ROYAL SOCIETY OF CHEMISTRY

PERSPECTIVE

Check for updates

Integrating autonomy into automated research platforms

Richard B. Canty, [ID] Brent A. Koscher, [ID] Matthew A. McDonald [ID] and Klavs F. Jensen [ID] *

Integrating automation and autonomy into self-driving laboratories promises more efficient and reproducible experimentation while freeing scientists to focus on intellectual challenges. In the rapid advances being made towards self-driving laboratories, automation and autonomy techniques are often convoluted due to similarities between them and ambiguous language, leaving the trade-offs between them overlooked. In this perspective, we address differences between making a process occur without human intervention (automation) and providing agency and flexibility in action (autonomy). We describe the challenges of autonomy in terms of (1) orchestration, how tasks are organized and coordinated; (2) facilitation, how devices are connected and brought under automated control; and (3) scripting languages, how workflows are encoded into digital representations. Autonomous systems require advanced control architectures to handle a reactive, evolving workflow, involving control abstractions and scheduling beyond what current automation approaches provide. The specification of an autonomous system requires goal-oriented commands and context awareness, whereas automation needs exact, unambiguous instructions for reproducibility and efficiency. We contend that this contrast in design creates a need for improved standards in automation and a set of guiding principles to facilitate the development of autonomy-enabling technologies.

Introduction

As machine learning and laboratory automation have advanced, interest in the creation of self-driving laboratories has increased and generated proof-of-concept platforms.[1–11] These platforms aim to improve experimental precision, accuracy, throughput, and reproducibility, accelerating the acquisition of scientific knowledge and freeing up the effort of scientists to focus on deeper theoretical questions.[12,13]

Previous discussions concerning automation and autonomy have included: (a) a theoretical basis for comparing the degree of cognitive automation in chemical design[14] and in holistic experimental design and interpretation,[15] (b) prior demonstrations of automated research in the chemical, materials, and life sciences,[16–20] (c) the interplay between the components of automation and improved artificial intelligence,[5,21–31] (d) the practical considerations of developing automated platforms,[32,33] and (e) the associated data-management of automated workflows.[34–36] In this perspective, we address the requirements and consequences of designing for autonomy in contrast to automation and discuss the challenges with its implementation and the community adoption of autonomy-enabling tools. In the discussion of design, control, and encoding of autonomy,

we include our approach[2] as an example of designing autonomy-supporting tools for a self-driving experimental platform, comprising multiple autonomous agents.

In prior publications on autonomy, the definition of autonomy has been flexible: automation coupled with machine learning for cognitive processes,[12–14,37] the extent to which the automation's research is indistinguishable from that of a human researcher,[15] a paradigm where automation is goal-oriented,[32] the extent to which a system can achieve a set of goals in a given context,[38] or as a synonym of automation.[34] To avoid ambiguity, we define automation herein as the act of making a process occur without human intervention and autonomy as a paradigm where feedback and adaptive decision-making afford the system agency over the manner of its actions. This self-determinism is often reflected in a change from instruction-oriented to goal-oriented automation—two design patterns that have different needs and have similar, but distinct, effects on self-driving laboratories (Table 1).†

The design of an automatic and autonomous system starts with the goals of the system and how these are codified. These objectives cascade into requirements and consequences for facilitation and orchestration then into scripting (Table 1). For

Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, USA. E-mail: kfjensen@mit.edu

† Similar to computer programming, it is possible to achieve any functionality using any design paradigm; different designs exist to make achieving desirable functionality easier and more extensible.

Table 1 High-level comparison of the design needs and ramifications of automation and autonomy in the development of experimental platforms

		Automation	Autonomy
Design. What is required and what is the objective?	Goal	Robust process Exact reproducibility	System actively learns robustness Adaptive operation (agency)
	Consequence	Process facilitation and orchestration requires input/output validation	Process facilitation and orchestration must support planning, analysis, and learning tasks
		Orchestrator manages concrete tasks	Orchestration needs to handle both abstract and concrete instructions
		Scripts require specific imperatives Designer needs to explicitly account for every (reasonable) eventuality	Designer needs to provide a means for control architecture to evolve
Facilitation & orchestration. How are interfaces designed? What controls scheduling and coordination?	Goal	Expedite platform setup Improve software and hardware maintainability Manage material and information throughput Maintain platform operation	
	Consequence	Standardized interfaces for control	Standardized interfaces for control and feedback
		Unidirectional/Hierarchical control-flow	Bidirectional/Collaborative control-flow
		Records of performance (logs) may be pass/fail (process is already fully specified)	Process control (feed-back/-forward) Precise logs must be generated during operation
Scripting. How are processes codified and to what end?	Goal	Full and exact specification of process	High-level specification of process Expressive power (generalizability)
	Consequence	Scripts concern tasks and workflows (well-defined)	Scripts concern goals and objectives (vague)
		Porting scripts between systems is pass/fail (the recipient system either has the capability to run the script or not)	Porting scripts between systems is adaptive, but potentially lossy (the recipient system may modify the workflow)

automation, the goals of robustness and precision result in a direct control architecture and rigorous instructions, which in turn yield scalable, portable, and transparent workflows. Conversely, for autonomy, the goals of adaptability and expressive power result in a highly context-aware control architecture and underspecified but powerful instructions, which in turn yield robust workflows with few discarded experiments. In this sense, the design pattern of autonomy trades efficiency in in-domain automation for the flexibility to make out-of-domain processes possible. Notably, autonomy is not the pinnacle of automation: an autonomous platform can have human agents, and a fully automatized, perfectly specified process leaves no degrees of freedom for there to be autonomy.[38]

Self-driving laboratory example

Our foray into autonomous experimentation is a self-driving, automated experimental platform to discover small, dye-like molecules.[2] The platform iteratively predicts molecules given a set of desired properties (UV/Vis spectra, water–octanol partitioning, and photodegradation rate), plans syntheses, organizes reactions into 96-well plates, performs syntheses and workup, isolates and characterizes products, and uses this information to update its predictors. The platform comprises a controller, four computer agents, and databases for experimental design and experimental results. Each of the four agents is provided access to a collection of laboratory equipment: (a) a liquid handler, (b) a robotic arm, (c) an HPLC-MS and fraction collector, and (d) a storage carousel, plate-reader, a bespoke high-temperature reactor for well plates, and a solar-simulating light source. The platform was designed to be generalist, but with concessions to air-sensitive reactions, photo- and electro-chemistries, and temperatures outside 5–200 °C. The challenges of machine-learning-driven, multi-step synthesis necessitated the inclusion of autonomy to handle unknowns in the workflow (e.g., new reactions or conditions proposed by the machine learning software). The primary injections of autonomy into the platform are: (a) the ability of the agents to modify the workflow, which comprises the order, duration, and identity of all operations required for the experiments (or batches thereof), and (b) high-level directives from the controller that the agents translate into hardware-level commands based on platform and experimental contexts found in the databases.

