EBU

OPERATING EUROVISION AND EURORADIO

TR 055

AUTOMATION: APPLICATIONS, TECHNIQUES AND TECHNOLOGIES

TECHNICAL REPORT

Geneva
June 2020

Abstract

Automation will play a key role as broadcasters transform their businesses to an all-IT approach.

Automating otherwise tedious and potentially error-prone tasks will help free up programme-maker effort to concentrate on more important work, while machine learning will create more data, increasing the value of content. Automation will also facilitate more flexibility in where operations can happen, and where they are resourced, including on location, at broadcaster facilities or “in the cloud”.

This will be an unfamiliar area for many, involving new techniques and technologies, and terminology that often differs from that used in more traditional broadcasting.

In this report we make a case for various types of automation and explain some key concepts and terminology, giving examples for how automation is being used by EBU members for development, deployment and configuration activities. We then look more broadly at the challenges that will face broadcasters considering automation across the enterprise, such as managing resource conflicts, and the need for staff training.

Acknowledgement

This publication was prepared by the Automation and Provisioning subgroup within the Strategic Programme on infrastructure & security (InfraSec). The group was chaired by Peter Brightwell (BBC), who also edited the document with key contributions from:

Gordana Macher and Nadege Nzoyem (IRT), Daniel Freytag and Markus Ostertag (SWR), Hugo Ortiz (RTBF), Karel De Bondt, Kristof Demulder, Joris Grauwels, Dries Tastenhoye and Willem Vermost (VRT), Nestor A. Amaya, Alexandre Dugas, Louis Lamarre, Sunday Nyamweno, Felix Poulin and Loc Tran (CBC/Radio Canada), Tim Cowin, Mike Ellis, Tom Dawes, James Gibson, Stuart Grace, Ivan Hassan, Nick Hopewell, Luke Hudson, Sam Mesterton-Gibbons, Jemma Phillips, Roland Rodgers, Ruth Sloan and Duncan Walker (BBC), Hans Hoffmann, Ievgen Kostiukevych (who also managed the project) and Paola Sunna (EBU).

Contents

Automation:
Applications, Techniques and Technologies

EBU Committee	First Issued	Revised	Re-issued
TC	2020		

Keywords: Broadcast, Automation, IT-based, Infrastructure, Cloud, Business Process, Robotic, Workflow.

1. Introduction

Broadcasters' businesses are changing to cope with an increasingly complex media landscape. Content needs to be available in many forms to suit different types of audience and platform. Productions engage with audiences in new ways and make increasing use of data and archive material. Programmes often involve many production partners and locations, with remote recording and operation becoming commonplace[1].

As a result, broadcasters' infrastructure has moved away from fixed function facilities to become more reconfigurable and relocatable, moving towards IT-based platforms based on IP connectivity, including on-premises and public cloud.

All the above has the potential to significantly increase the workload on editorial, operational and technical staff. Much of this will be mundane activity that is prone to human error, for example manually maintaining a list of network addresses or copy-pasting programme information between systems. The more such work be done automatically, the more time people will have to spend on creative aspects of their jobs.

Automated operation is nothing new for our industry. The term "automation" has long been used in quite specific ways, such as the mechanism for scheduled playout of television and radio programmes from a server, or to automatically change parameters such as audio level. But there are many opportunities to go further. Below are some examples, some already in use by EBU members, and others future possibilities. We explore a few of these in more detail in later sections, including what this might mean for broadcasters.

- Automated tools can help with resource allocation, including booking studio space, cameras and operators, and reserving graphics rendering capacity according to programme scheduling. If schedules need to change (for example because of breaking news), automation will help with re-prioritisation and reallocation of resources.
- Speech and video analysis can help automatically create clips and data as content is captured.
- When production content is delivered, several automated tests are triggered, to check technical quality, check the format is correct, check all required data is available.
- Audio analysis helps with reporting and compliance for music in programmes.

[1] This report was drafted during the COVID-19 lockdown of Spring 2020, when the majority of broadcasters' staffs were working from home.

- Different versions of a programme are made automatically, for different audiences, or for different platforms.
- Equipment and software can be kept up to date through automated testing and deployment; for example, rolling out a system update in a data centre, applying security patches to servers, and testing a new version of an on-demand web platform.
- Automation can help reduce the manual overhead in activities such as populating programme schedules, compliance risk assessment and contributor rights calculation and payments.

2. What do we mean by…?

The above examples cover a wider range of automated activity, and – not surprisingly – a range of different terminology is used. Unfortunately, this is not always consistent; for example, some use “orchestration” to refer to how different automated activities are sequenced, while others use it about how resources are made available in a consistent way. The same term might be used differently by different equipment and software vendors, and it is best to find out what is meant in each case.

