* **Where is it defined which Test Method Cell will provide the result of the Test Method?**
* **Is there other important Test Method Cell metadata which was not mentioned above?**

### Info Cards

Info Cards are dialogs (or sections of a combined dialog) consisting of a number of cells in which additional metadata for a business object can be entered. The cells can be freely arranged, similar to Test Method Cells, and they can have different types, writabilities, mandatoriness, etc. Unlike for Test Method Cells, there is probably less automated input/output logic connected to the cells on an Info Card. An Info Card can be assigned to Request Types, and probably also to Sample Types. Multiple Info Cards together can make up the full metadata dialog for these objects. Info Cards can be reused for multiple similar objects where it makes sense.

Questions:

* **Do cells on Info Cards and Test Method Cells actually have the same level of configuration options, or are Info Card cells much more restricted in terms of how data can be filled automatically?**
* **Is there any special business logic that has to be maintained on the Info Card level, or does it mostly involve defining which cells are mandatory?**
* **Is there such a thing as a mandatory Info Card which is shared between all Request Types or Sample Types?**
* **Can Info Cards be attached to other objects?**
* **Do Info Cards have an object lifecycle? If yes, is there any special logic to it or is it relatively standard (Cards must be set to “Released” in order to be used)?**
* **Can Info Cards be revisioned? Is there anything special to look out for?**

### Worksheets

The functionality behind Worksheets was not discussed in much detail during the user story sessions and workshops. They represent a testing task, or a group of testing tasks, all with the same Test Method (or just the same Parameter?) to be carried out on one or multiple Samples. Worksheets are automatically generated based on a Test Request, its Samples and their Test Plans, and have a unique ID which is based on the Request ID and a counter. The Worksheet also notes the Sub Program ID of its linked Parameter. For outdoor tests, the chosen test vehicle and location can be assigned on the Worksheet level. Worksheets can be scheduled for certain weeks and with a certain priority. They have a lifecycle. In outdoor testing there is additional logic connected to the subsampling process (see above in subsection 2.1.3), where status changes on one Worksheet can lead to the creation of new Worksheets or of additional Samples.

Questions:

* **Are there different types of Worksheets with different properties, fields, functions?**
* **Are Worksheets ever manually created, or are they always an automated result based on the Test Request?**
* **Are there additional important interactions, business logic or metadata which were not mentioned so far?**
* **Also see questions above in the “Generation of Worksheets and Subsamples” subsection 2.1.3**

### Equipments

An Equipment represents a certain machine, group of machines or maybe even location where a certain type of test can be carried out. An Equipment has a unique name and a description, as well as additional (optional) metadata about serial numbers, locations, installation dates or operational instructions. It is possible to specify a network drive location on the Equipment level, which will be used as the storage location for any control files generated from a Test Method for this Equipment. It is possible to create additional key-value pairs (aka constants) which describe properties of the Equipment which may be relevant in the calculation of values for certain Test Method Cells. Equipments have their own object lifecycle.

Questions:

* **Is the network drive location really used for control file output, or to look for test result input?**
* **Is it maintained on the Equipment level for which kinds of Test Methods an Equipment may be used?**
* **Can one Equipment be used for more than one Test Method?**
* **Is there anything special about the Equipment’s object lifecycle, or is it mainly used so that only currently active Equipments can be used in requesting and planning of tests?**
* **Are there other important functionalities connected to Equipment management that were not mentioned?**

### Others?

Questions:

* **Are there other fundamental business objects that play a role in Unilab processes which were not mentioned at all so far?**

## Others

### Roles and access rights

High-level requirement: It should be possible to clearly define, based on certain role templates, which user will have access to view and interact with exactly which data related to test management.

Unilab functionality: It is possible to define certain user groups which have the rights to perform certain operations, such as creating requests, changing the underlying configuration, viewing and entering results, etc.

Users are assigned to one or multiple User Groups and typically inherit all access rights from these groups.

### Role-based dashboards / default searches

High-level requirement: Depending on the roles inhabited by a user, they should have access to certain pre-defined dashboards and search favorites, in order to make their daily user experience and the onboarding process as convenient as possible.

Unilab functionality: Users belonging to multiple user groups can choose their “active” role, which changes the number of pre-defined dashboards and searches in the interface. On the user group level, it can also be pre-configured that certain searches and result lists will hide or display certain columns which are most relevant for users from this group.

Questions:

* **Are there additional settings, preferences, etc. which affect the look (or available functionality) based on the currently active user group?**

### Notifications

High-level requirement: Users should be kept informed about important developments which are relevant to their interests, for example when a submitted Test Request has been completed.

Unilab functionality: This was noted as a deficiency of the current system, where users are not automatically informed, and it is their own responsibility to get an overview of their own Requests using Athena reports or Unilab dashboards. It was also noted that it would be good to have more convenient ways of seeing all (open) Test Requests belonging to a certain project.

### Communication with test machines

High-level requirement: It must be possible to have an interface to all (connectable) test machines, where important test settings can be exported from the system to the machine, and the obtained test results can be imported back into the system with as much automation as possible.

Unilab process: Getting information out of the system to be used as control files for connected testing Equipment can likely be done in multiple ways, and is not completely understood (see also above). There are ways of defining on the test Equipment level where certain control files have to be stored. In addition, there are also certain reports in the external tool Athena, which are manually generated and whose results (either ASCII or Excel files) can be used as input for testing Equipment or additional planning tools such as the MATLAB tool ANITA. How much of this is currently automated and different from Test Method to Test Method is not fully understood.

Getting information back into the system is clearer, as this is always done using the text-based Unilink format. External test machines, or middleware tools on the machines or the external LIMS Client can parse the “raw” test result files and extract the relevant values to be written back into Unilab using this format. These Unilink files are stored in an exchange folder which is regularly polled for new files, which are then processed. A Unilink file can contain any number of instructions for controlling Unilab in place of manual user interactions. On the most simple level, it can identify a certain combination of Test Request, Sample and Test Method by its automatically generated Import ID, and then give instructions for which values to enter in which Test Method Cells. In addition, it can include instructions to perform status changes. In more complicated scenarios a Unilink file can include instructions for the creation of entire Test Methods or Parameters, which may for example be required in tests where the actual number of test measurements could not yet be known at the time of the Test Request creation.

For tests on tyres, communication to external systems already happens on the Sample level, where requested samples must be ordered and stocked in the warehouse. This is handled by the WMS client, which also uses the Unilink format to communicate with Unilab.

Not currently part of Unilab, but asked for, is a more convenient way of providing access to the raw result files which are only stored directly on the test machines or their connected network drives.

