Streamlining LOFAR operations with the new telescope manager specification system

Sander ter Veen[a], Jan David Mol[a], Jorrit Schaap[a], Jörn Künsemöller[b], Auke K. Klazema[a], Reinder Kraaij[a], Arno P. Schoenmakers[a], Fanna Lautenbach[a], Hannes Feldt[a], Corne Lukken[a], and Stefano di Frischia[c]

[a]ASTRON, Oude Hoogeveensedijk 4, 7991PD, Dwingeloo, The Netherlands
[b]Bielefeld University, Universitätsstr. 25, 33615, Bielefeld, Germany
[c]INAF - OAAb, via Mentore Maggini snc, 64100, Teramo, Italy

ABSTRACT

The Low Frequency Array (LOFAR), one of the worlds largest radio telescopes, is a complex instrument which had an manpower-intensive manual workflow and became difficult to adjust. The new Telescope Manager Specification System (TMSS) resolves this by the introduction of a dynamic scheduler, a data-quality assessment workflow and a specification system that allows easy versioned specification of known observing setups but also detailed adjustments of observations and processing pipelines.

Keywords: Telescope Manager, Specification Software, Radio telescope, Dynamic Scheduling, SPIE Proceedings

1. INTRODUCTION

An essential part of running a telescope is the specification and scheduling of the observations. This is also the case for the Low Frequency Array (LOFAR[1]), one of the worlds largest radio telescopes. We recently updated our specification and scheduling software into a single modern web application, the Telescope-Manager Specification System (TMSS)[*] . In this process, we incorporated lessons from 10 years of LOFAR operations to streamline operations and improve the structure and maintainability of the software itself. We took special care to easily enable new future functionality for this telescope. In the past, tasks like fine-tuning specifications and the scheduling were largely manual tasks, to be carried out on a daily basis. TMSS has extensively automated these tasks and allows for bulk handling of these. TMSS started to be used in operations since April 2022, and has been fully responsible for LOFAR observations as of June 1st 2023.

LOFAR consists of 52 stations in 9 European countries with its center in the Netherlands. LOFAR can run a wide variety of observations simultaneously, from wide-field interferometric observations to high time resolution observations to individual antenna voltage read-outs. The LOFAR stations consist of fields of two types of antennas, the low band antennas (sensitive in the frequency range from 10-90 MHz) and the high-band antennas (sensitive from 110-250 MHz). LOFAR is a phased-array telescope. This allows multiple pointings to be observed simultaneously, both at the station level and at the correlator, which adds a level of complexity to the observation specification. LOFAR currently runs around 6000 observations per year in 4000 scheduling blocks supporting 30-40 projects per semester.

Currently, LOFAR is undergoing an electronics upgrade, called LOFAR2.0.[2] Efficient use of the new capabilities became possible through TMSS as part of an overall improvement in operations.[3] For example, we can now support the large survey program that will run several science projects in parallel for 50% of the time over a 5 year period.

In this paper, we describe some of the improvements of the telescope management specification system software and the benefits that these improvements have brought us by operating this telescope more effectively.

Further author information: (Send correspondence to S.t.V.)
S.t.V.: E-mail: veen@astron.nl

[*]TMSS is open source software and is available at: https://git.astron.nl/ro/lofar/-/tree/master/SAS/TMSS

Observatory Operations: Strategies, Processes, and Systems X, edited by Chris R. Benn, Antonio Chrysostomou, Lisa J. Storrie-Lombardi, Proc. of SPIE Vol. 13098, 130981Z · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3018969

2. SPECIFICATION

Specification of observations is administered within a project, the result of an approved observing proposal. Projects are executed within a cycle of typically 6 or 12 months for which these proposals have been approved.

The specification is divided into three levels. At the highest level we have a scheduling unit (similar to a scheduling block on other instruments like the Karl G. Jansky Very Large Array (VLA) or the Atacama Large Millimeter/submillimeter Array (ALMA).[4] At the second level we use 'tasks'. At the lowest level, we use 'subtasks'.

In our concept, the scheduling unit is used as the basic specification unit. It describes the observations and pipelines to be executed and the relations between them. It also contains instructions for scheduling by specifying constraints, priority levels and array requirements.