With that caveat, we use several terms in specific ways in this report. These appear in **bold text** as they are introduced, starting with those shown in Figure 1:

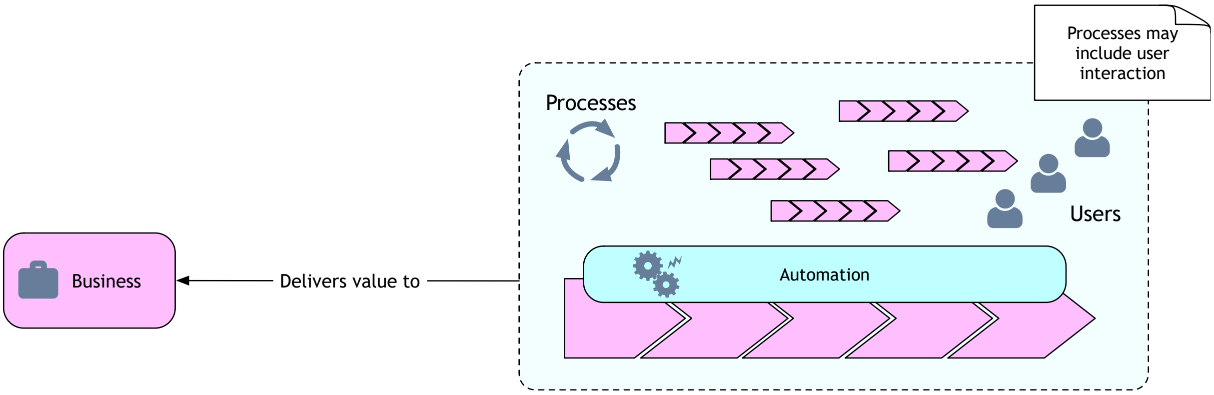

Figure 1: Business, Processes, Users, Automation

- A **business** is an organisation such as a public broadcaster, a commercial subsidiary, partnership, venture etc.
- A **process** is a defined pattern of repeatable activity steps that provides direct or indirect benefit to a business. Two (very different) examples are (a) upgrading the operating system on the servers in a data centre, and (b) creating and distributing a news programme
- A **user** is a person (for example a member of a production team, a developer, or technical operator) that interacts with a process. That interaction could be to initiate a process that is then carried out automatically (as in (a) above) or the user could perform some of the activity required for the process (as in (b)).
- An **automation** makes some or all steps of a process happen without user intervention, for example to run several checks on a delivered MXF file.

Let us now consider what typically happens “behind the scenes” to make automation happen (Figure 2):