Designing for autonomy

Unfortunately, there is no universal solution for adding autonomy to an automated experimental platform. On the task level, process control is a form of autonomy: disturbances from the setpoint are corrected *via* adjustments to input parameters such as temperature or flow rate.[39] Self-configuration[10,40] and automated error recovery[2] are forms of autonomy on the platform and workflow levels, respectively. Autonomous platforms rectify incompatibilities with the experimental design rather than discard experiments. Such rectification is necessary to deal with unexplored or poorly understood chemical or material spaces. Since predicting high-level process goals is easier than predicting specific instructions, there has been a continued interest in the abstraction of chemical processes.[2,32,41] This abstraction trades the (exact) reproducibility and specificity of instruction-oriented automation strategies for improved flexibility and fidelity (*i.e.*, the successful execution of an experiment, regardless of the experiment's outcome).

Our platform accounts for the uncertainties that arise from model-driven chemical synthesis, by utilizing workflow-level adaptivity and task-level operational insights. To this end, each agent is treated as a "robotic expert"[42]—afforded the ability to determine how best to accomplish its goals and, if it cannot, the ability to modify the workflow itself by changing or adding tasks to rectify obstacles. This provides a means for both automatic error recovery and reactive processing.

Rectifications make the platform more robust in executing workflows. Similarly, the ability to expand the workflow (*e.g.*, adding isolation steps for each successful reaction in a batch) allows for actions to be planned which could not be known at the inception of the workflow. While the mutability of the workflow greatly increases the autonomy of the platform, it comes at the cost of transparency as a completed workflow may be substantially different from one proposed by the experimental designer.

We suggest that robotic experts should be decoupled from their tools. This would allow a robotic expert to inspect all functionality available to it, including equipment shared between agents. This expands capability and reduces the chance of breaking workflows when modules are modified. Such separation of the decision logic from the agent also decouples the sharable aspect of the robotic expert (its decision logic) from the hardware- and software-specific implementation details, and allows the expert to be portable to other systems with minimal overhead.

While the first generations of (semi)autonomous platforms have used hard-coded logic, with standardized data capture goal-oriented platforms may learn from past experiments and self-optimize experimental protocols. For such fully autonomous agents to proliferate, their training data would need to be accessible to others. This will require publishing experimental logs or precise summaries of decision logic such that others do not have to train their robotic experts from scratch.

To achieve active learning in these robotic experts, data capture and formatting for autonomous systems need richer and more structured information than existing paradigms like FAIR[43] alone can manage.[36,41] To ensure adequate logging for scientific rigor, standardized loggers will need to capture information on the sample, hardware, and platform levels. Fortunately, there exist bases for (meta)data capture and formatting in the ORD,[44] CRIPT,[45] HELAO,[41] ESCALATE,[34] and ESAMP[46] approaches which can augment FAIR logging.

Challenges with such log mining include the question of how to capture complex interactions (*e.g.*, the previous workflow left a residue on a pipette) and how to share these observations with other platforms as these observations are highly contextual and can become too large to be sensibly utilized. Existing commercial automation hardware does not possess the level of logging detail required, often requiring supplementation with cameras or other sensors[3,8,11,39,47]—forestalling the development of such a standard until auditing features are sufficient.

Ultimately, the degree of autonomy appropriate for a system depends on its purpose. Autonomy comes at the cost of increased management complexity, more difficult scalability, and decreased raw throughput, but provides improved experimental fidelity and the ability to explore unfamiliar chemical spaces.

Orchestration and facilitation

Facilitator software and middleware, which ease the integration of hardware and software systems into a coherent, programmatically controllable platform, are crucial for the rapid development of self-driving laboratories. Orchestration software, which manages coordination, signaling, and the flow of information between systems (often including the user and any data repositories), is critical for the robust control of automated platforms with multiple instruments. Facilitation and orchestration software can be purpose-built and are often packaged together (as a controller) to align with specific applications.

The creation of a universal controller (applicable to any workflow, in any domain, on any platform) is challenged by the trade-off between structure (ease) and freedom (power) and how the nature of the application limits adaptability. Even within a single domain, the scalability of a controller to more intricate workflows or larger experimental platforms can be a challenge. Complexity and additional control layers may worsen latency, risk leaks in modularity (making code harder to modify for new applications), and hinder data capture and organization.[34,41] The structure that makes a flow-chemistry controller easy to use may preclude adaptations to batch-chemistry applications, and *vice versa*; and providing the power to handle both risks being too convoluted to be generally useful. Second, reproducible research requires rigorous data management (material provenance, detailed records of measurables and observations, transparent data processing, *etc.*); however, modifying how controllers handle data can often break other controller functionality. A final challenge to controller universality is human interaction and its variability. Humans, as administrators or agents, require user-interfaces and protocols to ensure experiments are not corrupted.

There are a few facilitators which were designed to be adopted by other laboratories, notably ChemOS[48,49] (commercially as Atinary), Hierarchical Experimental Laboratory Automation and Orchestration[7,41] (HELAO), Experiment Specification, Capture and Laboratory Automation Technology[34] (ESCALATE), Materials Acceleration Operation System[50] (MAOS) (and its in-cloud version MAOSIC[51]), LeyLab,[8] BlueSky + Ophyd,[52] and Autonomous Research System Operating System[4] (ARES OS). These facilitators reduce coding burdens, as they automate many of the integration steps, and can tutor the development of automated platforms through the analysis of their scaffolding; however, none are universal, and the nature of an application may require creating a new controller in-house.

Our application required the independent (and potentially concurrent) execution of workflows comprising batch-synthesis, isolation, and characterization experiments grouped into 96-well plates. Moreover, networked control was required as a single computer was insufficient to handle all the equipment. After reviewing available controllers at the time, we decided to design our controller in-house.

Traditional unidirectional control structures (Fig. 1A and B) do not lend themselves to agents which may need to construct or adapt their own methods on the fly. Our collaborative architecture uses minimal signals, and the bulk of information transfer is relegated to database queries (Fig. 1C). Commands specify only enough information to look-up and validate information from the design database (Fig. 1C, minimal signal). Responses specify enough information to guide how the orchestrator should proceed—if the operation succeeded, the system was already busy, there was a recoverable error, or there was an error requiring human intervention—along with some logging details (Fig. 1C, response).