Questions:

* **How common is it that test equipment control files are directly generated from Unilab?**
* **How common is it to manually use external tools such as Athena or DTT to query the underlying databases and create control files based on that?**
* **How common is it for test operators to actually open Unilab and look for relevant testing parameters there?**
* **How much automation is desired for these processes in the future LIMS solution?**
* **Are there any specific requirements about the format and content of Test Request/Sample/Test Method data that is sent to the testing Equipment?**
* **Are there additional scenarios which were not yet mentioned in which Test Request data must be exported or imported?**

### Connection to specification management

High-level requirement: It must be possible to connect Test Request Samples to any currently valid Specification, in order to validate the Test Method results against the target values and limits defined therein.

Unilab process: During the creation of Samples and the assignment of Part Nos., only those Interspec Part Nos. are shown for which a Specification in the status “Current” exists. This way, it is not possible to accidentally test against invalid or historic target quantities. In order to make all Specification values available to be used as reference values, for calculations, or for logic in assigning certain alternative Test Methods, every released Interspec Specification is synced to Unilab as its own Sample Type including all relevant property values.

### Others?

Questions:

* **Is there anything else related to test data management (except for the topics of reporting and data migration, which will be covered below), which has not yet been mentioned but should not be neglected?**

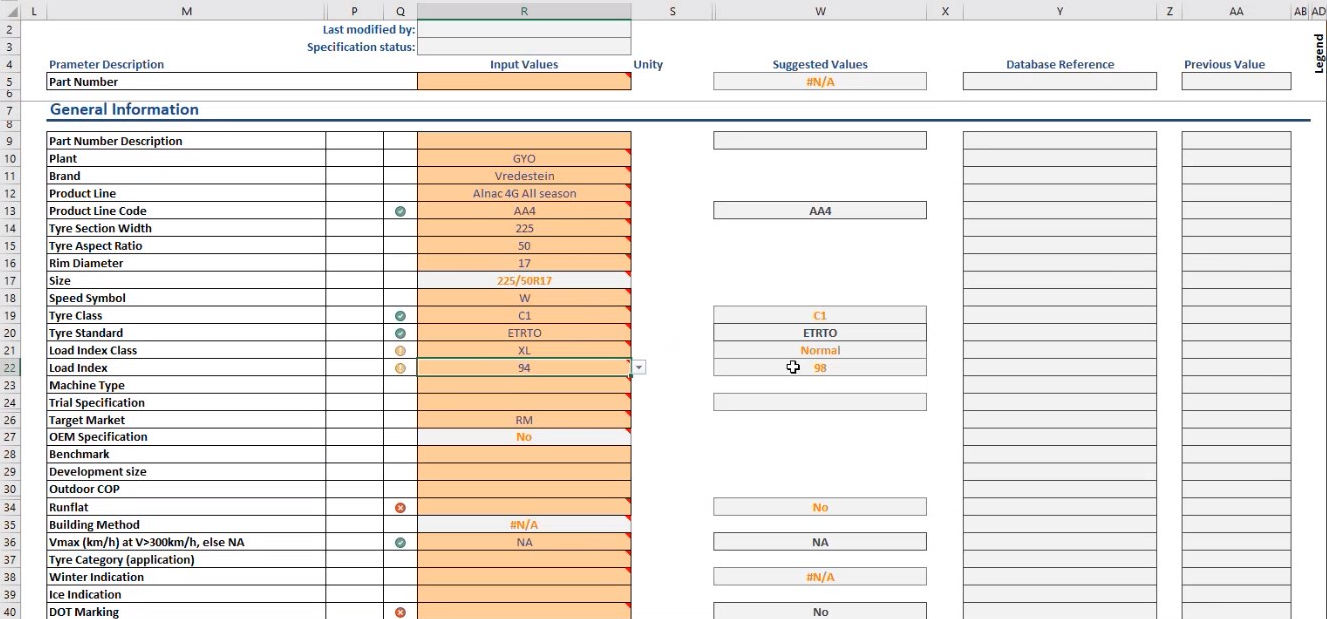
# Engineering process

## Product creation / size extension

### QR0: Marketing specification

QR0 is mainly managed out of CIM Database, marketing specifications are stored is CIM Database as a document. However, SKUs must be available in CIM Database and basic information is created in CIM Database. SAP code is also required here (marketing must create first the product in SAP).

The system should also offer the possibility to start the new specification from an existing one (load existing specification and seeing the values also on the right-hand side)



The output of QR0 is a finished good specification with only basic details.

**CONTACT will need the form details and mandatory information for QR0. Today, there are 2 different ways to create a new product or a size extension (using Xpert or Interspec), they will be unified in CIM Database.**

### QR1: Mould information

The QR0 specification details is enhanced (machines, mould details) and several disciplines are adding their details (Cavity engineers…)

*(SPPL)* Moulds are mainly managed out of CIM Database. The engineer generates from Catia and a VBA macro within Catia. These files must be stored in CIM Database – **manually / automatically?**

* A mould drawing in PDF format
* A Catia text file based on the analysis of the mould CAD file
* *(The mould drawing native format is not planned to be stored in CIM database, however, this is illogical if it is generated from this process)*

The system should inform required people about the mould delivery date.

Questions / Needed information:

* **CONTACT will need again the form details and mandatory information for QR1**
* **Please specify who must be informed and how (email alert…).**

### QR2: Specification enhancement

At QR2, the tyre engineer starts creating the product structure to be ready for prototyping (finished good, vulcanised tyre, green tyre, components, compounds etc.)

From

* Design guidelines
* OEM requirements
* Standards
* Plant limitations
* Product acceptance criteria
* Tyre specification

Are calculated:

* Valid ranges
* Optimum ranges
* Value suggestions

Some parameters are critical – value must be in strict range so parameter is valid

Filling out values can be:

* Manual
* Using suggested values
* Loading a reference tyre

There are various types of values:

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Strict | Range | Description |
| **Constant Default Value** | ✘ | ✘ | Initial value filled in, can be overwritten |
| **Conditional Default Value** | ✘ | ✘ | Initial value based on conditions |
| **Calculated Value** | ✔ | ✘ | Fixed value based on conditions |
| **Suggested Value** | ✘ | ✘ | Suggested value based on conditions |
| **Filtered Options** | ✘ | ✔ | List of values based on conditions |
| **Data Validation** | ✔/✘ | ✔ | Standards & Regulations, Design Guide |
| **Transformation Rules** | ✘ | ✘ | Old Revision ⟶ New Revision |

Some values can be selectable out of an existing table or calculated.