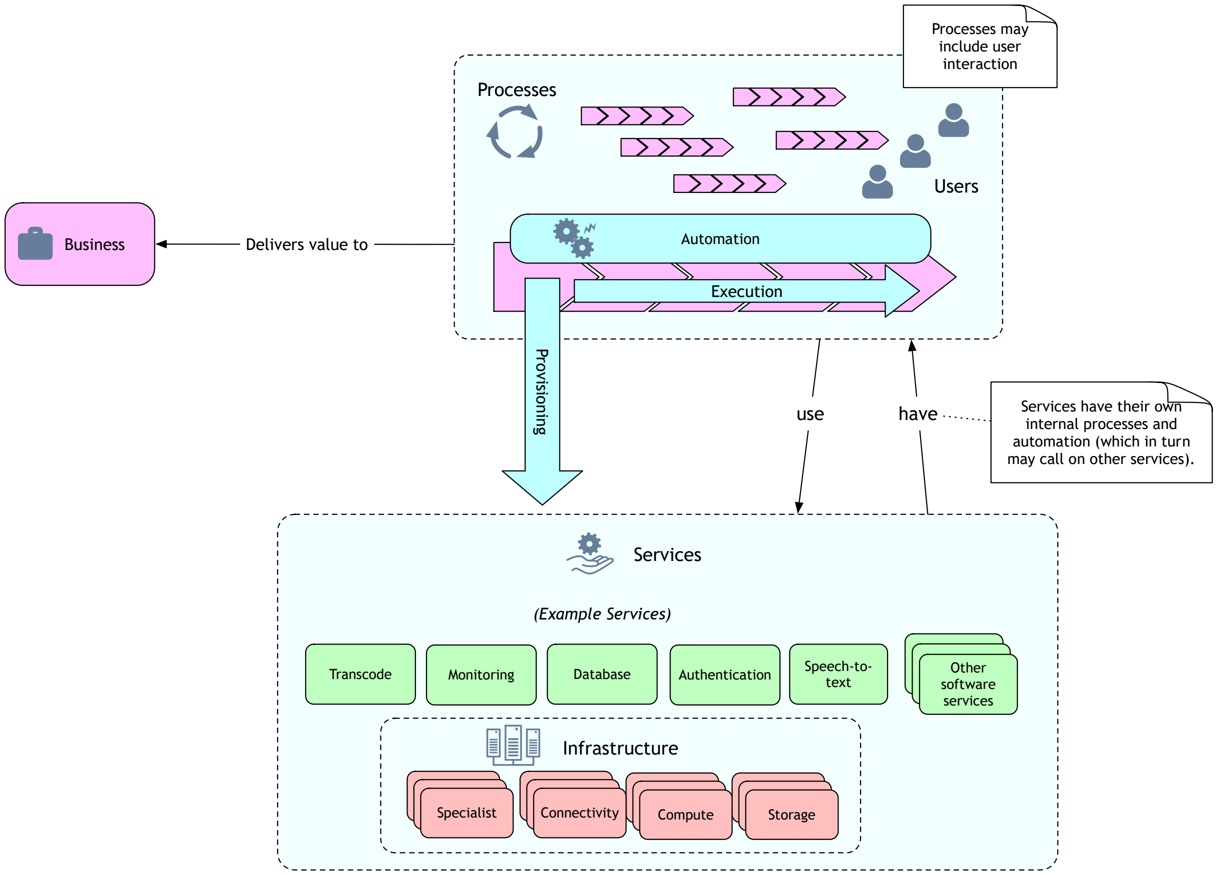

Figure 2: Services, Provisioning, Execution, Infrastructure

- A **service** carries out some functionality required by automation. Here we are primarily concerned with software services (which will have their own internal processes and automation). For our MXF example might make use of a PSE (photo-sensitive epilepsy) checking service.
- **Provisioning** concerns how services become available to an automation. This might include creating virtual machines, starting up cloud-based services or allocating storage.
- **Execution** concerns how an automation carries out the steps of the process by invoking services.
- Services ultimately depend on **infrastructure** such as compute, network, storage, or specialist hardware. This could be on broadcaster premises, or at a cloud provider.

3. Automation scenarios

We now examine some cases where automation can be of use, including examples from EBU members.

3.1 Managing IP-based broadcast facilities

Traditional broadcast facilities were built around large items of broadcast infrastructure, such as SDI and AES routers. Typically, these would use both *static configuration*, perhaps with a spreadsheet holding port names (which don’t change often), and *dynamic configuration* such as cross-connection setting (which changes frequently).

In a typical IP-based facility, the large routers are replaced with many networked media devices and switches, which again require both static and dynamic configuration. In many cases this is done through a REST API. (Often these APIs are proprietary - EBU Tech 3371 recommends the use of open APIs.)

Doing this manually is tedious and error-prone, even if the configuration does not change often. The need to update device software/firmware, and to reconfigure devices (to provide the flexibility that IP promises) makes automation essential. A basic way to do this is to run shell scripts, but these are not fool proof, especially if a failure happens part-way through.

Fortunately, we can learn from the wider IT world, which has learned how to manage virtualised and dynamic facilities effectively. There are good provisioning and configuration management tools available for managing IT facilities that can be used to do a better job. Some well-known examples are Chef, Puppet and Ansible.

As an example, CBC[2] uses Ansible to automate configuration of 250 Embrionix IP endpoint devices for their new facility in Montréal.

With Ansible, a controlling machine looks up an inventory of nodes (devices) to be configured, and a **playbook** of actions to carry out on the node. This can use an API, or - as in this case - the controlling machine uses SSH to temporarily install a configuration module on the node. This is *idempotent*, meaning that the resulting state is the same even if the process is repeated - this is helpful, for example, if there is a connectivity problem part way through.

Doing this entirely manually would take days and would inevitably lead to errors. Some user interaction is still required, as shown in Figure 3, but through partial automation, it took CBC three hours to install new licences on all devices, and 20 minutes to update all their IP addresses. Furthermore, after new firmware was found to cause a problem on 144 devices, it was possible to roll back the update in 30 minutes.

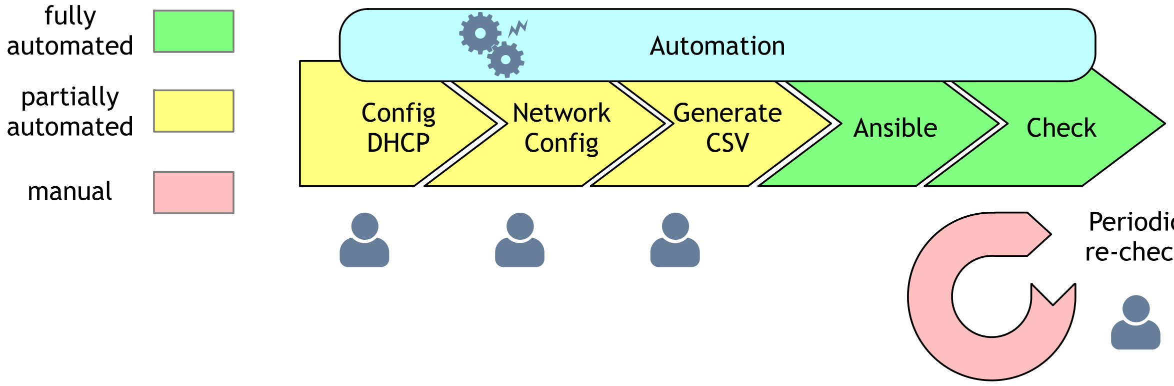

Figure 3: Automating the configuration of IP devices

By using such automation in a systematic and continuous way and using automated testing after any changes have been made, we can have confidence that the running systems stay correct, allowing facilities to remain operational during updates. This is called a **continuous deployment** (CD) approach and it is common practice in running IT data centres.