These minimal, standardized signals helped keep the controller flexible and expandable despite the poor standardizations of low-level labware commands and application programming interfaces (APIs). The lack of commercial automation standards meant that instrument-specific details for each operation on the platform had to be manually coded. These could be organized and contained within each agent, since the minimal signal structure meant that changes to agents would not affect the controller.

This use of multiple independent agents is shared between many controllers—owing to how modularity eases the modification and implementation of new capabilities and protects networks of systems from cascading crashes (*e.g.*, one agent going offline does not crash other agents or halt the controller). Paradigms that improve containment will likely guide the design of future controllers. Importantly, it makes substituting systems easier; for example, simulated and user-mediated operations are invaluable for development, maintenance, and minimizing downtime. While it would seem antithetical to implement a mode where a human can seamlessly perform tasks on an autonomous platform, an autonomous platform need not be completely automated—sometimes a task is more efficiently performed by a human.

While the interfaces of goal-oriented agents and their networks are crucial for both automated and autonomous systems, the adaptability of autonomous systems imposes a unique burden on orchestration, particularly on scheduling. When a system can modify its own workflow in response to feedback, which may be delayed by multiple steps (*e.g.*, a reaction's success or failure is not known until after analysis), it risks scheduling conflicts and resource mismanagement. Absent parallelized, autonomous agents, previous schedulers could rely on (prioritized) first-come first-served algorithms[3,41,49,51] and ready-checks.[37,53]

To overcome the challenges of nondeterministic workflows, our orchestrator supports the dynamic allocation of resources, performs safety checks for agent operation and potential congestion of resource traffic on the platform, and a scheduling algorithm that has both a planning horizon extending beyond one task and handles temporal constraints between tasks.
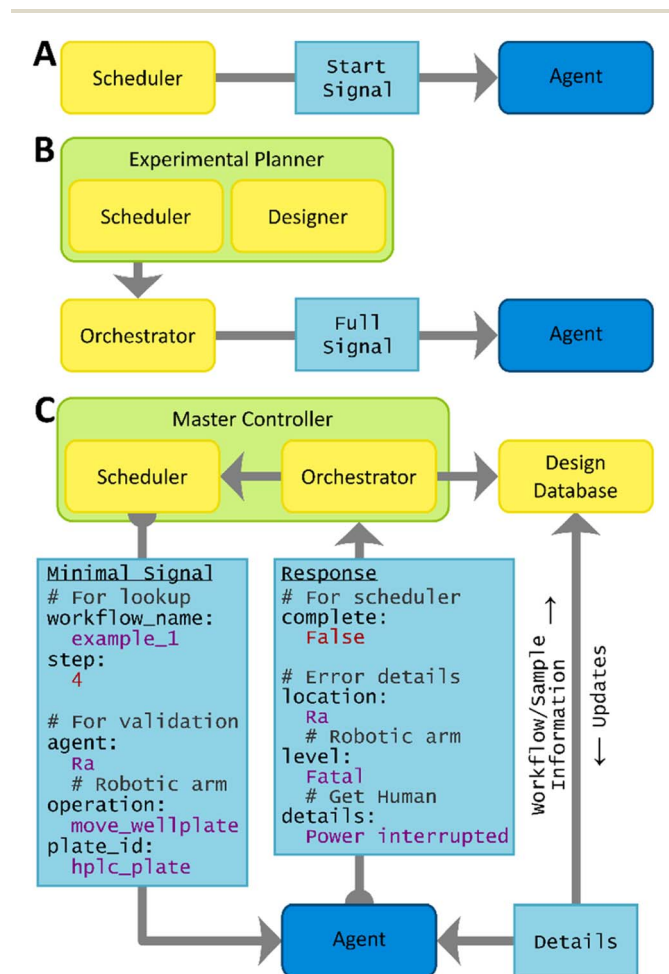

Fig. 1 Visualization of the offloading of information from the intersystem messages to requests made to the database. (A) Physically automated approach in which agents with preloaded methods are triggered by a scheduler. (B) Recent approaches in which experiments with preloaded methods are selected by an algorithm and orchestrated in parallel with triggers or fully specified signals. (C) Collaborative approach in which algorithmically selected experiments with preloaded workflows are orchestrated by live scheduling and specified by agents from database queries. Examples of the minimal signals and responses used in this work are shown.
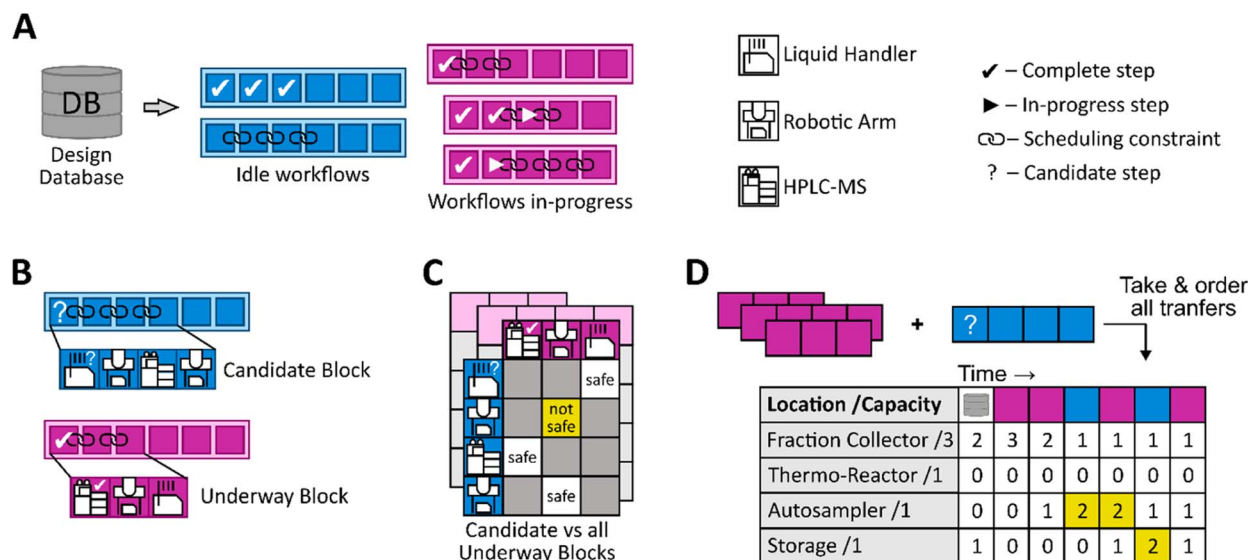
Fig. 2   Overview of the scheduling and conflict management algorithm. (A) Preparation of workflow information from the database. (B) Grouping of time-constrained operations into blocks. (C) Comparison of the candidate scheduling block and in-progress scheduling blocks to ensure there is no agent-time conflict. (D) Simulation (initialized from the database) of resource positions on the platform to detect potential gridlocks. Conflicts in (C) and (D) are marked in yellow.