The scheduling unit is created from an observing strategy. The observing strategy is a template for a collection of observation and pipeline tasks with some fields of these tasks pre-defined and some fields user-specifiable. It describes the tasks and relations and which parameters of the tasks can be changed when creating or editing a scheduling unit. Here, depending on the strategy, only a small or medium set of parameters such as target coordinates, durations, filter selection and frequency range needs to be specified by the user. The user-defined parameters can relate to a single task or multiple tasks in a scheduling unit and even to multiple settings within a task, for example the duration of all calibrators or the time averaging for all pipelines, which helps to minimize the specification effort.

The tasks describe individual observations, pipelines and managing tasks such as data transfers to the archive and data cleanup processes. They are automatically created when creating a scheduling unit, based on the settings in the template and the user-specified parameters. The user can view and edit all settings, beyond the parameters available on the scheduling unit.

A task itself consists of one or more subtasks. While the tasks contain the human readable interpretation of the settings of for example an observation, the subtasks contain the specification as system parameters. The subtasks manage the processes that actually run on the telescope, and they track the progress and gather feedback information. There are different subtasks for observations, pipelines, but also for data-quality-assessment processes. Subtasks are created automatically when a task is approved and are not user-editable.

The specification of scheduling units can be done in three ways. Firstly, a single scheduling unit can be specified in a form that is auto-generated from the model by selecting the appropriate observing strategy. In this form the station requirements, scheduling constraints and priorities, and scheduling unit parameters can be specified. Secondly, to create or edit multiple scheduling units at once, a spreadsheet editor is provided (Fig. 1). Instead of the provided editor, also external editors such as Excel can be used for this, after which the specification values can be copied back into TMSS. This allows for easy generation of scheduling units based on an external user source list with properties to be translated to parameters, like the duration of the targets. Lastly, the TMSS web interface uses the TMSS API, but this API can also be utilized in user-written scripts directly. This allows, for example, the creation of test observations or fast specification of a large number of observations for a survey.

The specification model as described above allows for easy adjustment of the operational model, by defining new versions of tasks and observing strategies or by creating new types of tasks or observing strategies altogether.

3. DYNAMIC SCHEDULING

One of the exciting new developments that TMSS brings is that of automated, dynamic scheduling of observations and other tasks. Dynamic scheduling chooses the best next observation from a queue of available observations based on priority and scheduling constraints. To understand this concept we explain how priority and scheduling constraints are defined and implemented for LOFAR.

The scheduling unit priority is set on a few different levels. An overall priority rank is assigned to a project (the administrative unit for an approved observing proposal), based on the scientific ranking. A secondary priority ranking is set within a project, where each scheduling unit is assigned a rank from 0-1 and a priority A or B. Priority A units should be scheduled during the cycle and are eligible for repetition in case of failure, while