[2] Canadian Broadcasting Corporation / Radio-Canada

3.2 Cloud-based media operations

Broadcasters already make much use of cloud-based processes (for example transcoding and web playout). All signs are that this will extend to other parts of their business, including production processes. This could make use of services on public cloud (e.g. AWS, GCP), or on-premises infrastructure using a cloud framework (e.g. Azure Stack, OpenStack). Feedback from EBU members suggests a hybrid approach is possible; some processes (such as studio recording, some types of edit) that require large amounts of "local bandwidth" use on-premise services, while others (such as remote recording, and personalisation/recommendation) use public cloud.

In either case, automation is essential for running services on cloud-based infrastructure. Fortunately cloud frameworks include services (e.g. Amazon CloudFormation, Azure ResourceManager, OpenStack Heat/Magnum [3]) to help hide the complexity.

In the past, "putting something in the cloud" has meant moving from a physical server to cloud-hosted virtual machine and storage, with only partial automation. However, there is now a trend towards using microservices in media operations. These break down the parts required by processes as far as possible and expose them as lightweight services, often as "functions" or "lambdas". Microservices often form part of serverless architectures, in which services are provided by a third party and the developer/user need not be concerned with the details of the infrastructure (such as deploying virtual machines).

This approach is being investigated by the EBU's MCMA project, which is building open source microservices for AI-based processes [4] such as speech-to-text and social media tagging (some of this is now in use at Bloomberg). An aim of the project is to minimise the user development overhead by making maximum use of what is provided in the cloud platform. The project is investigating technologies used to automate this, which is the subject of the next section.

3.3 Continuous integration and deployment of services

A common methodology in the wider software development world is **continuous integration** (CI). In a CI environment, the software services and tools are kept in a state where they are always as correct as possible – the idea is that any issues introduced during development are identified and addressed as soon as possible, and especially before they are deployed (where they might cause failures). CI is often used by teams practicing agile development, which is becoming increasingly common.

In a CI approach, code is normally kept in a shared repository, and automation is used to check code (linting), to compile (and where required, link), to run tests, and to build **artefacts**. These are the software parts that comprise services and tools. Several mature tools are available for CI automation, many of them open source. A commonly-used example is Jenkins, which was developed from an earlier proprietary tool (Hudson). Depending on what is required, CI automation tools can run locally or in the cloud, and there are also third-party cloud-based CI services such as CircleCI and TravisCI to choose from

The artefacts created through continuous integration can then be deployed automatically through continuous deployment automation, or the deployment may be triggered manually (this is often

[3] OpenStack adopters often take a more "hybrid" approach to tooling, so might instead/also use automation technologies from outside the OpenStack framework.

[4] MCMA uses the term "workflows" rather than "processes"; it is another term that is used to mean multiple things.

called continuous delivery). Deployment might be onto a physical platform (as above), or onto some type of virtualised platform.

An example of continuous integration can be seen in BBC R&D[5]'s work on developing cloud-based media services for streaming, storing and processing media. Source code is stored in GitHub repositories, with changes managed through pull requests, a common technique in which proposed changes are peer-reviewed and corrected before being merged into the main code. Tests are applied automatically when pull requests are made to spot obvious errors.

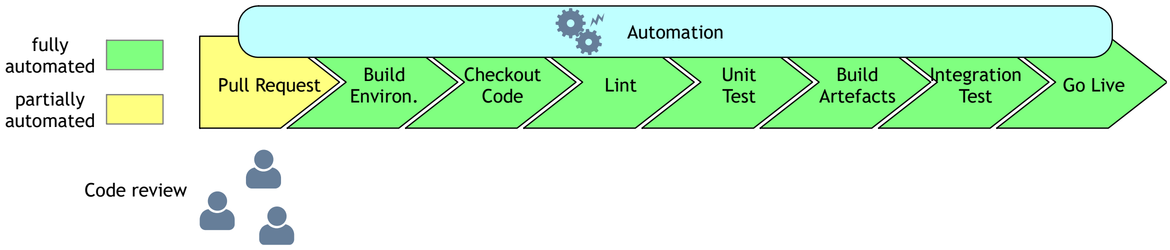

Figure 4: Continuous Integration

Developers work in their own virtual environments; these are provisioned using Vagrant, an open-source tool for automation, configuration and management of VMs. Testing and building of artefacts is automated by Jenkins, and Terraform (another open-source tool) builds a cloud-ready environment for deploying these artefacts. Finally, Ansible handles provisioning and configuration of the actual deployment. This is all coordinated through custom Python tools.

As in the example above, continuous integration is often used with VM-based virtualisation, but there is a trend to more fine-grained deployment of artefacts. **Containerisation** is now often used in CI environments, with Docker (again, open source) probably the best known offering. Like VMs, containers allow multiple isolated compute/storage/network instances to exist on the same hardware, but unlike VMs, they make use of the host OS's system calls infrastructure, making them much more lightweight.