|  |  |
| --- | --- |
| Format | Description |
| **Business Data** | Name/Attribute ⟹ Value |
| **Lookup Table** | Search Key ⟹ Value |
| **Multikey** | (Key A, Key B, …) ⟹ Value |
| **Range Key** | (lower <= Key < upper) ⟹ Value |
| **Range Value** | Key ⟹ (Lower Limit, Upper Limit) |
| ***Compatibility Matrix*** | *(Key A, Key B, …) ⟹ True/False* |
| **Calculation** | *f(conditions)* ⟹ Value |
| **Calculation** | *Advanced code (macros)* ⟹ Value(s) |

Questions / Needed information:

* **CONTACT needs to know what kind of calculations should be supported, in addition to a detailed description of which data sources can be referenced in these calculations**
* **CONTACT will need again the form details and mandatory information for QR1**

### QR2: Extrudate Design (SPPL)

The engineer adjusts the inner and outer extrudate layout and select compounds. The system shows a real-time graph updating after each change of the engineer.

There are 2 graphs: the tread graph and the bead/apex graph

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Questions / Needed information:

* **What is the template to start from? (Understood there is 1 template per extrudate type, the Engineer should select the right one)**
* **Is there a pre-selection of compounds (EG: based on the functional description)? If yes, details are needed.**
* **What are all possible points on the extrudate the engineer can adjust? Are there some rules which must be checked.**
* **CONTACT needs details about the “flattening calculations” from the text file**

Then, the system calculates weight and volume of extrudate per unit.

Output will be stored in CIM Database:

* Coordinates values
* Compound selection **(BOM?)**
* Extrudates metrics = calculated weight and volume

Questions / Needed information:

* **CONTACT needs all details about the calculations**
* **Are calculations made for each compound layer? (Current understanding: yes)**
* **How is the chimney added automatically?**

### QR2: Die creation (SPPL)

The engineer processes the coordinates (**only?**) with Triscan and creates the die base on existing ones. The engineer selects the die to start from in the system, possible dies are identified thanks to their width.

In the system, the die consists of two objects:

* The die CAD file (stored in CIM Database)
* The die recipe which contains also properties / suggested values / calculations in the same style like Xpert or Spectrac.

Questions / Needed information:

* **How should the data load in Triscan happen / How should the storage of the data from Triscan happen (manual / automatic)?**
* **If an automated solution is required, information about how this is performed today is needed.**
* **What is the “performer CAD file”?**
* **Is it correct that profile process is performed during the die creation process? It sounds like it, however, these two processes seem independent from a non-tyre engineer perspective.**

### QR2: Profile validation

Once the engineer finished the previous steps, they must get the profile approved by the plant technology. Before that, the engineer can optionally run an FEA simulation.

Once the layout and BOM are finalised, the engineer must generate the Product Acceptance Criteria (PAC) and specify the required test for these PAC. Then, they can communicate to the plant technology for validation.

Deliverables for the validation are:

* The specification
* The BOM (containing material / compounds / components…)
* The Product Acceptance Criteria
* The cured tyre layout (CTL)
* Optionally the MFS **(?)**

If the profile is satisfactory, the Triscan file is uploaded in CIM Database. If it is not satisfactory, the approver decides what must be done again and informs the engineer.

Then the engineer requests the plan technology to approve and release the specification.

Questions / Needed information:

* **For the simulation, we need the workflows details.**
* **For the validation request, we need the workflows details as well as the rules that determine whether or not the process may proceed to prototyping.**
* **We need details about simulation data management.**
* **Shouldn’t the engineer upload the Triscan in CIM Database even before validation? That would allow it to be included in the validation workflow. Furthermore, it would help at previous step if an automatic data flow is expected between Triscan and CIM database.**
* **To which steps the process should return in case plant technology rejects one of the elements or asks for corrections?**
* **What are the links with design guidelines and certificates?**
* **What is MFS?**
* **What about the “extrusion templates”?**

### QR3: Prototyping

Once the profile is satisfactory, the engineer requests the production of a prototype in the chosen plant. After the prototype is produced, the plant technology measures the actual prototype profile value:

* They produce a 2-profiles Triscan comparison file
* They calculate again the metrics (weight and volume)

Then, the engineer compares the actual prototype profile values with the expected ones from QR2. This work is done thanks to a tabular view (in the Xpert / Spectrac style) with both prototype and QR2 values shown side-by-side. There are rules to highlight deviations and to know if they are acceptable or not, following the product acceptance criteria:

* RED 🡪 out of range
* BLUE 🡪 different but in range
* GREEN 🡪 same as QR2

A screenshot of a computer

Description automatically generated

***Is this a right screen?***

Following the discrepancies criticality, the engineering as several options:

* They can rework the specification in QR2 (change values)
* They can update / modify the PAC
* They can rework the product in QR2

Questions / Needed information:

* **Is a workflow needed for acceptance or not? In case of rework, where can the engineer potentially restart? How should the rollback to QR2 happen?**
* **Shouldn’t plant technology at this point upload the Triscan file to the system instead of keeping it on a local disk?**

### QR3: Specification release

When the comparison is satisfactory, the engineer creates the QR3 specification. To do so, they should have the possibility to copy all values or all green values from prototype specification to the QR3 specification.

Output will be stored in CIM Database:

* Extrudate prototype
* Updated prototype specification
* The 2-profiles Triscan comparison file

Then, the engineer must

* Assign the labels
* Find an existing certificate which is compatible, if not found, request new one

Then, the engineer informs the plant the specification should be released. Finally, the plant technology creates a new index of the specification according to prototype results and they approve and release the specification.

When the QR3 is released, the final drawings are released automatically by the system and the green tyre specification is released.