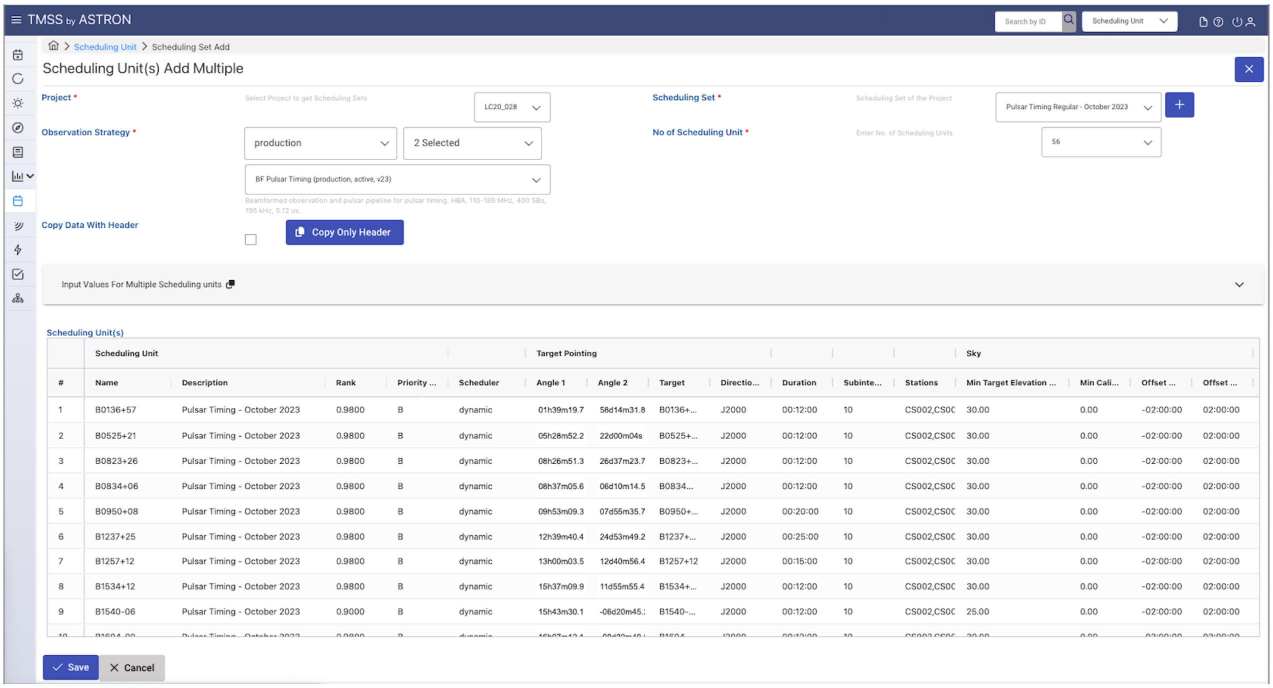


Figure 1. This spreadsheet editor allows quick specification and editing of multiple scheduling units. The selected observing strategy determines which parameters the user can specify. In this example, the target pointing, duration and integration time are sufficient.

priority B units can run under favourable availability circumstances, but are not eligible for repetition. In the creation of the schedule, the priority A observations are therefore scheduled first and the priority B observations are scheduled in the remaining gaps, based on an overall weighted rank.

The dynamic scheduler predicts the schedule taking several scheduling constraints into account. A few examples of these are: time constraints, closeness to known interfering radio sources, minimum elevation, and transit offset. There are time constraints for scheduling at a specific time, preferably using the "fixed time" scheduler setting, between or not between specified time windows and before and after given points in time. The scheduling units will also only be scheduled within the cycles assigned to a project. The quality of the data in radio is also influenced by the Sun, Moon (blocking sources) and Jupiter (below 30 MHz). Therefore a minimum distance constraint for these sources is also available. There are further constraints on the minimum elevation of the targets and calibrators. Lastly, there is a transit offset constraint. This will either hold for all targets within a scheduling unit or can be set to hold for a reference pointing.

The dynamic scheduler also takes into account the required station availability. The requested stations and the minimum amount that should be present are set according to the default data policy or scientifically motivated requests to include for example the outermost stations for the best spatial resolution. The dynamic scheduler will evaluate these and not schedule observations for which too few stations are available. If additional stations become unavailable, they are removed from the observations automatically. If this results in too few stations to be available, the observations will be rescheduled.

The dynamic scheduler can be tuned to favor different circumstances. One could choose for a dense schedule, scheduling as early as possible within the given constraints, or schedule more optimally around transit. This is currently a global setting.

The dynamic scheduler aims to predict the full future schedule. The predicted schedule is interrupted and (partially) recalculated when the information changes, by adding or cancelling scheduling units, changing the constraints or changing the availability of the stations.

The dynamic schedule can be overruled in two ways: firstly, scheduling units can be scheduled at a ”fixed time” for time critical observations, for example those that need to run together with other observatories. These observations take priority over the dynamically scheduled observations. Secondly, for target of opportunity observations, the schedule can be overruled, aborting the current observations if necessary. This can be done through the API and thus can be integrated with user scripts to respond for example to gamma-ray bursts[5] , gravitational-wave events[6] or lightning activity.[7]

The design of the dynamic scheduler allows for adding more constraints in the future, even strongly time-variant constraints. For example the RFI situation of the ionospheric conditions, but also instrument parameters and real-time telescope quality parameters can be added. This will make it a truly dynamic scheduler, as we intended.