When planning multiple, non-identical workflows in parallel, the scheduler does not perform scheduling optimization as workflows frequently change and the appropriate optimization algorithms require significant computational effort. We generate a satisfactory schedule with an algorithm that selects the best option at each decision point which does not risk a future scheduling conflict (Fig. 2).

Orchestration is constrained by the inherent ordering of the workflow and the capabilities of each instrument requested. In addition, we imposed that partially completed tasks cannot be suspended to run another task and that tasks may contain scheduling constraints. Supported temporal constraints include (a) no relation, (b) a defined wait period between two tasks, and (c) a minimum wait period between two tasks. For constraints requiring a maximum allowable time between tasks, the window of opportunity was assumed to be zero seconds. Such windows are often contextual (e.g., before significant evaporation or hydration of solvent) and would require sensors not currently present on our platform. Each task's specification requires a record of start and end times as well as time estimates that can be adjusted by agents during execution to provide the scheduler with up-to-date information. This temporal information is then used to detect timeout errors and conflicts when scheduling new tasks.

Scheduling is accomplished by building and filtering a pool of candidate workflows. The design database is pulled for all complete, underway, and idle workflows (Fig. 2A). Idle workflows are filtered based on the completion of prerequisites, and the first incomplete task is identified. The last complete and first incomplete operations are inspected for scheduling constraints. This inspection cascades forward, and any group of constrained tasks is considered as one block (Fig. 2B). Candidate blocks are then screened to ensure all agents are operational. Each candidate block is evaluated against underway task blocks using a conflict matrix (Fig. 2C), and intersections of both agent and expected operational time are identified. The traffic of items on the platform is simulated with the candidate block to check for gridlocks with the existing, underway blocks (Fig. 2D). The remaining candidates are scored on the age of their workflow and their time overdue for a scheduled operation. There is a penalty for workflows that report recoverable errors without initiating a recovery protocol or have a status discrepancy, such as a human using a system without proper check-out. The highest-ranked task is selected for execution.

In reflecting on this scheduler, it became clear that the inclusion of sample monitoring would be advantageous for an autonomous platform. Given a means to oversee samples not undergoing an operation (such as tracking evaporation or exposure), an autonomous system could adjust scheduling constraints or insert operations to better preserve/restore samples[32] (also known as "parking" samples[19]). Moreover, the requisite sensors and models could provide a way to afford meaningful estimates for maximum allowable times between operations.

The design of autonomy-supporting facilitation and orchestration software remains a challenge. Interfaces need to transmit sufficient and properly formatted streams of information for adaptive control at the task and workflow levels, and orchestrators must be accommodating to flexibility in the workflows. With the differences in workflows between fields and applications, it becomes difficult to select a single architecture that can meet all potential application needs while also minimizing overhead and remaining accessible to the scientists using it.

As developments in chemistry automation integrate the chemical, material, robotic, and software sciences, laboratories could collaborate with skilled software engineers to ensure the necessary support is present for developing these projects. This would address a critical gap in current self-driving laboratory software: the depth of software development skill required to make such automata functional and accessible is daunting to those who would benefit most from its general use. Such in-house expertise would balance the power of autonomous chemistry software and establish its accessibility to future researchers. Eventually, it is possible that commercial or community projects can be shared to increase the accessibility of software tools without the need for software experts for deployment or further development. Laboratories could then either use a software package wholesale or build their own system piece-wise from standardized modules to accommodate new research domains.

Encoding workflows

Workflows that support branching or procedural generation and comprise diverse, concurrent operations, can improve robustness and the overall capabilities of a platform. Increased dynamism and inter-workflow heterogeneity, however, add the burden of allocating resources, coordinating materials and data, scheduling/parallelization complexity, and unambiguously representing the workflow in a digital format. The nature and capabilities of this digital encoding determine the complexity of platform-executable workflows. Three notable chemistry process scripting languages that aim to be both generalizable and maintained for community use are χDL,[37,54–57] CRIPT,[45] and Autoprotocol.[58,59]

χDL and Autoprotocol are human- and machine-readable prescriptive languages that aim to be hardware agnostic and computationally unambiguous for reproduction between laboratories. Languages may make assumptions about the basic conceptual unit of the platform (e.g., an experiment, a reaction template, a batch, a laboratory vessel, etc.). The scripting language employed may affect the adaptability of workflows. Autoprotocol only supports fixed workflows (lacking runtime evaluations and branching logic). χDL does allow monitoring steps and loops with dynamic feedback control, but not branching workflows. Another challenge of prescriptive scripting languages is the disconnect between instructed and realized actions and quantities. More descriptive languages, such as CRIPT, can capture the full history of reagents, materials, and processes—though often at the cost of facile human interpretability.

Any scripting language will require some level of compilation to be translated into instructions for a physical platform. More concise languages, such as χDL, require more interpolation to fill in gaps whereas highly descriptive languages, such as CRIPT, require more interpretation to discern which details are relevant. While individual versions of each language may be modified to meet the goals of a single project, a consistent community standard improves the transferability of scripts between laboratories and can accelerate collaborative projects.

Scripting languages highlight the difference between automatic and autonomous design. An automatic workflow requires fully specified actions that can be reproduced without interpolation or interpretations, whereas an autonomous workflow is conducted entirely through interpolation and interpretation of goals and contexts. Languages that attempt to handle both risk being too vague for automation and too restrictive for autonomy.

Two fundamental challenges with scripting languages for autonomy are (a) accounting for new operations and adaptations required by an ever advancing scientific field and by evolving autonomous agents and (b) balancing abstraction with reproducibility. The inclusion of a means for incorporating new base directives‡ in a scripting language inhibits the shareability and reproducibility of a workflow as implementation details become required to translate or compile the script. Similarly, the adaptability required by autonomous workflows contradicts a community goal for automation whereby scripting languages act as a standard to help laboratories develop, share, and reproduce experiments as any adaptations are context-dependent and may be governed by stochastic or unpredictable events (e.g., network connectivity, nucleation events, etc.).