Questions / Needed information:

* **Should plant technology directly create a “prototype specification” which could be used as a base for the QR3 specification?**
* **What is the use of the extrudate prototype? Coordinates only?**
* **How is the label assignment managed?**
* **How are the certificates managed? What are the criteria for a certificate to be compatible with a new specification?**
* **Is there a workflow for plant technology approval?**
* **What about the cut section analysis?**
* **What about the final mould drawings?**
* **What about the homologation requirements?**
* **What about the sidewall stamping?**
* **What are the deck design and the drum requirements?**

## Size extension specific process

**CONTACT needs to know about the size extension specific documentation that was mentioned without details yet, and the specific steps that are not covered by the common product creation / size extension process**

## Design possibilities

### All CAD jobs

We have the possibility to use the existing Catia server developed by Apollo or to use the CIM Database CAD job server, which is an improved version of the Document Conversion Server introduced with version 15.8 and meant to serve Apollo requirements.

|  |  |
| --- | --- |
| Pros CIM Database CAD job server | Cons Apollo Catia server |
| The framework is supported by CONTACT and doesn’t need any interface development | Using the existing solution would require developing an interface to CIM Database and to maintain it |
| Management of input and output files is integrated and communication to Catia machine happen through Apollo network | Management of input and output files must be built and communication channel AWS to Apollo premises must be setup |
| Management of input and output metadata would be integrated, meta data would be stored in the database | Some metadata is stored in PDF files and the engineer must input it manually into the system or the integration must take care of it (adding extra development effort) |
| Smart job management: the job server queues the jobs, keeps an history, failed jobs can be reprocessed, and queue can be dispatched over several Catia machines |  |

|  |  |
| --- | --- |
| Cons CIM Database CAD job server | Pros Apollo Catia server |
| Requires configuring or developing some basics on the existing framework that are already existing at Apollo | Solution is already existing at Apollo |
| CONTACT is unlikely to support multiple Catia sessions on the same machine (\*) | Apollo saves time running multiple Catia sessions on the same machine |
| The CIM Database CAD job server runs asynchronous tasks, so the result does not popup back to the engineer’s screen, customisation would be required here | Engineers ares able to generate multiple new layout or cavity drawings on the fly |

***(\*) because it is a bad practice: due to Catia limitations, properties of one session can overwrite properties of the other one, exposing Apollo to possible errors.***

**Either way:**

* Possible errors and their management have to be specified
* The existing Catia macro could be re-used
* Machine(s) running Catia are required on Apollo premises

### Calculations

Today, the engineering process calculations are made thanks to:

* Excel formulas
* VBA macros

CONTACT should provide a framework which supports:

* All types of values that are listed in subsection 3.1.3
* As well as all formats (some requiring the creation of database tables by CONTACT)
* Python as a programming language to convert the VBA macros.

In this scenario, Apollo will ultimately be able to maintain the calculation scripts and business logic on their own.

**Note: this framework should ideally be developed in a generic manner so that it can also serve the test data management’s Test Method Cells (see subsection 2.2.9).**

# Interfaces to other systems

## General information

### Required data

For each interface synchronisation event, CONTACT needs to know:

* The direction of the data flow (CIM Database ⬄ SAP PO)
* The required data which must flow, for each field:
  + Name of the field & explanations about what it is
  + Explanation about the data type
  + Its link with other objects in the system and/or for what it is useful
* The trigger
  + If synchronisation must be trigged synchronously (EG: release of an object): the other events that trigger this synchronisation and the required condition so it can happen
  + If synchronisation must be trigged asynchronously (EG: synchronisation of catalogues for multiple choice menus): the interval of time between 2 synchronisations events and how to identify what must be synchronized
* If an acknowledgement of the operation is required:
  + If acknowledgement must be given synchronously (not preferred due to potential delay for SAP PO to answer)
  + If acknowledgement can be given asynchronously
* Error management, what happens if the synchronisation fails:
  + SAP PO returns an error
  + CIM Database did not received complete information or properly formatted information from SAP PO
* Additional information, see descriptions of the different interfaces below

### Interfaces through SAP PO

CONTACT knows that all interfaces will be done through SAP PO, including SAP instances, MES systems etc.

The only exceptions are the test machines, see below.

### Test machines interfaces possibilities

In order to avoid building an interface for each single test machine, CONTACT’s proposal is to use a file repository accessible by both the test machines and CIM Database.

This file repository would host the test machines’ output files and transform them, so they are in a unified format for CIM Database. It would be maintained by Apollo so they would be able to add machines and add new fields easily by themselves.

Then CIM Database would be able to read these files, and this second stage of the interface would be maintained by CONTACT.

Furthermore, this file repository would be useful to store control files from CIM Database to the test machines.

## Interface to SAP

### Part number generation

While it was at previous points discussed whether generation of new part numbers should always happen on the SAP side, the current understanding of CONTACT is that new Part numbers are manually created by users inside of the CONTACT system (according to certain limitations on the allowed length and character sets) and transmitted to SAP, where the corresponding objects are then created as well.

Questions:

* **Is this correctly understood?**
* **Can we guarantee that users cannot create Part numbers in the CONTACT system which, for whatever reason, are not valid to be used in SAP?**
* **If not, what is the procedure for dealing with this scenario?**
* **Recommendation from CONTACT and best practice: a unique identifier should be unique and valid for the whole life of the object**

### Synchronization of specifications

This has been discussed in multiple sessions, and at least for the interface for Raw Material specifications first overviews over all relevant information to be sent to SAP was provided by Apollo. The actual interface has not been implemented yet.

For the full interface, many open questions must still be finalized:

* **Which Specifications must be sent to SAP? Are there any kinds of Specifications that must not be sent (such as Raw Material Supplier Specifications, certain product lines, trial Specifications, …)?**
* **What is the exact trigger point at which information must first be sent? Likely this should happen at a fixed moment during the review/approval workflow, since it does not make sense to update SAP after every change in a draft Specification.**
* **What kind of responses can be expected from SAP (both instantaneously and delayed after some SAP user has performed a manual action)? What should the result of these responses be? Can certain steps in the CONTACT system (such as a status change to “Released”) only happen after SAP has given the green light?**
* **What are the differences, if any, between entirely new Specifications and new revisions of existing Specifications?**
* **What additional information must be transmitted to SAP for Specification at higher levels than Raw Materials? Will BOM information be sent out, and if yes, what information exactly?**

### Synchronization of catalogues

See also subsection 6.1 in the data migration section. For many of the attributes which are to be transferred from CONTACT to SAP, only a limited range of values is allowed, because they are maintained as catalogues in both systems. In order to avoid any errors in the system, it should ideally never be possible for the CONTACT system to transform a value to SAP which is not recognized (less ideally, there should at least be a well-defined procedure for recognizing this as an error and for resolving it).

To guarantee that these catalogues are kept in sync, the following must be agreed:

* **What are all relevant catalogues to be kept in sync between SAP and CONTACT?**
* **Which system is the leading system for any of these catalogues?**
* **What is the procedure for updating the non-leading system after a change (adding, renaming, deleting) of any catalogue entry in the leading system? Should this also be an automated interface, or should somebody have the responsibility of manually performing the same update in the non-leading system? How is that responsible person informed about any new relevant change?**

### Synchronization of pricing

How to deal with pricing of Raw Materials was under discussion at various points in the project, resulting in CONTACT’s understanding that any sort of price maintenance (or just display) inside of the CONTACT system is currently out of scope.