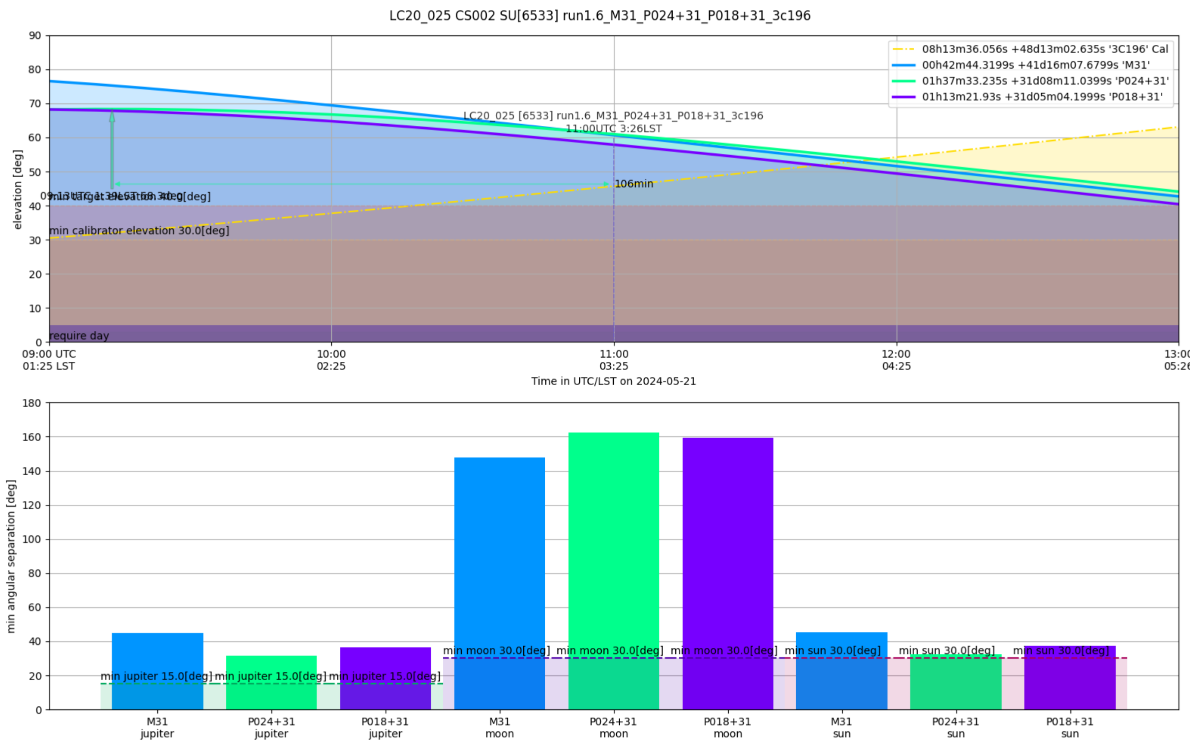


Figure 2. This figure shows the scheduling constraints for a single observation. The values for elevation and distance to celestial objects are shown as well as their minimum requirements.

Several inspection plots are available to visualize the quality of what the scheduler has done. First, a daily scheduling plot is available, showing the elevation vs time of all scheduled targets. This plot is used to check the quality of the predicted schedule of the day. Second, a plot for a single scheduling unit is available, showing all the constraints and their limits (Fig. 2). This can help debug an issue if a scheduling unit cannot be scheduled. Third, a plot indicating schedulability for each constraint for a scheduling unit until the end of the cycle to see which constraints are most limiting. Lastly, there are local sidereal time (LST) pressure plots for the cycle and per project that show in which LST ranges observations are requested and give an overall feasibility to complete the observing program.

The resulting schedule is visualised in a calendar view (Fig. 3). Here the scheduling units are displayed for the different days. Their aggregated status, like scheduled, observing or processing is displayed in color giving an easy overview for the past day and the future predicted schedule. With a click on a scheduling unit, additional information such as the scheduling constraints, are displayed and the user has access to a few actions, like changing those constraints or starting the data quality assessment. Several filters for example on scheduling unit status or project. In addition, several views are available, for example separating the A and B queue, the dynamically versus fixed time scheduled units, displaying the schedule for each station or displaying all the tasks.