Recently, automation software has been developed for managing containerised environments; the best-known of these is probably Kubernetes, originally developed by Google but now available as a service on all major cloud platforms. However, as containerisation is still a relatively new technology, it is yet to find significant adoption by broadcasters.

3.4 Business Process Automation

The scenarios so far described have been mostly technical, but automation can help reduce manual overhead in more general processes that have grown organically and that still depend on manual use of documents, spreadsheets, web forms, emails, and other "electronic paperwork". These are likely to suffer from:

- Reliance on possibly insecure spreadsheets and other generic office tools
- Errors caused by re-keying data
- Difficulty in knowing the status of a process
- Lack of metrics to support improvement

[5] British Broadcasting Corporation's Research and Development department.

- Lack of agility to support low cost operating models

The aim is to create exception-based "straight through" processes based around automation of repetitive steps, with users providing higher-value input where their cognitive skills are required. This can be achieved by combining the following technologies:

- **Robotic Process Automation** (RPA) tools process inputs from various sources (spreadsheets, PDF documents, web forms, etc.) and enter data into human-oriented user interfaces. In effect RPA creates a pool of "virtual workers" and is a means of integrating automation where more computer-friendly interfaces (e.g. APIs) are not available. Examples of RPA products include UIPath, BluePrism and Automation Anywhere.
- **Workflow Engines** (also known as business process management engines) manage the provisioning and execution of processes; they make decisions on what to do next and assign tasks to users or automations (including RPAs) based on the available resources Business Process Modelling Notation (BPMN) and Decision Modelling Notation (DMN) are widely-adopted notations for telling the engine what it needs to know about the process and decisions it has to make. Examples of workflow engine products include Bizagi, Aris, Alfresco Process Services, Imixs, Activiti, Camunda, Zapier and Pipefy.

The BBC has tried out UIPath and Camunda for automating some business processes. An example is for handling TV repeats, including artist payments. Figure 5 shows a very-simplified version of the overall process; in practice, however, the workflow engine must take into account different sets of rules for compliance and payments, such as when, and how many times a programme has been repeated. Based on these, the engine makes decisions about what step is needed next, and how it shall be performed.

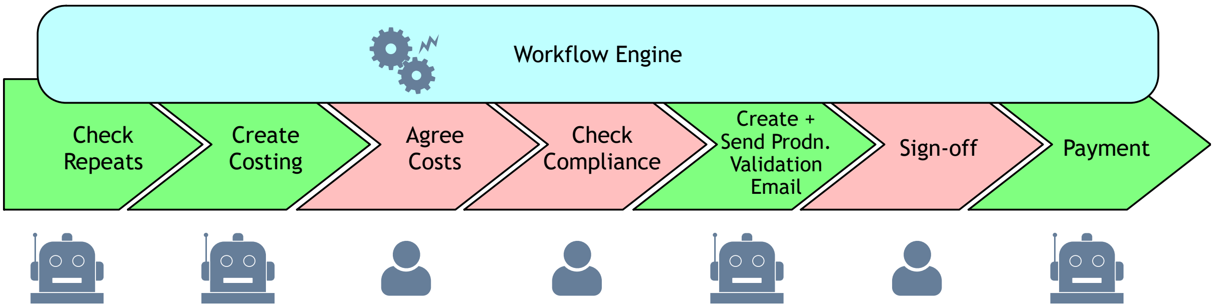

Figure 5: Simplified automation for TV repeats and artist payments

4. Managing constraints and priorities

As automation allows broadcasters to scale their processes dynamically to meet business demands, they will need to consider how to manage access to the services and underlying infrastructure, whether these are local or provided by a third party such as a public cloud. There may be constraints on the use of a service, including:

- Need to use specific infrastructure, as in the example above of a local rendering cluster, or access to a specialised item of equipment.
- Prices charged by third-party service.
- Latency of accessing remote services; this is likely to be important for some types of real-time production process.

- Available bandwidth for remote services.
- Connectivity costs of accessing remote services.
- Environmental impact, such as the power, cooling and space requirements of local vs cloud infrastructure
- Data sensitivity and security; cloud providers consider this especially important, but regulations may also prohibit storage of some data remotely or outside the country/EU.
- Operational factors important to the service; for example, it may be necessary to run some service locally due to on-site expertise, while other services can be stopped, moved to a different infrastructure and restarted.

Such constraints may influence *where* and *how* it should run, and maybe *when* it should run; for example, where energy costs or available bandwidth vary through the day.

Furthermore, some processes may compete for access to services running on limited infrastructure. To resolve any such conflicts, we can consider the idea of an **orchestrator**[6] (see Figure 6). This takes account of what the process needs and also the priorities of the business, which might for example favour processes required to air this evening's news bulletin, or for live football coverage.

