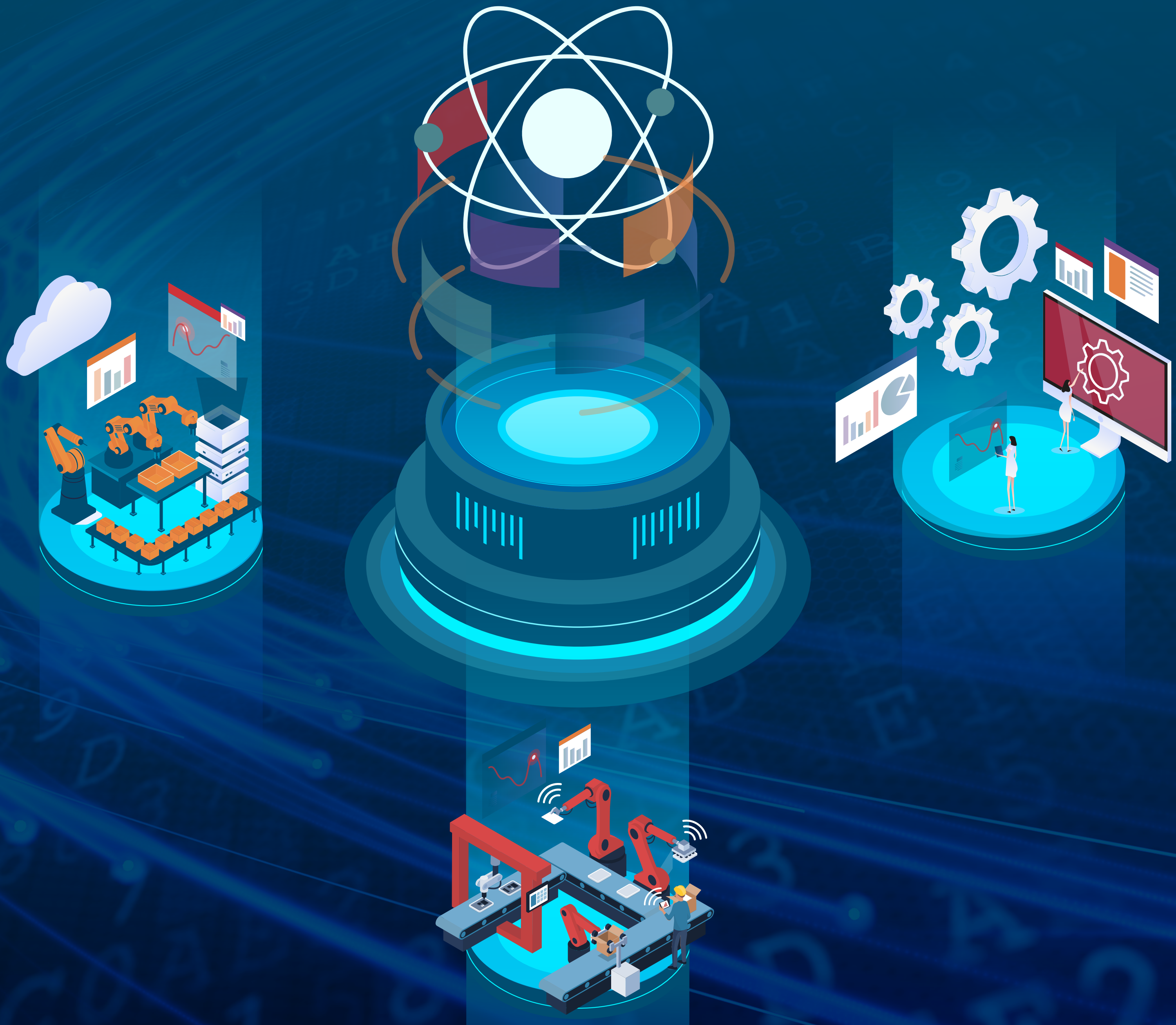
WHITEPAPER

AEGIS
SOFTWARE

CONTEXTUALIZATION:
THE ART OF DATA SCIENCE



The majority of computerized software solutions are simple in nature. Data is acquired through some input mechanism, and effectively converted into some form of output. We are used to various software solutions in manufacturing displaying operational status, triggering actions based on a threshold value and rules. How simple or complex the data and the algorithmic rules behind the decision-making is not apparent in most cases, as the best software has built-in automation that requires little from the user, no matter how simple or intelligent the process may be.