We required a language for batch chemistry in well plates that was dynamic and could be interpreted by a human in both its planned and ultimate form. We encoded workflows and their metadata in dictionaries and hosted them in a local database. These digitized workflows were constructed by extending and filling templates based on the required steps and reaction conditions for batches of multistep syntheses, organized by well plate. The data representation of the workflow serves to support adaptability, and human readability, through three primary choices:

Firstly, steps in the workflow document are high-level, well-plate-oriented commands, such as "prepare_wellplate" or "hplc_semiprep". While the abstraction of multiple subtasks into a single task is utilized by most orchestrators, these are either fixed abstractions[34,41] or are defined at compile-time.[54] By separating the directive (workflow specifications) from the details (databases), the agent, acting as the robotic expert, can fill in and implement these high-level commands using live data.

Secondly, the well plates are referenced by a semantic alias in the workflow (e.g., "reaction_plate" or "filtrate_plate"). This improves human readability and allows any compatible, available well plate to be used and linked to the given alias at runtime—a useful feature when managing resources between multiple workflows in parallel.

Finally, the workflow itself is mutable, allowing agents to modify the workflow, typically by inserting new steps, and convey information to other agents, typically by adding new wellplates or updating step or well detail fields. The actual actions performed on each well or well plate (as applicable) are logged or can be recovered from either the workflow document or the scripts generated for hardware execution.

---

‡ This is distinct from the composition of existing base directives into a higher-order directive, often called a macro or method, which exists as a means of organizing and simplifying code.

Table 2 Desiderata of APIs for chemical automation and autonomy, in decreasing order of priority

Control	Reporting
(1) Publicly accessible and thorough documentation	
(2) Safe changes of state	(2) Confirmation of receipt responses
(3) Method generation & validation, as applicable	(3) Report current system state
(4) Validate current system state	(4) Non-proprietary data export
(5) Perform any operation accessible *via* the user interface, as applicable	(5) Rich, meaningful operational responses
(6) Unrestricted changes of state	(6) Logging and data-capture integration

Standardization

Autonomy benefits from standards in automation; however, autonomy is itself challenging to standardize as it is characterized by its deviation from established protocols.

Concerning automation, the improvement of APIs, both vendor- and user-made, would facilitate the development of automated and autonomous platforms—a list of needs for control and reporting is provided in Table 2. When published, APIs should have minimal technical debt.[60,61] Exposing low-level functionality allows platforms the power to accomplish tasks beyond the well-structured methods often exposed to users for ease. Moreover, rich reporting allows more robust, responsive operation and provides the details required for a platform to learn. The construction of a public repository of these APIs, where all submissions are cleaned of stylistic and programmatic errors, and versioned (both for their own features and for what they control), would greatly facilitate laboratory automation. However, this may represent an opportunity cost and a potential threat to the intellectual property of industry. It is also difficult to justify the development cost of an industry standard when it would limit market opportunities to license proprietary automation software packages/expansions. A government program or consortium of academic and industrial laboratories collaboratively pushing for the realization of chemical and material automation software and hardware standards may be best equipped to realize this goal.

Any singular automation research laboratory is ill-suited to drive standards in automation as its role is to push boundaries and develop new technologies. Existing frameworks, without considerable modification, may not meet the needs of novel automated chemistry platforms. Individual laboratories are likely to create their own code and autonomy frameworks (intra-laboratory standards), applicable to the focus areas of the laboratory, to facilitate expansion and cross-generational use. However, collective action for the standardization of automation between laboratories would make it easier to expand self-driving laboratories into new research areas in chemical and materials sciences, enabling new discoveries. In the meantime, sharing intra-laboratory standards can help accelerate standardization. By providing publicly accessible and thorough documentation, other groups can better learn from, deploy, and adapt existing technologies.

While there are many automated chemistry systems, spanning discovery to optimization, platforms capable of autonomous experimentation are much less common. Few of the latter exhibit much beyond process-level autonomy. Despite their scarcity, each demonstrates different architectures for autonomous research. This journey toward autonomy in chemistry is still in its infancy. It may be best, for the time being, to embrace the diversity of ideas and architectures presented in existing and future automated systems before attempting to establish standards for autonomy. Considering the field of computation, a field to which chemical automation has been routinely compared,[61] despite its age and maturity, there exists a thriving ecosystem of coding paradigms, languages (*e.g.*, Python, C++, Rust, MATLAB), operating systems (*e.g.*, Windows, macOS, Unix), and architectures. If chemistry automation is similar, then it is likely that multiple application standards will arise, each suited for a general use-case or experimental architecture, as will a more general set of guidelines or principles for automation and autonomy which are multidisciplinary.

Conclusions

Great strides are being made to automate chemistry to the point that entire experimental workflows can be executed under programmatic, machine-learning-driven control with little to no human intervention. The development of automation- and autonomy-supporting tools, however, have historically been convoluted due to the ambiguity between them. The self-determinism required by autonomy lends itself to flexible, goal-oriented chemical programming which in turn is poised to accelerate research in novel domains; however, further work is needed to develop this paradigm as its flexibility presents a design challenge with a broad solution space.

Data availability

As this is a Perspective article, no primary research results, data, software or code have been included.

Author contributions

RBC drafted the manuscript. All authors edited the manuscript and approved the final version. The system described in the manuscript was conceptualized by all authors, and RBC, BAK, and MAM built and operated the system.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

Notes and references

1 Q. Zhu, F. Zhang, Y. Huang, H. Xiao, L. Zhao, X. Zhang, T. Song, X. Tang, X. Li, G. He, B. Chong, J. Zhou, Y. Zhang, B. Zhang, J. Cao, M. Luo, S. Wang, G. Ye, W. Zhang, X. Chen, S. Cong, D. Zhou, H. Li, J. Li, G. Zou, W. Shang, J. Jiang and Y. Luo, An All-Round AI-Chemist with a Scientific Mind, *Natl. Sci. Rev.*, 2022, **9**(10), nwac190, DOI: **10.1093/nsr/nwac190**.

2 B. Koscher, R. B. Canty, M. A. McDonald, K. P. Greenman, C. J. McGill, C. L. Bilodeau, W. Jin, H. Wu, F. H. Vermeire, B. Jin, T. Hart, T. Kulesza, S.-C. Li, T. S. Jaakkola, R. Barzilay, R. Gómez-Bombarelli, W. H. Green, K. F. Jensen Autonomous, Multi-Property-Driven Molecular Discovery: From Predictions to Measurements and Back. *ChemRxiv*, 2023, preprint, DOI: **10.26434/chemrxiv-2023-r7b01**.

3 H. Fakhruldeen, G. Pizzuto, J. Glowacki and A. I. Cooper, ARChemist: Autonomous Robotic Chemistry System Architecture, in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6013–6019, DOI: **10.1109/ICRA46639.2022.9811996**.