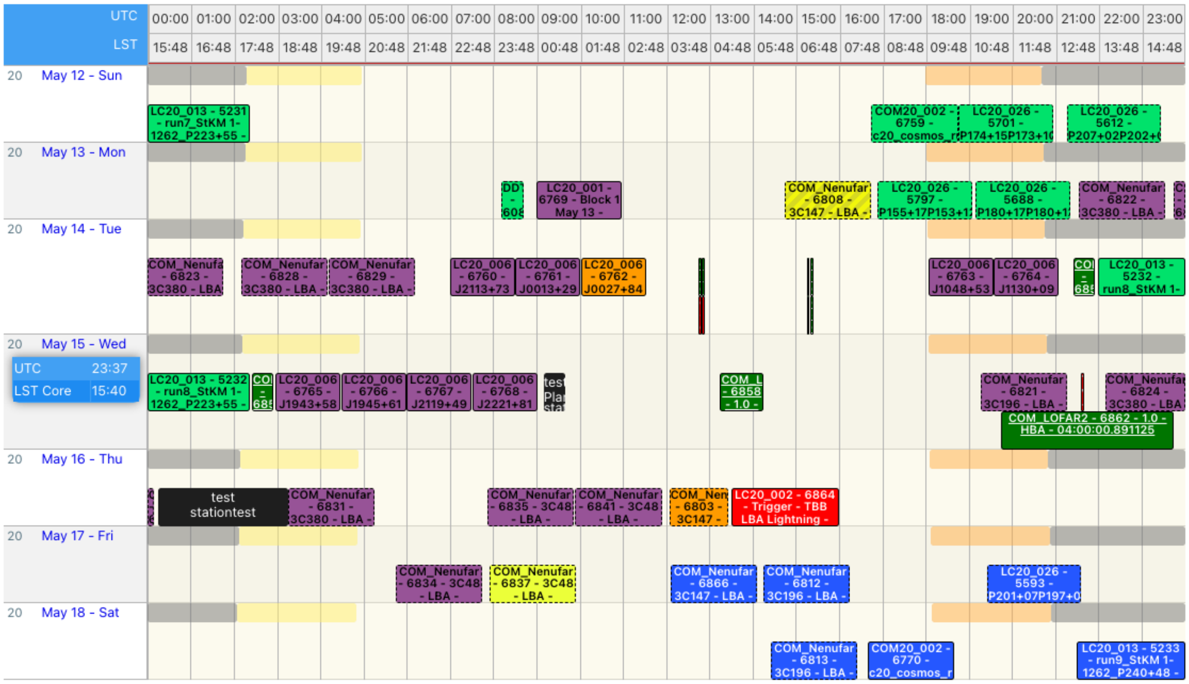


Figure 3. The schedule is shown in a calendar view. This gives a quick overview of the progress of the observations. For example, green observations are fully finished, blue observations are scheduled in the future, the yellow observations are observing or processing and the purple observations have been archived but still need to be removed from disk. This view shows the situation towards the end of the cycle where less observations are still available.

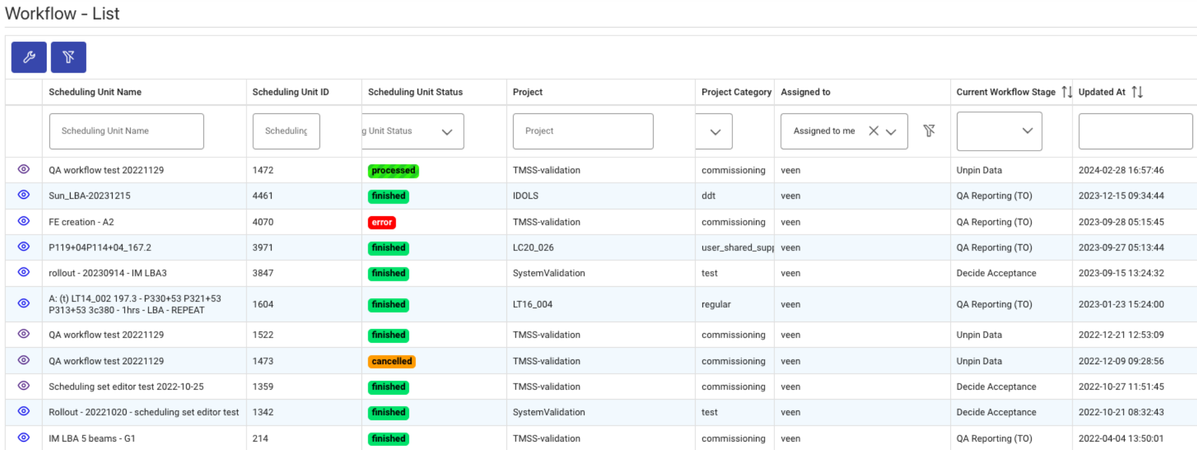
4. DATA QUALITY ASSESSMENT WORKFLOW

Part of the operational process is to do a data quality assessment. If the data quality is below an agreed-upon threshold, the observations are rejected without further discussion. If the data quality is good enough, the PI is informed about any potential issues and is asked to accept the data. With TMSS this process is integrated in a data quality assessment workflow (QA workflow).