In movies, we see animated displays, scrolling images, code dumps, flickering light shows and sound effects that help the narrative along, as some form of software intelligence does whatever it is meant to be doing. Obviously, the majority of what is being shown is irrelevant—only there to show how significant the activity is meant to be. In reality, the cleverness of the software, whether you think of it as "AI" or not, is intangible from the perspective of the user. This ought to be the case, as, unlike in a mathematics examination, users only need to know the results, not the workings out. **The cleverer the software, the less user involvement is required, with software being designed following established built-in ontology that provides value-generation based on knowledge of the interaction, and relationship between data points, configurations and processes.**

In the case of FactoryLogix, if MES users were exposed to the hundreds of different ways that data is being simultaneously processed, it would be quite distracting. In manufacturing, people in key roles need to get on with their work. After all, most of the time, the majority of the factory is working without the need to take unplanned actions relating to some event or emerging trend. When such conditions do occur however, **it is essential to have full visibility of causes, effects, on-going challenges, opportunities and potential consequences.** In this paper, we take a look behind the scenes to discover examples of how FactoryLogix processes data, contextualizing different elements together, using rules and algorithms specifically developed for the most beneficial Smart factory execution, out of the box, with the lowest code development overhead in its class.

Table of Contents

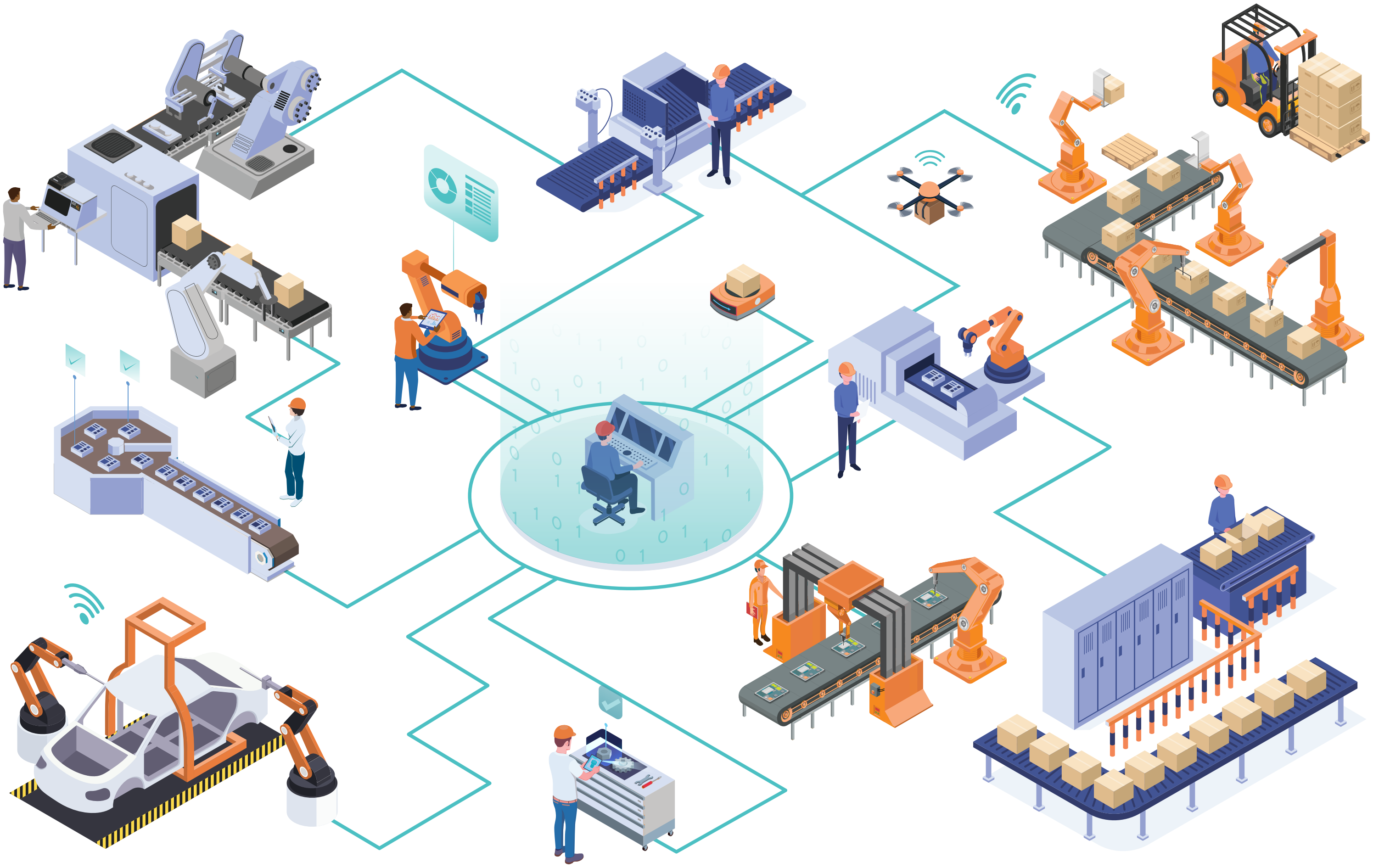
AEGIS SOFTWARE

Perspectives

The art of data science is to create actionable value from an expansive array of data points that are related to each other in several different ways, which can be thought of as "perspectives." The data content from a single IIoT message sent by an automated process contributes to many different perspectives, relating to such things as, for example, operational performance, product quality, materials and supply-chain, conformance and traceability, etc.