4 P. Nikolaev, D. Hooper, F. Webber, R. Rao, K. Decker, M. Krein, J. Poleski, R. Barto and B. Maruyama, Autonomy in Materials Research: A Case Study in Carbon Nanotube Growth, *npj Comput Mater*, 2016, **2**(1), 1–6, DOI: **10.1038/npjcompumats.2016.31**.

5 M. Christensen, L. P. E. Yunker, F. Adedeji, F. Häse, L. M. Roch, T. Gensch, G. dos Passos Gomes, T. Zepel, M. S. Sigman, A. Aspuru-Guzik and J. E. Hein, Data-Science Driven Autonomous Process Optimization, *Commun. Chem.*, 2021, **4**(1), 1–12, DOI: **10.1038/s42004-021-00550-x**.

6 V. Shekar, G. Nicholas, M. A. Najeeb, M. Zeile, V. Yu, X. Wang, D. Slack, Z. Li, P. W. Nega, E. M. Chan, A. J. Norquist, J. Schrier and S. A. Friedler, Active Meta-Learning for Predicting and Selecting Perovskite Crystallization Experiments, *J. Chem. Phys.*, 2022, **156**(6), 064108, DOI: **10.1063/5.0076636**.

7 M. Vogler, J. Busk, H. Hajiyani, P. B. Jørgensen, N. Safaei, I. Castelli, F. F. Ramírez, J. Carlsson, G. Pizzi, S. Clark, F. Hanke, A. Bhowmik, H. S. Stein Brokering between Tenants for an International Materials Acceleration Platform. *ChemRxiv*, 2022, preprint, DOI: **10.26434/chemrxiv-2022-grgrd**.

8 D. E. Fitzpatrick, C. Battilocchio and S. V. Ley, A Novel Internet-Based Reaction Monitoring, Control and Autonomous Self-Optimization Platform for Chemical Synthesis, *Org. Process Res. Dev.*, 2016, **20**(2), 386–394, DOI: **10.1021/acs.oprd.5b00313**.

9 J. Liang, S. Xu, L. Hu, Y. Zhao and X. Zhu, Machine-Learning-Assisted Low Dielectric Constant Polymer Discovery, *Mater. Chem. Front.*, 2021, **5**(10), 3823–3829, DOI: **10.1039/D0QM01093F**.

10 J. S. Manzano, W. Hou, S. S. Zalesskiy, P. Frei, H. Wang, P. J. Kitson and L. Cronin, An Autonomous Portable Platform for Universal Chemical Synthesis, *Nat. Chem.*, 2022, **14**(11), 1311–1318, DOI: **10.1038/s41557-022-01016-w**.

11 B. P. MacLeod, F. G. L. Parlane, T. D. Morrissey, F. Häse, L. M. Roch, K. E. Dettelbach, R. Moreira, L. P. E. Yunker, M. B. Rooney, J. R. Deeth, V. Lai, G. J. Ng, H. Situ, R. H. Zhang, M. S. Elliott, T. H. Haley, D. J. Dvorak, A. Aspuru-Guzik, J. E. Hein and C. P. Berlinguette, Self-Driving Laboratory for Accelerated Discovery of Thin-Film Materials, *Sci. Adv.*, 2020, **6**(20), eaaz8867, DOI: **10.1126/sciadv.aaz8867**.

12 M. L. Green, B. Maruyama and J. Schrier, Autonomous (AI-Driven) Materials Science, *Applied Physics Reviews*, 2022, **9**(3), 030401, DOI: **10.1063/5.0118872**.

13 J. H. Montoya, M. Aykol, A. Anapolsky, C. B. Gopal, P. K. Herring, J. S. Hummelshøj, L. Hung, H.-K. Kwon, D. Schweigert, S. Sun, S. K. Suram, S. B. Torrisi, A. Trewartha and B. D. Storey, Toward Autonomous Materials Research: Recent Progress and Future Challenges, *Applied Physics Reviews*, 2022, **9**(1), 011405, DOI: **10.1063/5.0076324**.

14 B. Goldman, S. Kearnes, T. Kramer, P. Riley and W. P. Walters, Defining Levels of Automated Chemical Design, *J. Med. Chem.*, 2022, **65**(10), 7073–7087, DOI: **10.1021/acs.jmedchem.2c00334**.

15 C. W. Coley, N. S. Eyke and K. F. Jensen, Autonomous Discovery in the Chemical Sciences Part I: Progress, *Angew. Chem., Int. Ed.*, 2020, **59**(51), 22858–22893, DOI: **10.1002/anie.201909987**.

16 R. D. King, J. Rowland, W. Aubrey, M. Liakata, M. Markham, L. N. Soldatova, K. E. Whelan, A. Clare, M. Young, A. Sparkes, S. G. Oliver and P. Pir, The Robot Scientist Adam, *Computer*, 2009, **42**(8), 46–54, DOI: **10.1109/MC.2009.270**.

17 A. Sparkes, W. Aubrey, E. Byrne, A. Clare, M. N. Khan, M. Liakata, M. Markham, J. Rowland, L. N. Soldatova, K. E. Whelan, M. Young and R. D. King, Towards Robot Scientists for Autonomous Scientific Discovery, *Autom. Exp.*, 2010, **2**(1), 1, DOI: **10.1186/1759-4499-2-1**.

18 E. M. Chan, C. Xu, A. W. Mao, G. Han, J. S. Owen, B. E. Cohen and D. J. Milliron, Reproducible, High-Throughput Synthesis of Colloidal Nanocrystals for Optimization in Multidimensional Parameter Space, *Nano Lett.*, 2010, **10**(5), 1874–1885, DOI: **10.1021/nl100669s**.

19 C. W. Coley, N. S. Eyke and K. F. Jensen, Autonomous Discovery in the Chemical Sciences Part II: Outlook, *Angew. Chem., Int. Ed.*, 2020, **59**(52), 23414–23436, DOI: **10.1002/anie.201909989**.

20 M. B. Rooney, B. P. MacLeod, R. Oldford, Z. J. Thompson, K. L. White, J. Tungjunyatham, B. J. Stankiewicz and C. P. Berlinguette, A Self-Driving Laboratory Designed to Accelerate the Discovery of Adhesive Materials, *Digital Discovery*, 2022, **1**(4), 382–389, DOI: **10.1039/D2DD00029F**.

21 A.-C. Bédard, A. Adamo, K. C. Aroh, M. G. Russell, A. A. Bedermann, J. Torosian, B. Yue, K. F. Jensen and T. F. Jamison, Reconfigurable System for Automated Optimization of Diverse Chemical Reactions, *Science*, 2018, **361**(6408), 1220–1225, DOI: **10.1126/science.aat0650**.