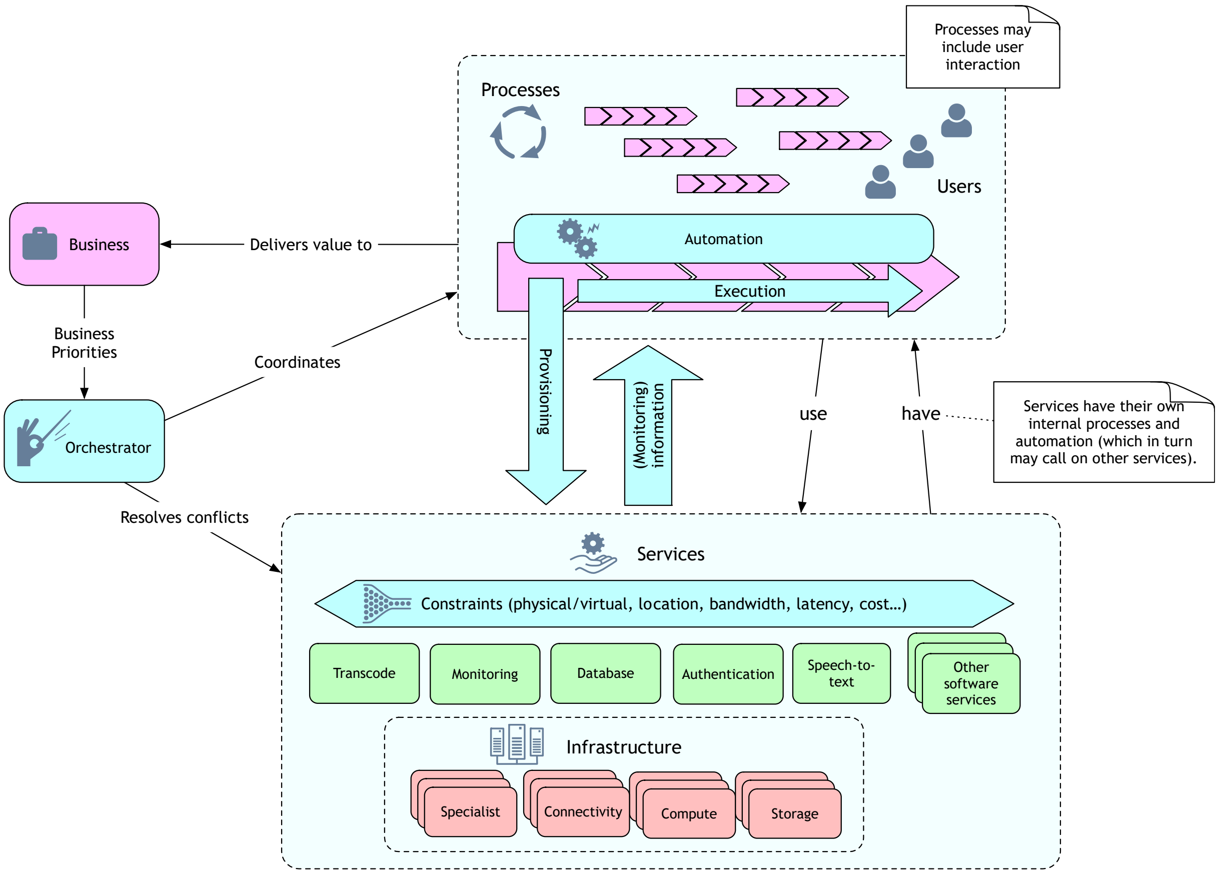

Figure 6: Business priorities and conflict resolution

As an example, consider a case where two production processes both use a rendering service that would normally run on local infrastructure. One process is for a programme that will go to air soon, and it needs priority access to the render service to avoid the deadline being missed. The orchestrator

[6] As noted in § 2, "orchestration" tends to be used in two different ways; the orchestrator in Figure 6 is more concerned with the "how resources are used in a consistent way" meaning.

knows that this is a priority, and knows about deadlines, and can resolve any possible conflict. This may involve delaying (or pausing) the later-to-air process until the first-to-air render has finished or it may involve moving the second-to-air render to an off-site facility.

Also consider a live production process that needs access to a locally hosted service for latency reasons, but the infrastructure is too busy to accommodate it. The orchestrator may move more latency-tolerant production processes, or even non-production processes such as document processing, onto a public cloud, or they might be paused until a later time.

The orchestrator functionality discussed here is a theoretical concept rather than something that is available in a single product; it perhaps may be implemented through a combination of business process management, resource management and automation tools.

5. Monitoring, logging and analysis

It is important to have timely and accurate information about automated processes. Typical functionality will include:

- Providing **metrics** about the state/performance of the process and the services and infrastructure it uses. These might be **monitored** continuously or **measured** on request.
- Sending an **alert** when an unusual situation occurs (e.g. a metric goes out of range, or a service fails); this might trigger an automated failover action or might raise an **alarm** to inform an operator that action is needed.
- **Logging** to help track down problems (e.g. if the process takes longer than expected, or picture quality is degraded).
- Providing a **dashboard** of current and historic metrics.
- Processing of metrics to support **operational and trend analysis**; this is discussed below in the context of business transformation.