The QA workflow includes the following steps: a telescope operations report, written by the operator on duty, a science support amendment of the report, written by the instrument scientist assigned to the project, a PI confirmation step and a final acceptance step. Furthermore there are data management steps that show transfer of data to the long-term archive or removing dataproducts. The latter two steps usually happen automatically, as most observations are accepted and disk space must be cleared as soon as possible. They could however be required actions in the QA workflow by specifying this in the project settings.

Instead of creating a full report from scratch, the operator is presented with a report template in which available data such as data loss and missing stations are already provided. The already logged system issues can also be attached to a report, if they affected the observations, or newly found issues can be logged directly. This report is then assigned to a support scientist for additional insights and finally assigned to the contact author representing the science team for acceptance.

An overview of non-finished workflows is available to the various users (Fig. 4). The operator can filter for any open issues that require their attention. The support scientist and contact author can filter on the issues assigned to them. In this way, they can easily spot which reports require their input.



Workflow - List

Scheduling Unit Name	Scheduling Unit ID	Scheduling Unit Status	Project	Project Category	Assigned to	Current Workflow Stage	Updated At
Scheduling Unit Name	Scheduling	g Unit Status	Project		Assigned to me		
QA workflow test 20221129	1472	processed	TMSS-validation	commissioning	veen	Unpin Data	2024-02-28 16:57:46
Sun_LBA-20231215	4461	finished	IDOLS	ddt	veen	QA Reporting (TO)	2023-12-15 09:34:44
FE creation - A2	4070	error	TMSS-validation	commissioning	veen	QA Reporting (TO)	2023-09-28 05:15:45
P119+04P114+04_167.2	3971	finished	LC20_026	user_shared_sup	veen	QA Reporting (TO)	2023-09-27 05:13:44
rollout - 20230914 - IM LBA3	3847	finished	SystemValidation	test	veen	Decide Acceptance	2023-09-15 13:24:32
A: (t) LT14_002 197.3 - P330+53 P321+53 P313+53 3c380 - 1hrs - LBA - REPEAT	1604	finished	LT16_004	regular	veen	QA Reporting (TO)	2023-01-23 15:24:00
QA workflow test 20221129	1522	finished	TMSS-validation	commissioning	veen	Unpin Data	2022-12-21 12:53:09
QA workflow test 20221129	1473	cancelled	TMSS-validation	commissioning	veen	Unpin Data	2022-12-09 09:28:56
Scheduling set editor test 2022-10-25	1359	finished	TMSS-validation	commissioning	veen	Decide Acceptance	2022-10-27 11:51:45
Rollout - 20221020 - scheduling set editor test	1342	finished	SystemValidation	test	veen	Decide Acceptance	2022-10-21 08:32:43
IM LBA 5 beams - G1	214	finished	TMSS-validation	commissioning	veen	QA Reporting (TO)	2022-04-04 13:50:01

Figure 4. The data quality assessment workflows that require action are presented in a table.

5. ADDITIONAL OPTIMIZATION FEATURES

The features described above are the main improvements that help streamline the LOFAR operations. Here are a few more features and innovations we want to highlight that have improved LOFAR operations.

- A project based *project-based* authentication and authorisation implementation allows expert users to assist with the execution of their projects, reducing the load of operators and science support.
- The API does not only allow fast specification, as highlighted above, but also automation of other interactions, for example to create custom reports, resolve system issues or integration with other applications.
- Using modern web technology allows for user customisation of tables by filtering and selecting which columns to display. In this way each user can optimise for their specific tasks and perform bulk actions by selecting all relevant scheduling units.
- The front-end provides live updates, not only for the scheduling view, but also on the other webpages, allowing users to see directly that their newly approved observation is scheduled.