22 H. Masood, C. Y. Toe, W. Y. Teoh, V. Sethu and R. Amal, Machine Learning for Accelerated Discovery of Solar Photocatalysts, *ACS Catal.*, 2019, **9**(12), 11774–11787, DOI: **10.1021/acscatal.9b02531**.

23 D. V. S. Green, S. Pickett, C. Luscombe, S. Senger, D. Marcus, J. Meslamani, D. Brett, A. Powell and J. Masson, BRADSHAW: A System for Automated Molecular Design, *J. Comput.-Aided Mol. Des.*, 2020, **34**(7), 747–765, DOI: **10.1007/s10822-019-00234-8**.

24 A. G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, S. Sarker, C. Oses, C. Toher, S. Curtarolo, A. V. Davydov, R. Agarwal, L. A. Bendersky, M. Li, A. Mehta and I. Takeuchi, On-the-Fly Closed-Loop Materials Discovery via Bayesian Active Learning, *Nat. Commun.*, 2020, **11**(1), 5966, DOI: **10.1038/s41467-020-19597-w**.

25 K. Abdel-Latif, R. W. Epps, F. Bateni, S. Han, K. G. Reyes and M. Abolhasani, Self-Driven Multistep Quantum Dot Synthesis Enabled by Autonomous Robotic Experimentation in Flow, *Adv. Intell. Syst.*, 2021, **3**(2), 2000245, DOI: **10.1002/aisy.202000245**.

26 E. Stach, B. DeCost, A. G. Kusne, J. Hattrick-Simpers, K. A. Brown, K. G. Reyes, J. Schrier, S. Billinge, T. Buonassisi, I. Foster, C. P. Gomes, J. M. Gregoire, A. Mehta, J. Montoya, E. Olivetti, C. Park, E. Rotenberg, S. K. Saikin, S. Smullin, V. Stanev and B. Maruyama, Autonomous Experimentation Systems for Materials Development: A Community Perspective, *Matter*, 2021, **4**(9), 2702–2726, DOI: **10.1016/j.matt.2021.06.036**.

27 D. Caramelli, J. M. Granda, S. H. M. Mehr, D. Cambié, A. B. Henson and L. Cronin, Discovering New Chemistry with an Autonomous Robotic Platform Driven by a Reactivity-Seeking Neural Network, *ACS Cent. Sci.*, 2021, **7**(11), 1821–1830, DOI: **10.1021/acscentsci.1c00435**.

28 A. E. Gongora, K. L. Snapp, E. Whiting, P. Riley, K. G. Reyes, E. F. Morgan and K. A. Brown, Using Simulation to Accelerate Autonomous Experimentation: A Case Study Using Mechanics, *iScience*, 2021, **24**(4), 102262, DOI: **10.1016/j.isci.2021.102262**.

29 M. Seifrid, R. Pollice, A. Aguilar-Granda, Z. Morgan Chan, K. Hotta, C. T. Ser, J. Vestfrid, T. C. Wu and A. Aspuru-Guzik, Autonomous Chemical Experiments: Challenges and Perspectives on Establishing a Self-Driving Lab, *Acc. Chem. Res.*, 2022, **55**(17), 2454–2466, DOI: **10.1021/acs.accounts.2c00220**.

30 H. S. Stein and J. M. Gregoire, Progress and Prospects for Accelerating Materials Science with Automated and Autonomous Workflows, *Chem. Sci.*, 2019, **10**(42), 9640–9649, DOI: **10.1039/C9SC03766G**.

31 B. P. MacLeod, F. G. L. Parlane, A. K. Brown, J. E. Hein and C. P. Berlinguette, Flexible Automation Accelerates Materials Discovery, *Nat. Mater.*, 2022, **21**(7), 722–726, DOI: **10.1038/s41563-021-01156-3**.

32 M. Christensen, L. P. E. Yunker, P. Shiri, T. Zepel, P. L. Prieto, S. Grunert, F. Bork and J. E. Hein, Automation Isn't Automatic, *Chem. Sci.*, 2021, **12**(47), 15473–15490, DOI: **10.1039/D1SC04588A**.

33 W. Gao, P. Raghavan and C. W. Coley, Autonomous Platforms for Data-Driven Organic Synthesis, *Nat. Commun.*, 2022, **13**(1), 1075, DOI: **10.1038/s41467-022-28736-4**.

34 I. M. Pendleton, G. Cattabriga, Z. Li, M. A. Najeeb, S. A. Friedler, A. J. Norquist, E. M. Chan and J. Schrier, Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): A Software Pipeline for Automated Chemical Experimentation and Data Management, *MRS Commun.*, 2019, **9**(3), 846–859, DOI: **10.1557/mrc.2019.72**.

35 J. Bai, L. Cao, S. Mosbach, J. Akroyd, A. A. Lapkin and M. Kraft, From Platform to Knowledge Graph: Evolution of Laboratory Automation, *JACS Au*, 2022, **2**(2), 292–309, DOI: **10.1021/jacsau.1c00438**.

36 C. Willoughby and J. G. Frey, Data Management Matters, *Digital Discovery*, 2022, **1**(3), 183–194, DOI: **10.1039/D1DD00046B**.

37 L. Wilbraham, S. H. M. Mehr and L. Cronin, Digitizing Chemistry Using the Chemical Processing Unit: From Synthesis to Discovery, *Acc. Chem. Res.*, 2021, **54**(2), 253–262, DOI: **10.1021/acs.accounts.0c00674**.

38 P. Antsaklis, Autonomy and Metrics of Autonomy, *Annu. Rev. Control*, 2020, **49**, 15–26, DOI: **10.1016/j.arcontrol.2020.05.001**.

39 A. M. Alb, M. F. Drenski and W. F. Reed, Automatic Continuous Online Monitoring of Polymerization Reactions (ACOMP), *Polym. Int.*, 2008, **57**(3), 390–396, DOI: **10.1002/pi.2367**.

40 A. M. K. Nambiar, C. P. Breen, T. Hart, T. Kulesza, T. F. Jamison and K. F. Jensen, Bayesian Optimization of Computer-Proposed Multistep Synthetic Routes on an Automated Robotic Flow Platform, *ACS Cent. Sci.*, 2022, **8**(6), 825–836, DOI: **10.1021/acscentsci.2c00207**.

41 F. Rahmanian, J. Flowers, D. Guevarra, M. Richter, M. Fichtner, P. Donnely, J. M. Gregoire and H. S. Stein, Enabling Modular Autonomous Feedback-Loops in Materials Science through Hierarchical Experimental Laboratory Automation and Orchestration, *Adv. Mater. Interfaces*, 2022, **9**(8), 2101987, DOI: **10.1002/admi.202101987**.