Figure 7 shows a typical architecture used to provide such functionality.

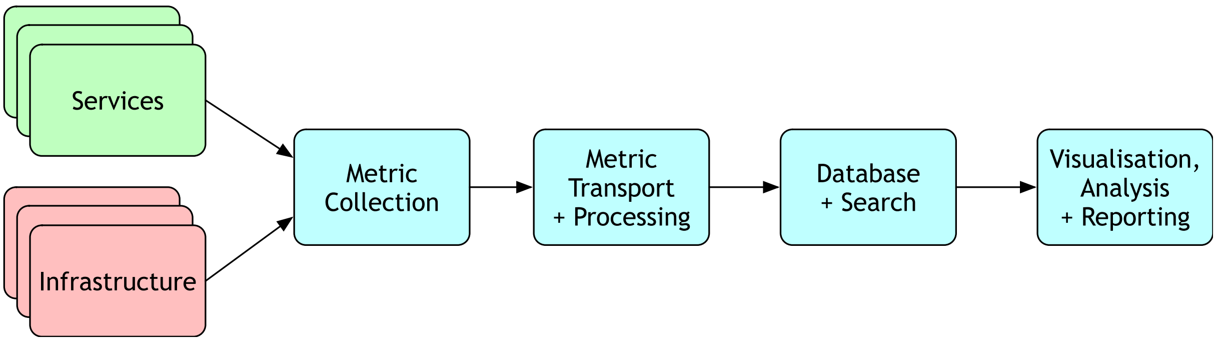

Figure 7: Typical monitoring and logging architecture

In practice, such an architecture is likely to use multiple technologies. This is particularly true for collecting metrics, where low level techniques such as pinging, getting system logs and HTTP/SSH/SNMP polling and traps may be useful, alongside vendor-specific telemetry products, and open-source collection technologies such as Prometheus. For network infrastructure, Netflow, sFlow, and more recently OpenConfig, are useful for conveying network/traffic metrics to network/server logging products such as Nagios, Zabbix, Observium and LibreNMS.

Agents such as Apache Flume, Telegraf, and Beats help collect metrics and send through queues (e.g. Kafka, RabbitMQ) into databases (e.g. InfluxDB, HDFS) for further use. Products such as Logstash provide more abstraction and higher-level logging, and tools such as Kibana, Grafana and Kapacitor provide visualisation, analytics and prediction.

There are of course many more monitoring, logging and analysis tools, including complete frameworks such as Splunk, AppDynamics and ELK[7] as well as specialist tools provided by broadcast vendors, and services provided by cloud vendors (e.g. Amazon CloudWatch and Azure Monitoring).

6. Automation for business transformation

Aside from automation providing tactical help with how broadcasters transform their business processes, it also provides value at a strategic level.

Processes can be seen in terms of the business capabilities they provide; these describe the ability of the broadcaster to deliver value to its audience and other stakeholders. This value can be used in assessing the benefit of automating processes, which feeds into development of a business strategy (Figure 8). The EBU Business Capabilities Map[8] identifies capabilities that are important to broadcasters and can be used to identify where processes should implement technology capabilities such as Metadata Tagging and Compliance Checking.