6. CONCLUSION

Introducing TMSS to LOFAR has helped us streamline operations and move from many small daily tasks for specification, scheduling, archiving, data removal and communication about data quality with the scientists, to an improved operational model in which many tasks are automated or at least readily available. The specification can now be done before the start of the cycle, by using a set of standard observing strategies and specification creation and editing through a spreadsheet editor or using scripts. The scheduling is performed by a dynamic scheduler that incorporates various astronomical and temporal constraints and can automatically react to changes in array availability. The daily administration is integrated and is available through a data quality assessment workflow with an overview for each type of user. A flexible front-end makes it easy to adjust to changes to the capabilities of the telescope and allows for the introduction of new standard observing modes. The API allows automation through scripts or integration with different systems for further streamlining of operations.

The improvements in the system free up man-hours which allows us to move to a different operational model, where we can focus on further post-processing of data to deliver science-ready dataproducts[3] and supporting more complex observing campaigns, like the planned large survey program that will run several science projects in parallel.

Further development will allow even more streamlining by reacting to environmental (RFI, ionospheric) circumstances in the dynamic scheduler and automatic reporting of system problems towards the data quality assessment workflow. With that we look forward to many more science observations and discoveries in radio astronomy.

ACKNOWLEDGMENTS

The authors would like to thank their former team members and contractors for the realisation of TMSS: Roy de Goei, Marco Iacobelli, Tom Kamphuis, Muthu Krishnan, Ramesh Kumar, Mario Raciti, Roberto Pizzo, Nithya Santhanam, Harshith Davanam Subbaramaiah, Fabio Vitello. Contributions of JK were largely funded by BMBF projects D-LOFAR2.0 (grant 05A20PB1) and D-LOFAR-ERIC (grant 05A23PB1).

REFERENCES

[1] van Haarlem, M. P., Wise, M. W., Gunst, A. W., Heald, G., McKean, J. P., Hessels, J. W. T., de Bruyn, A. G., Nijboer, R., Swinbank, J., Fallows, R., Brentjens, M., Nelles, A., Beck, R., Falcke, H., Fender, R., Hörandel, J., Koopmans, L. V. E., Mann, G., Miley, G., Röttgering, H., Stappers, B. W., Wijers, R. A. M. J., Zaroubi, S., van den Akker, M., Alexov, A., Anderson, J., Anderson, K., van Ardenne, A., Arts, M., Asgekar, A., Avruch, I. M., Batejat, F., Bähren, L., Bell, M. E., Bell, M. R., van Bemmel, I., Bennema, P., Bentum, M. J., Bernardi, G., Best, P., Bîrzan, L., Bonafede, A., Boonstra, A. J., Braun, R., Bregman, J., Breitling, F., van de Brink, R. H., Broderick, J., Broekema, P. C., Brouw, W. N., Brüggen, M., Butcher, H. R., van Cappellen, W., Ciardi, B., Coenen, T., Conway, J., Coolen, A., Corstanje, A., Damstra, S., Davies, O., Deller, A. T., Dettmar, R. J., van Diepen, G., Dijkstra, K., Donker, P., Doorduin, A., Dromer, J., Drost, M., van Duin, A., Eislöffel, J., van Enst, J., Ferrari, C., Frieswijk, W., Gankema, H., Garrett, M. A., de Gasperin, F., Gerbers, M., de Geus, E., Grießmeier, J. M., Grit, T., Gruppen, P., Hamaker, J. P., Hassall, T., Hoeft, M., Holties, H. A., Horneffer, A., van der Horst, A., van Houwelingen, A., Huijgen, A., Iacobelli, M., Intema, H., Jackson, N., Jelic, V., de Jong, A., Juette, E., Kant, D., Karastergiou, A., Koers, A., Kollen, H., Kondratiev, V. I., Kooistra, E., Koopman, Y., Koster, A., Kuniyoshi, M., Kramer, M., Kuper, G., Lambropoulos, P., Law, C., van Leeuwen, J., Lemaitre, J., Loose, M., Maat, P., Macario, G., Markoff, S., Masters, J., McFadden, R. A., McKay-Bukowski, D., Meijering, H., Meulman, H., Mevius, M., Middelberg, E., Millenaar, R., Miller-Jones, J. C. A., Mohan, R. N., Mol, J. D., Morawietz, J., Morganti, R., Mulcahy, D. D., Mulder, E., Munk, H., Nieuwenhuis, L., van Nieuwpoort, R., Noordam, J. E., Norden, M., Noutsos, A., Offringa, A. R., Olofsson, H., Omar, A., Orrú, E., Overeem, R., Paas, H., Pandey-Pommier, M., Pandey, V. N., Pizzo, R., Polatidis, A., Rafferty, D., Rawlings, S., Reich, W., de Reijer, J. P., Reitsma, J., Renting, G. A., Riemers, P., Rol, E., Romein, J. W., Roosjen, J., Ruiter, M., Scaife, A., van der Schaaf, K., Scheers, B., Schellart, P., Schoenmakers, A., Schoonderbeek, G., Serylak, M., Shulevski, A., Sluman, J., Smirnov, O., Sobey, C., Spreeuw, H., Steinmetz, M., Sterks, C. G. M., Stiepel, H. J., Stuurwold, K., Tagger, M., Tang, Y., Tasse, C., Thomas, I., Thoudam, S., Toribio, M. C., van der Tol, B., Usov, O., van Veelen, M., van der Veen, A. J., ter Veen, S., Verbiest, J. P. W., Vermeulen, R., Vermaas, N., Vocks, C., Vogt, C., de Vos, M., van der Wal, E., van Weeren, R., Weggemans, H., Weltevrede, P., White, S., Wijnholds, S. J., Wilhelmsson, T., Wucknitz, O., Yatawatta, S., Zarka, P., Zensus, A., and van Zwieten, J., "LOFAR: The LOw-Frequency ARray," **556**, A2 (Aug. 2013).