42 J. H. Montoya, K. T. Winther, R. A. Flores, T. Bligaard, J. S. Hummelshøj and M. Aykol, Autonomous Intelligent Agents for Accelerated Materials Discovery, *Chem. Sci.*, 2020, **11**(32), 8517–8532, DOI: **10.1039/D0SC01101K**.

43 M. D. Wilkinson, M. Dumontier, Ij. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer,

B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, The FAIR Guiding Principles for Scientific Data Management and Stewardship, *Sci. Data*, 2016, **3**(1), 160018, DOI: **10.1038/sdata.2016.18**.

44 S. M. Kearnes, M. R. Maser, M. Wleklinski, A. Kast, A. G. Doyle, S. D. Dreher, J. M. Hawkins, K. F. Jensen and C. W. Coley, The Open Reaction Database, *J. Am. Chem. Soc.*, 2021, **143**(45), 18820–18826, DOI: **10.1021/jacs.1c09820**.

45 D. J. Walsh, W. Zou, L. Schneider, R. Mello, M. E. Deagen, J. Mysona, T.-S. Lin, J. J. de Pablo, K. F. Jensen, D. J. Audus and B. D. Olsen, Community Resource for Innovation in Polymer Technology (CRIPT): A Scalable Polymer Material Data Structure, *ACS Cent. Sci.*, 2023, **9**(3), 330–338, DOI: **10.1021/acscentsci.3c00011**.

46 M. Statt, B. A. Rohr, K. S. Brown, D. Guevarra, J. S. Hummelshøj, L. Hung, A. Anapolsky, J. Gregoire, S. Suram ESAMP: Event-Sourced Architecture for Materials Provenance Management and Application to Accelerated Materials Discovery. *ChemRxiv*, 2023, preprint, DOI: **10.26434/chemrxiv-2023-z877v-v2**.

47 J. M. P. Gutierrez, T. Hinkley, J. W. Taylor, K. Yanev and L. Cronin, Evolution of Oil Droplets in a Chemorobotic Platform, *Nat. Commun.*, 2014, **5**(1), 5571, DOI: **10.1038/ncomms6571**.

48 L. M. Roch, F. Häse, C. Kreisbeck, T. Tamayo-Mendoza, L. P. E. Yunker, J. E. Hein and A. Aspuru-Guzik, ChemOS: Orchestrating Autonomous Experimentation, *Sci. Robot.*, 2018, **3**(19), eaat5559, DOI: **10.1126/scirobotics.aat5559**.

49 L. M. Roch, F. Häse, C. Kreisbeck, T. Tamayo-Mendoza, L. P. E. Yunker, J. E. Hein and A. Aspuru-Guzik, ChemOS: An Orchestration Software to Democratize Autonomous Discovery, *PLoS One*, 2020, **15**(4), e0229862, DOI: **10.1371/journal.pone.0229862**.

50 J. Li, Y. Tu, R. Liu, Y. Lu and X. Zhu, Toward "On-Demand" Materials Synthesis and Scientific Discovery through Intelligent Robots, *Advanced Science*, 2020, **7**(7), 1901957, DOI: **10.1002/advs.201901957**.

51 J. Li, J. Li, R. Liu, Y. Tu, Y. Li, J. Cheng, T. He and X. Zhu, Autonomous Discovery of Optically Active Chiral Inorganic Perovskite Nanocrystals through an Intelligent Cloud Lab, *Nat. Commun.*, 2020, **11**(1), 2046, DOI: **10.1038/s41467-020-15728-5**.

52 D. Allan, T. Caswell, S. Campbell and M. Rakitin, Bluesky's Ahead: A Multi-Facility Collaboration for an a La Carte Software Project for Data Acquisition and Management, *Synchrotron Radiat. News*, 2019, **32**(3), 19–22, DOI: **10.1080/08940886.2019.1608121**.

53 J. Wagner, C. G. Berger, X. Du, T. Stubhan, J. A. Hauch and C. J. Brabec, The Evolution of Materials Acceleration Platforms: Toward the Laboratory of the Future with AMANDA, *J. Mater. Sci.*, 2021, **56**(29), 16422–16446, DOI: **10.1007/s10853-021-06281-7**.

54 S. Steiner, J. Wolf, S. Glatzel, A. Andreou, J. M. Granda, G. Keenan, T. Hinkley, G. Aragon-Camarasa, P. J. Kitson, D. Angelone and L. Cronin, Organic Synthesis in a Modular Robotic System Driven by a Chemical Programming Language, *Science*, 2019, **363**(6423), eaav2211, DOI: **10.1126/science.aav2211**.

55 S. H. M. Mehr, M. Craven, A. I. Leonov, G. Keenan and L. Cronin, A Universal System for Digitization and Automatic Execution of the Chemical Synthesis Literature, *Science*, 2020, **370**(6512), 101–108, DOI: **10.1126/science.abc2986**.

56 S. Rohrbach, M. Šiaučiulis, G. Chisholm, P.-A. Pirvan, M. Saleeb, S. H. M. Mehr, E. Trushina, A. I. Leonov, G. Keenan, A. Khan, A. Hammer and L. Cronin, Digitization and Validation of a Chemical Synthesis Literature Database in the ChemPU, *Science*, 2022, **377**(6602), 172–180, DOI: **10.1126/science.abo0058**.

57 P. S. Gromski, J. M. Granda and L. Cronin, Universal Chemical Synthesis and Discovery with 'The Chemputer, *Trends Chem.*, 2020, **2**(1), 4–12, DOI: **10.1016/j.trechm.2019.07.004**.

58 B. Miles and P. L. Lee, Achieving Reproducibility and Closed-Loop Automation in Biological Experimentation with an IoT-Enabled Lab of the Future, *SLAS Technol.*, 2018, **23**(5), 432–439, DOI: **10.1177/2472630318784506**.

59 Autoprotocol, **https://autoprotocol.org/**, accessed 2023-05-15.

60 V. Lenarduzzi, T. Besker, D. Taibi, A. Martini and F. Arcelli Fontana, A Systematic Literature Review on Technical Debt Prioritization: Strategies, Processes, Factors, and Tools, *J. Syst. Softw.*, 2021, **171**, 110827, DOI: **10.1016/j.jss.2020.110827**.

61 A. J. S. Hammer, A. I. Leonov, N. L. Bell and L. Cronin, Chemputation and the Standardization of Chemical Informatics, *JACS Au*, 2021, **1**(10), 1572–1587, DOI: **10.1021/jacsau.1c00303**.