**Is this correctly understood, or is Apollo still expecting some sort of integration to display RM prices, which are maintained in SAP, inside of the CONTACT system?**

**If such an integration is still necessary, it must be defined how the information should be kept in sync. Should SAP actively update CONTACT after every relevant change to a price? Should all prices be sent out in bulk in regular interface and updated in the CONTACT system where needed? Should CONTACT actively poll prices, either on a regular basis or on demand?**

### Synchronization of projects

This was not discussed in much detail so far, but at some points in the Unilab and Athena user story sessions it was mentioned that project information is currently being synchronized from SAP to these systems, a setup that would presumably also need to be replicated for the CONTACT LIMS solution.

Questions:

* **Would SAP be the leading system in this case, i.e., should new projects (or certain kinds of projects) only be created in SAP and then synced to CONTACT?**
* **What information will be interfaced from SAP to CONTACT? To what degree should it be possible to add additional metadata in CONTACT in order to enrich the information and to increase searchability etc.?**
* **Would this interface involve synchronizing anything else? Is information ever sent back to SAP?**

## Interface to WMS

The integration to the external WMS tool was brought up during the user story sessions for outdoor testing (see also subsection 2.1.3), and the tool itself was demonstrated as part of the Enschede workshop.

CONTACT’s current understanding is that the tool is currently somehow informed about new Worksheets for outdoor tests which would require the ordering and mounting of tyre samples in certain locations/warehouses. Once these samples are available, the WMS tool communicates back to the system using the Unilink format, and tests can be scheduled & carried out.

**For a proper estimation of the integration, a more detailed written description of the use cases and the interfacing processes would be needed.**

Questions:

* **Does Unilab currently actively push any information to the WMS tool, or is the tool querying for any new relevant information on its own?**
* **Can the WMS tool be adapted in order to change the ways in which it is sending (and potentially querying) information to the LIMS system? Or should the approach be to change the WMS tool as little and possible and instead parse its current Unilink output to perform the corresponding actions in the new CONTACT system?**
* **Is it planned to roll the WMS tool out across all Apollo locations/warehouses, or will other, separate solutions need to be found for the Indian locations?**

## Interface to Test Machines / Tools

### Outgoing calls & generation of control files

The questions around this topic are already mostly covered by subsection 2.3.4.

**Mainly it needs to be agreed to which degree of automation, under which circumstances, with how much information and in which format information is supposed to be generated directly from the system with the purpose of sending it to external test Equipment or intermediary analysis and test preparation tools.**

### Incoming calls

The main source of incoming information from test Equipment and similar external tools, is expected to be for the recording of test results. In the current Unilab solution all such communication is handled using the Unilink format, which allows a wide range of operations to be carried out within the software.

In some scenarios Unilink may also be used to trigger entire testing processes, for example for regular testing according to control plans, or in the case of simulation tests through the Desktop Tool.

**It must be decided whether the format should be kept the same, with responsibility on the CONTACT side for translating any Unilink commands and Unilab object identifiers to the correct actions in the new system, or whether CONTACT should define the optimal new format for addressing the system, with responsibility on Apollo side to update all external systems and tools to this new format.**

For the first option, the advantage would be that most of the current test Equipment code would not need to be touched. Disadvantages include the need for maintaining links to the old Unilab data structure in places where they may no longer be needed, just for legacy reasons, and the need for introducing the Unilink format in Indian plants where Unilab has never been used and will never be used.

For the second option, updating all systems which are currently using Unilink to communicate with Unilab may present unexpected complexity, in case that certain systems have not been touched in a long time and that updating their code may prove more complicated due to lack of know-how or personnel.

**Additional question: are there any cases where external machines or tools are not just sending Unilink control files in order to record results or trigger certain actions, but are actively querying Unilab data? (Similar to Athena for example, in order to display certain information related to Test Requests)**

## Interface to CATIA

*See section 3.3.1.*

# Reports

This section outlines information about the various kind of reports, most of which currently exist in some form in Apollo’s Athena tool, which are (or might be) expected to become available directly through the new LIMS solution.

## General overview of reports

### Required Data

**CONTACT needs a consolidated list of all reports that have to be generated by the system itself (and must therefore be implemented within this projects and will not be created using external reporting tools, such as the PowerBI reports covered by subsection 5.1.3)**

* With samples
* Explanations of the fields / columns / line
* The origin of the information which is contained in them (so that it is clear how each value in the report can be retrieved from the correct Specification Properties, BOMs, Organizations, Test Methods, etc.)
* Explanation of the use cases and circumstances where the report is generated (Automatically vs. on demand, which user groups do this, is this part of some larger workflow/process, how often is it generated, does it need to be stored as a document in the system, …)

And more if necessary

**In the first reporting user story session it was mentioned that there is a functional description of all Athena reports with screenshots and explanations available in Apollo’s Watson system stored in the “Knowledge documents” section. Was this information ever collected and shared?**

### Report types

**For each report, CONTACT needs the category under which one it falls among these:**

1. No specific form is required, output could be a search query, a dynamic display, an Office document, a PDF – for these kinds of reports the important thing is that users have a convenient way of requesting and viewing certain kind of information, but the solution can be flexible.
2. Exportable documents (Office document, PDF) without specific layout or format – same as before, the main focus is that people want a convenient way of viewing certain information, only in this case there must also be some way to download it to a file. (For example, to view it on systems without internet access, or maybe to print it out)
3. Same as above, but with file a specific file format (Excel, Word, PDF…)
4. PDF export with strict layout requirements – This is like the use case of the Raw Material report that is sent out to suppliers, which had to follow the established format as closely as possible.
5. Workflow approval report (with date & name of user) – This is similar to the SCN and SDN documents for Raw Materials, these kind of reports should be generated automatically during (or at the end of) workflows to serve as an official record for auditing purposes (who signed off on what, and when?)

**Please specify anything that doesn’t fall into these categories.**

### Others

**CONTACT needs a consolidated list of PowerBI reports where its support is needed to locate data in CIM Database.**

## Specification management Reports

This section describes various reports related to Specification management that were already discussed or initially presented during the reporting user story session in 02/2023. Because the project target scope, especially with regards to reporting, has been under discussion since then, it is important to also re-evaluate these reports under the criteria of the previous section, i.e.: **which of these reports should still be implemented in the system, which should be handled by external reporting tools, which of these reports must follow specific formal requirements, and which of these could be covered by entirely different solutions as long as the basic information can still be gathered in some way?**

### PDF Export based on Specification filters

The Specification filter and report creation tool was already delivered for raw material management, and was built as a generic functionality to make it possible to create similar types of PDF reports for any type of Specification (including options of including images, hiding headers, renaming columns, …)

Preliminary support for including BOM structures in these reports has also already been built but was not yet presented to and tested by Apollo.