[2] B. Hut, C. J. Baldovin, C. B. W. v. C. A. W. G. M. R. A. P. S., "The lofar2.0 upgrade: validation results of the qualification model," in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Serie*], *Proc. SPIE* **13096** (2024).

[3] E. Orrú, M.J. Norden, M. I. S. t. V. A. A., "Lofar vs lofar2.0 operations: new challenges," in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Serie*], *Proc. SPIE* **13098** (2024).

[4] Lucero, S. and Farris, A., "Dynamic Scheduling in ALMA," in [*Astronomical Data Analysis Software and Systems XVI*], Shaw, R. A., Hill, F., and Bell, D. J., eds., *Astronomical Society of the Pacific Conference Series* **376**, 673 (Oct. 2007).

[5] Broderick, J. W., Shimwell, T. W., Gourdji, K., Rowlinson, A., Nissanke, S., Hotokezaka, K., Jonker, P. G., Tasse, C., Hardcastle, M. J., Oonk, J. B. R., Fender, R. P., Wijers, R. A. M. J., Shulevski, A., Stewart, A. J., ter Veen, S., Moss, V. A., van der Wiel, M. H. D., Nichols, D. A., Piette, A., Bell, M. E., Carbone, D., Corbel, S., Eislöffel, J., Grießmeier, J. M., Keane, E. F., Law, C. J., Muñoz-Darias, T., Pietka, M., Serylak, M., van der Horst, A. J., van Leeuwen, J., Wijnands, R., Zarka, P., Anderson, J. M., Bentum,

M. J., Blaauw, R., Brouw, W. N., Brüggen, M., Ciardi, B., de Vos, M., Duscha, S., Fallows, R. A., Franzen, T. M. O., Garrett, M. A., Gunst, A. W., Hoeft, M., Hörandel, J. R., Iacobelli, M., Jütte, E., Koopmans, L. V. E., Krankowski, A., Maat, P., Mann, G., Mulder, H., Nelles, A., Paas, H., Pandey-Pommier, M., Pekal, R., Reich, W., Röttgering, H. J. A., Schwarz, D. J., Smirnov, O., Soida, M., Toribio, M. C., van Haarlem, M. P., van Weeren, R. J., Vocks, C., Wucknitz, O., and Zucca, P., "LOFAR 144-MHz follow-up observations of GW170817," **494**, 5110–5117 (June 2020).

[6] Rowlinson, A., de Ruiter, I., Starling, R. L. C., Rajwade, K. M., Hennessy, A., Wijers, R. A. M. J., Anderson, G. E., Mevius, M., Ruhe, D., Gourdji, K., van der Horst, A. J., ter Veen, S., and Wiersema, K., "A coherent radio flash following a neutron star merger," *arXiv e-prints* , arXiv:2312.04237 (Dec. 2023).

[7] Hare, B. M., Scholten, O., Buitink, S., Dwyer, J. R., Liu, N., Sterpka, C., and ter Veen, S., "Characteristics of recoil leaders as observed by LOFAR," **107**, 023025 (Jan. 2023).