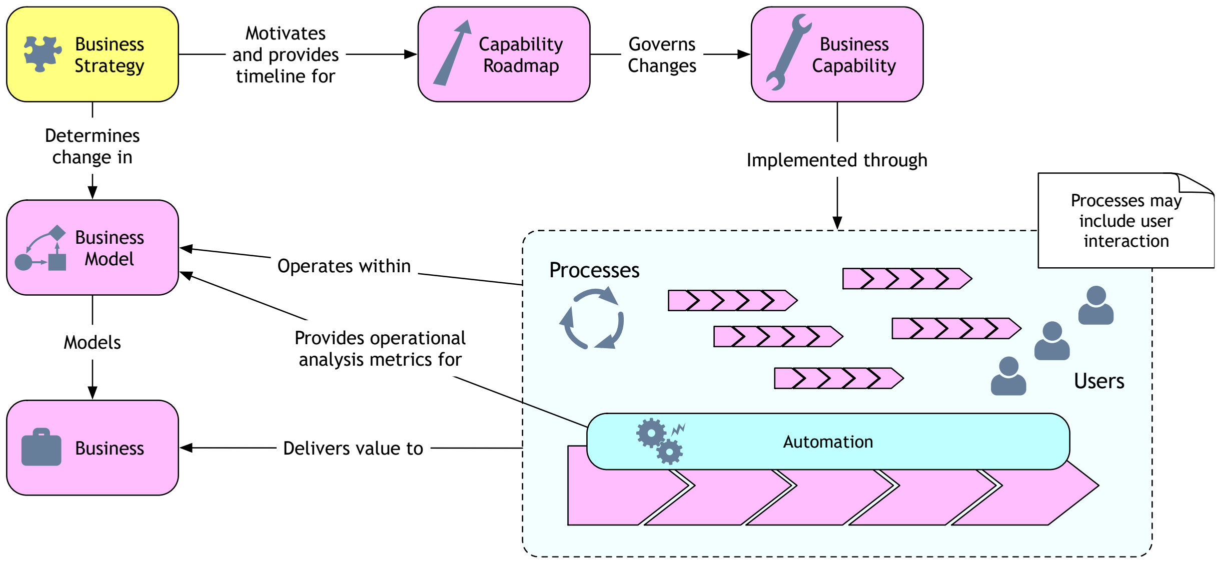

Figure 8: Business Transformation

Automatically generated data enables the use of AI/ML techniques (as well as high-level monitoring; see above) to generate metrics for predicting how business processes and models may need to change. This is called predictive operational analysis. For example, content schedules and services can be managed based on expected demand, by measuring previous demand.

[7] Beats, Logstash, Elasticsearch and Kibana are often used together; this “stack” of products is known as ELK or Elasticstack.
[8] At time of writing, this publication is available to EBU members, exclusively.

7. Challenges

Broadcasters' processes have been carefully refined over many years to "get the job done" efficiently, often to meet on-air deadlines. As business models change, and the role of automation widens, many challenges will be encountered in adopting what are still relatively new technologies.

These go further than just knowing what the best tool is to use and should be seen in the context of how to tackle business transformation more widely.

Staff will need training, both in how to work the automation technology, and also how it might affect what they are used to; for example any latency of an automation system may affect how soon it can be made available, and reaction times in operation. New monitoring tools may require different techniques for identifying operational problems, and the move towards more dynamic provisioning may be unfamiliar for some.

While automation reduces dependence on laborious manual tasks, maintaining and managing the technology will introduce some new overheads. In some cases, this can be delegated to a third party or cloud provider, albeit at a price, but broadcasters will need to allow for some manual intervention when processes/services fail, and they should include sufficient monitoring to meet this requirement.

All industries have specialist needs, and broadcasting is no exception. Often this is related to real-time operation and the latencies that are introduced. In some cases, specialist infrastructure may be unavoidable, introducing challenges of integration with automation systems; ease of integration should be considered during design and procurement. When choosing technology, it is important to consider how it can be automated (and understanding how vendors use terminology!)

In practice, automation systems may need to work with multiple, different (and often proprietary) APIs, so interoperability and open approaches for configuration and control are becoming increasingly relevant. Specialist needs may also introduce constraints that will need to be factored in as broadcasters how to realise the orchestrator concept within their architectures.

More generally, as broadcasters move towards virtualised and cloud-based services, they will need to consider how they balance capital versus operational costs (CapEx vs OpEx). For example, they may reduce the amount of on-premises infrastructure so that it can be busy most of the time, with open-source cloud used to handle peaks, something that may have implications for the architecture of automation solutions.

As existing processes become more automated, there will be the challenges of change management to keep business running during the migration, those concerning the effect on the numbers and types of job, and those concerning the training that will need to be supported.

Security is a major concern for broadcasters; while automation can help reduce vulnerabilities caused by operator error or outdated software, attackers may also try to exploit the automation technology itself, for example by delivering malicious media files to a production. Robust checking and monitoring will be important to avoid such problems. The EBU Media Cyber Security group has published several relevant best practices and technical recommendations for broadcasters.

Finally, as broadcasters re-architect their processes and technologies, they need to consider what might happen in the future; the more agile the approach that they can take now, the easier it will be to adapt to later changes. This is particularly important at the time of writing as broadcasters consider how they will function post-COVID-19 crisis. Automation suggests a mature approach to process management, and more generally is one indicator of the organisation's maturity to support changing goals and objectives. There may be something to learn from Operational Excellence methodology as practiced by organisations such as Netflix, Amazon and Google.

8. Conclusions

Automation will play a key role as broadcasters effect their transition to an all-IT approach, by freeing up effort from error-prone manual tasks to concentrate on more creative activities. It will help increase the value of content through AI/ML-based creation of data and will provide more flexibility in how and where processes can run, including on-premises and in a public cloud.

This will be an unfamiliar area for many, with sometimes-confusing terminology and a wide variety of commercial and open-source tools and services available. In this report we have explored some of the technology and terminology used in a number of areas where EBU members are using or investigating automation: configuration of live IP infrastructure, running media processes in cloud-based services, continuous development and deployment of such services, and using robotic process automation and workflow engines to modernise general business processes.

Automation introduces challenges along with opportunities. Potential conflicts need to be resolved where resources are limited; we have presented examples of how this might occur, and what this may mean for future architectures. Architects and users will have to learn new skills and businesses need to consider the implications on their staff. Virtualisation, containerisation and serverless architectures will provide more options for how and where automated processes can be provisioned, which in turn will affect future business models and procurement.

The EBU's Business Capability Model provides a systematic means of identifying the scope and value of automation, by considering capabilities across people, process and technology. This may be of use to broadcasters in prioritising their own automation initiatives and funding.