**If any additional features are needed for this default PDF export tool, they must be specified.**

### Full Specification export

Apollo needs a report which would be a full specification export with all properties to be able to compare the data with InterSpec (especially during the duration where both systems are being used).

The form and information to be included in this report was already discussed, but smaller additional alignments will likely be necessary once the first version of this report has been created and shared with Apollo.

### First Product Check Card

High-level requirement: To verify that newly developed tyres are fulfilling all specified requirements, there is a need for manual verification of all important properties from the finished tyre, cured tyre and greentyre Specifications, carried out by multiple people from different teams / development stages, in order to remove any room for errors and to serve as official documentation.

Athena solution: There is a report which is currently only used by the GYO plant. It accumulates property values from multiple linked Specifications: the finished tyre, the cured tyre and the green tyre. The purpose of the reports is to be printed out as a checklist, which is then used in up to four different steps (Mould, Curing, Prod, QA) to verify that each property of the finished tyre is matching the Specifications.

In the current system the information about which properties from the complete Specifications are shown in this report is derived from so-called mask Specifications. These are created like regular Specifications for the respective frames, but instead of real property values they have a value of 1 to mark properties that should be included in a report, and no value to mark properties that should be omitted.

Questions:

* **As this report is currently only used by one plant, is it actually important enough to be worth implementing in the new system?**
* **Is it necessary to have the report as a printable document which is tick-marked by hand, or would an in-system solution also be an option (there could be a workflow in which people get the task to review properties, which they can then tick off a checklist in the browser – these functions are already available in the standard)?**
* **Most properties in the report only had a single value to be checked. But for some there were also multiple columns (target value with upper and lower tolerance, for example). What are the different kinds of scenarios that can occur here? How freely should this be configurable?**

### Multi-level Targeted Specification

High-level requirement: For certain situations and user groups, such as getting an overview of a greentyre and its components, it is requested to have the capability of defining report templates that consolidate selected subsets of important data from multiple source specifications.

Athena solution: Similar in the basic principle to the “First product check card” report, this functionality can be used to create a singled report that is consolidating data from multiple source Specifications which are linked through their BOM. In the presented example, which is assumed to be the most common use case, such a report was generated for a Greentyre specification, containing data from the Specifications of its various BOM items (Sidewalls, Tread, Belts, …).

The report starts with metadata and the first level of the BOM for the greentyre (showing each of the greentyre’s child components, but not going down to any underlying levels with Compound, RMs, etc.).

This is followed by a list of important properties with UoM, target and upper/lower limit values for each of these components, grouped by their functional code.

Afterwards there are additional groups of properties which mainly appear to be coming from the processing section of the greentyre Specification, but it is not quite clear whether some information may also come from the Specifications of the various BOM components.

The configuration of which properties appear in the report is once again handled with the Interspec mask Specification approach, where the BOM of the green tyre mask links to all relevant child mask Specifications.

Questions:

* **Standard questions: should this still be handled inside of the system, or left to external reporting tools? Is the actual form flexible as long as there is a way to configure which information should be exported?**
* **This could most likely be implemented as an extension of the already existing functionality around Specification filters and PDF exports (see subsection 5.2.1). Would a PDF export be acceptable as a result, or should the report be displayed as a web page? (Generally it might make sense to always also offer a web view as an alternative to the PDF export)**

### Targeted Specification

In Athena, this is the simple version of the report from the preceding subsection, where selected information from a single Specification is included (it may also show that Specification’s BOM, but it will not extract any additional properties from other Specifications which appear in that BOM). This is also configured using the mask Specification mechanism.

**We believe that the full functionality of this report is already covered by the existing functionality for Spec. filters and PDF export (see subsection 5.2.1) and no further development will be necessary, unless a single-page web view instead of a PDF export is required.**

### Interspec analysis / Specification overview reports

High-level requirement: For certain scenarios it can be useful to have full export of all data for all Specifications derived from a certain frame (or all data from certain frame sections), for example to scan for missing information or identify possible duplicates.

Athena solution: The “Interspec analysis” report creates an Excel export for all Specifications derived from a selected frame. One or multiple relevant frame sections can be chosen.

For each property in each property group in each subsection this report lists the values of all columns for each Specification (e.g., one out of many such blocks in the full report could be used to show all target values, all lower limits, all upper limits, etc. for the Mooney viscosity for every polymer raw material in the system).

The result is therefore usually very large and unwieldy, depending on the amount of properties in a given frame and the amount of Specifications derived from that frame, and may require the use of additional transformations and pivot tables in Excel to extract useful information.

It was noted as a disadvantage of the current system that the report always exports all data from a selected section, where often the user might only be interested in a specific property group, or even just specific columns for a certain property.

The “Specification overview” report is similar in principle, but instead of selecting specific sections, this report uses the mask functionality described above and always includes all properties which are included in the selected mask.

Questions:

* **How many people actually use these reports, and is this maybe a case that would be better suited for external reporting tools? (Since it may often require additional manual analysis steps in Excel anyway)**
* **Could this maybe be covered by or combined with the report outlined in subsection 5.2.2? In that case, all data for all Specifications of a certain type is also exported, but grouped by Specification and not by property. So in the end it would contain the same data as the full Interspec analysis report, only in a different structure.**

### Tyre Composition & Weight

High-level requirement: There should be a convenient way to see both the full BOM tree of a tyre, with a view of the various weights and quantities of its sub-components, as well as a flat summarized BOM of all Raw Materials and their respective weights.

Athena solution: The “Tyre Composition & Weight” report is an interactive report with two different views. The default view shows the full BOM tree of a selected tyre Specification. For each element in the tree its quantity is given either in kg, m, m² or pcs.

There is a link to open the Raw Materials view, which shows a flat list of all Raw Materials that are found at any level of the BOM tree, with their kg weights. If the same Raw Material shows up multiple times at different positions, its weights are summed up. At the bottom the total summe up Raw Material weight of the tyre is displayed as well.

Questions:

* **Is the BOM tree view that is already available on the “Product Structure/BOM” tab or in the xBOM manager sufficient to cover the first part of this report?**

### Compound and mixing reports

At various times, different example reports for Compound Specifications with slightly different formatting were shared by Apollo.

The common ground seems to be that there should be at least one kind of report which gives an overview of the recipes for a final mix Compound and all preceding master batch Compounds, with additional information about certain mixing steps. Also, there needs to be a compact view of any single Compound Specification for use in a specific mixer (hence including information from at least one alternative BOM) with information about its recipe (grouped by types of ingredients and/or chemical bags), about the necessary processing/mixing steps, and some accumulated metadata (such as the specific gravity). This second type of report is used in the plants (either viewed on a computer or printed out?) to serve as a guideline for the mixing process.

**Apollo needs to determine the minimum number of such reports that must be included in the system, and must clearly define which information has to be included, where it is coming from (so that the reports can automatically be generated from the involved Specifications and do not require manual user input), and whether there are any hard limitations on the format (e.g. “the report must fit on one printable page”)**

## Test data management reports

### Test overview report

High-level requirement: Users need a convenient way of viewing all important information related to a Test Request, with a special focus on being able to view test results across all involved Samples and Specifications.

Athena solution: The current Athena report for test results, “Request overview and results” was partially demonstrated during the Unilab user sessions. It is an interactive report with different views that users can use to view all relevant information for any given Test Request. A header shows the main metadata of the Test Request (i.e., description, status, relevant dates, the associated Project, motivation, …). Summary tables display the number of Worksheets, Samples (grouped by Part No.) and carried out Test Methods, with clickable links to go into various detail views. Only some of these links were demonstrated.

One core feature is the option of being able to open a side-by-side view of all test results for all Test Methods for all involved Samples, in order to make it easy to compare the range of test results across multiple similar Samples. These results can be displayed as mean values, or with more detail and with comparison of additional detail fields (likely coming from further Test Method Cells of the Test Methods, which are not the “official” result cell)

Questions:

* **What is the full range of information that needs to be available to users? I.e., what information should be seen in the first overview, which detail views and comparisons have to be available on further request?**

### Test planning report

High-level requirement: Users from certain testing departments need a convenient overview for seeing how many Test Methods/Parameters are scheduled in a certain period for a certain location.

Athena solution: The “Planning view” of the interactive report “Outdoor testing” was demonstrated during the user story session on reporting. The report shows, grouped by outdoor test track, how many test runs were scheduled for any given Parameter in any given week (as well as how many are still unscheduled).

By selecting a certain week (and optionally also location), this is further broken down and shows, grouped by project, Test Request and Sub Program ID how many test runs for a certain Parameter are to be carried out (and on which car, with which tyre and which priority)

Questions:

* **Is it correct that this is mainly used by the outdoor testing department? Do other testing departments have similar kind of planning overviews? Do they need them if they do not have them now?**

### Test requester overview

The “Requester view” of the interactive Athena report “Outdoor testing” was briefly shown, but not explained in much detail. It appears to offer the option of showing on the first level all Test Requests belonging to a certain project, made by a specific requester, as well as for which week(s) their tests are scheduled, how many Worksheets there are and what the status of these tests (Worksheets?) is. Opening any given Test Request in this requester view than shows further details about the Samples, with links to the various Worksheets for each Sample and Sub Program.

**More details, as well as the exact use cases for this report should be further specified.**

### Test preparation overview

High-level requirement: Certain users need to have a convenient way of seeing how many tyres have to be mounted prior to testing in any given week, grouped by testing location and Test Request.

Athena solution: The “Preparation view” of the interactive Athena report “Outdoor testing” is a three-level review that accumulates this information. On the top level it shows the number of tyres to be mounted at any given test location in any given week. Opening a specific week then shows the number of tyres further broken down, grouped by the Test Request and Sub Program ID. Opening a specific Test Request then shows explicit details for each Sample and Subsample within this Test Request (Part no., description, rim width, brand, …)

### Tyre evaluation sheet (TES) report

This is one of the cases where Athena is currently used to generate an intermediate file which is used for external testing machines or programes, as addressed in subsection 2.3.4. For a given *outdoor* Test Request, this report generates overviews of all Worksheets, Samples, Subsamples and most importantly an overview of all Test Methods which must be carried out for each individual subsample. This file is used as input for the test result recording MATLAB tool (is this ANITA?) which is used by the outdoor testing department, so that this tool can in the end generate the proper Unilink files to correctly store all test results in the correct places within Unilab.

The users stressed that this current way of working is not ideal, especially since the control file is quite large and takes a long time to generate and to open, so a better (or at least faster) solution would be preferred.

**The exact requirements and use cases should be defined, so that an optimized solution approach can be developed.**

### Conformity of Production

This was shown only very briefly and without much explanation, but it appears to give an over all AV Test Methods which were performed on tyres of a certain product line in a certain time range, highlighting the number of measurements per paramater and color-coding any measurements that were outside the warning or spec. limits.

**What are the exact use cases and requirements here? Are there other related reports/overview which are needed?**

## Other reports

### Size extension report

This was also only shown briefly, but appears to be a very large report based on many different input sources. It shows, grouped by QR phase, which tyres Specifications exist, what their linked moulds are, whether these are available, what prototypes exist, what tests were carried out, what certificates exist, how many tyres are in stock, and a lot more.

**If this report must be replicated in a similar form, a much more detailed description of the various information collected in this report must be provided.**

### Other reports

**Any additional reports that do not fit into either the Specification management or Test data management sections need to be explained, with all relevant information as given in introduction.**

# Data migration

This section is intended to give an overview of all outstanding data migration tasks.

## SAP harmonisation

Apollo needs to harmonise all CIM Database catalogues with SAP catalogues where it is needed. As of today, CONTACT knows about the following:

* Material groups
* Article group PG
* UoM
* Specification types (= Three-level classification from Interspec)
* Specification categories (“Raw Material”, “Compound”…)
* Apollo plants

**For each case, Apollo should provide a full list of values that will be accepted by SAP, so that the system can work together. Furthermore, once a full interface to SAP is in place, these catalogs must be kept in sync either through automated interfacing from one system to another, or through clearly defined processes where a responsible person is always notified after a relevant change in one system, so that the second system can be updated accordingly.**

## Specification management

### Frames

Raw material, compound and some semi-finished good frames were already imported or created in CIM Database, so they are excluded from this paragraph.

The same migration script that was used for these frames is also expected to work for any outstanding frame migrations from Interspec, as it was written to work with any kind of Interspec frame.

Apollo started to consolidate the list of frames with information about the ones that are planned to be manually created by Apollo (rather than imported). This list should be updated and completed.

**CONTACT will also need:**

* An updated overview of how many frames will still have to be migrated using the migration script (to make a proper estimate for the total effort)
* An export of these frames in the same format as the most recent frame exports (which were various SFG frames)

### EU Specifications from Interspec

Raw material specifications were already imported or created in CIM Database, so they are excluded from this paragraph. Additionally, many compound and SFG Specifications from Interspec were also already migrated.

The migration script was written to work with any Interspec Specification, so even for upcoming categories where no data was migrated yet, it is not expected that additional programming effort will be needed.

**CONTACT will need:**

* For each frame, roughly how many Specifications are expected to be migrated from Interspec (for time estimation)
* An export of the specifications in the same format as the last batch of Specifications (for the newly created and updated RM Specs, including the additional metadata of the internal Interspec IDs)

**Additionally, CONTACT needs to know if any special procedures are expected to be followed for Specifications which were created using an older frame revision. For all Specifications that were migrated so far, this was not the case.**

### APMEA Specifications from Excel / SAP

While migration scripts exist and are well-established for migrating Interspec data into the CONTACT system, there is currently no dedicated script or finalized approach for migrating APMEA Specification and BOM data.

As a possible solution it was suggested by Apollo leadership that it should be Apollo’s responsibility for providing data in exactly the same input format as was provided by Interspec exports.

**If this is a feasible solution, the existing migration scripts would suffice and no additional effort from CONTACT (beyond the standard effort for running and validating any migration) is to be expected.**

### Specification documents

**CONTACT needs to know if there are documents to be migrated for specifications**. Today, we only know about the safety datasheets for raw material (already migrated). If there are more, they need to be explained and provided.

The existing migration script for Safety Data Sheets is expected to be mostly reusable for other document migrations from Interspec, with minor variations (to write additional metadata or use different document categories).

**If there are additional documents from other sources, for example relevant documents from the APMEA plants, that need to be mass-migrated into the system (instead of manually created by Apollo users), then the creation of additional migration scripts will be necessary.**

### Keywords

**The basic functionality for handling frame and Specification keywords similar to Interspec was already implemented as part of the Specification management for Raw Materials. If there is additional keyword functionality that only concerns non-RM Specifications or Frames, it must be specified.**

Keywords for raw material were already imported. The migration scripts for frame and specification keywords are expected to be completely reusable for any upcoming keyword migrations.

**CONTACT needs exports of Keywords for all Frames and for all migrated (and not-yet migrated) Specifications above the RM level.**

### Display formats

Updated display formats were already imported, last updated on 11/09/2023.

**If any new display format were created since, a complete export is needed in the same format like the previous one.**

No additional programming effort is expected if the format of the source data does not change.

### BOM functional description

The full Interspec BOM export contains information about the functional description of BOM items, however it only contains the internal Interspec ID of these functional descriptions and not the actual labels which would be displayed in the system. This data has not yet been imported into the new LIMS solution and it is not yet possible to assign a functional description manually.

**To make this functionality available and to migrate the functional descriptions for all BOM items, CONTACT needs**:

* A full export of all functional descriptions (with code and text).
* To know how they are used or connected by other objects (see Section 1.8.2).

A migration script for this purpose has not yet been written.

### Work Centers

Work Centers information contained in the different specifications were imported, but we do not know if information is up to date.

**If required, CONTACT would need**:

* A consolidated list/export of all Work Centers with all relevant properties that should be maintained in the system (as a minimum: name of the Work Center & Apollo plant to which it belongs)
* To know if there is additional use than for the specifications, how they are connected to other objects (see Section 1.8.1).

A migration script for this purpose has not yet been written.

## Test Data Management

### Relevant Unilab objects

For the migration of the current test management functionality from Unilab to the new LIMS solution, migration scripts and approaches will need to be developed. The exact details of how the information and functionality for each kind of object will be treated in the new system will of course be strongly based on the answers to the outstanding questions related to test data management.

**Whatever solution is arrived at, full Unilab exports of the following configuration objects and all relevant metadata will be needed:**

* Test Request Types
* Sample Types (and any n:m links to Request Types)
* Test Plans (in case these are distinct objects, see section 2.2.5)
* Parameter Group definitions (and any n:m links to Sample Types or Request Types)
* Parameter definitions (and any n:m links to Parameter Groups)
* Test Method definitions (and any n:m links to Parameters)
* Test Method Cells (including all formulas and VBA scripts that may be used by them)
* As explained in section 2.2.11, it is currently not fully understood whether there are different kinds of Worksheets which may have their own templates (Worksheet Types?). If that is the case, these templates will likely also need to be exported.
* Any additional tables which may include important metadata, if those are not yet fully covered by the above
* Info Cards (and all Info Card cells), and n:m links to Request Types, Sample Types or other objects which can have Info Cards
* Equipments
* All Attributes, constants and values which may be attached to any of the above objects and might play a role in formulas, Method Cell input values, etc.
* Access rights management: groups overview and explanations

**In the user story sessions, it was mentioned that the Enschede plant and the GYO plant are currently using different, incompatible versions of Unilab. Therefore, it is expected that separate exports from both systems may need to be made and harmonized, leading to additional efforts (probably both on Apollo and CONTACT side) to avoid unnecessary duplication or data inconsistencies.**

It is not currently expected to migrate any test results, therefore it will not be necessary to export Test Request, Samples, Worksheets, or any result data which was entered into or calculated for Test Method Cells.

**For testing and demonstration purposes, it will likely be sufficient to create Test Requests and Samples manually, but it may be advantageous to have screenshots or Athena reports for several example Test Request which cover a range of scenarios, so that the functionality and information in the new LIMS system can be validated against these examples.**

### APMEA test management data

Currently the APMEA plants are not using Unilab to keep track of their test management, but of course all relevant information which is specific to the APMEA plants must also be entered into the new system.

A migration approach has not yet been discussed, and the expected amount of effort can therefore not be estimated at the moment. One scenario for minimal effort from CONTACT side, similar to the suggested solution for the Specification migration, would be for Apollo to prepare all APMEA-related data in the same format as the Unilab exports from the European testing departments. But, depending on how different the test management will be handled in the new CONTACT system in comparison with Unilab, it may actually be more sensible for the APMEA data to be prepared directly in the most suitable format for importing into the CONTACT world, instead of first trying to adapt it to the Unilab format only to be able to reuse the same migration scripts.

Depending on the amount of information that will have to be migrated, and the variety of how it is currently stored, manual data entry may ultimately be another viable option.

### Documents related to test management

It is currently not known if there are any additional documents and files related to test data management (for example operational instructions for test equipments or certain Test Methods) which should be stored in the system.

**If there are, please share the types of documents and rough estimates for the amount, so that it can be estimated whether developing a migration strategy is necessary or whether manual import would be sufficient.**