Each perspective utilizes the data within each message in a different way, combining the contributed information with that from other data points, as well as the internal live digital-twin model of the production operation. FactoryLogix is a single-platform IIoT-based solution, providing the ability to contextualize data from many different sources, together with knowledge of production configurations and product data, replacing the need for the many legacy and disparate "point solutions," as well as the complex connecting infrastructure that would be otherwise needed.

This approach ensures high performance, accuracy, consistency and conformity across the holistic manufacturing operation, avoiding duplication of effort and cost, as well as preventing ineffective or irrelevant actions from being needed.



Micro-facts

As FactoryLogix considers each perspective, an associated set of "micro-facts" is created. Micro-facts are calculations based on simple aspects of information derived from multiple messages, set against the knowledge of the current operation. An example of a micro-fact could be the length of time it took for a production unit to complete processing on a production line configuration, based on the arrival time of a product to the first station on the line and the departure time at which the production unit left the final station on the line. The micro-fact itself is a very simple calculation of the difference between the two time recordings, representing the overall processing time taken on the line, but can be used in a variety of different perspectives:

Micro-facts

- Continuous indication of line performance against target, including OEE (Operational Equipment Effectiveness).
- The trigger for replenishment materials, or the report of consumption to ERP.
- The ability to extrapolate line conditions to determine when related or dependent work-orders can be reliably started on this configuration.
- Analysis of the variation in timings between products to discover the degree to which the production configuration is under control, used to highlight risk towards potential delivery issues, equipment failure or other disruptions that may need attention.
- The knowledge that any specific product was left mid-processing within a line over a night, weekend, or holiday period, that may contribute to a quality concern.

Micro-facts yield value when concurrently considered as components of the many software algorithms associated with the different perspectives. If we were to view all micro-facts as pixels in a picture, we would simply see it as static, completely random looking. The knowledge of how the data is linked or related to other data, as well as the understanding of precisely what activities were being performed converts the static into clear, multi-dimensional pictures. This process of contextualization effectively tunes the signal, in the same way as a television set can receive hundreds of video programs as a single signal through a single cable.

Contextualized events are constructed using a combination of many different micro-facts, building sequential, converging and dividing paths of activity, connecting causes, triggers, events and consequences. These contextualized events in turn become contributors of value, and in turn triggers of other events, all combined and linked through a common timeline. Contextualization is therefore analogous to the creation of the sequence of frames that make up multiple simultaneous movies, with each micro-fact representing a pixel used in many of them, all encoded within a single platform—the "cable" that is the singular FactoryLogix platform.

FactoryLogix®

Using the Ontology

Within FactoryLogix, ontology exists across many layers and instances, where software is continuously maintaining a live digital twin of the operation, based on production configurations and product engineering information, planning assignments and production progress as recorded by the micro-facts obtained from the reading of reliable and timely data points. Many actions, displays, events, as well as the selection and display of information to help guide user interactions, are performed continuously as a result of software calculations based on understanding from many perspectives, using a multitude of micro-facts derived from many data sources.

The Need for Data Integrity & Interoperability

A prerequisite for contextualization to be effective is to have access to comprehensive and consistent data. FactoryLogix has many built-in machine, device, and system interfaces, designed to exchange data with all forms of automation across the shopfloor.

Limitations exist, however, according to how each other vendor has structured and developed their native communication mechanisms. There is a wide range of variation as to whether and how each connection is made, and how to correctly interpret the data. Basing contextualization on the best available data creates gaps from other sources that often go unnoticed, leading to incomplete or misleading overall analysis, inconsistencies and inaccurate assessments.

On the other hand, **basing contextualization only on the commonly supported areas reduces the amount of opportunity to use the data collectively,** which varies from factory to factory depending on the equipment set, with ongoing risk that inconsistencies may still appear as equipment is exchanged or is updated.

To address this condition, the Connected Factory Exchange (CFX) committee was formed within the IPC standards organization, which has created a unique standard that defines, in addition to the protocol and coding methods as found with other standards, the precise language that is used to communicate data within IIoT messages. Machines that have been qualified for CFX by IPC communicate on a true "plug and play" basis for all standard functionality, avoiding the need for middleware and data translation, as required when working with other standards.

Having complete and consistent data from each production station enables a richer set of micro-facts to be constructed across all perspectives. Aegis works very closely with IPC, machine vendors, and complementary enterprise solution providers to ensure that their data exchange is as complete as possible for use with FactoryLogix.

IPC CFX

Having complete and consistent data from each production station **enables a richer set of micro-facts to be constructed across all perspectives.**

Security in Manufacturing

Industry 4.0 represents enhanced automation of factory operations using software that works with increasing amounts of data, which introduces greater challenges related to the security of that data. The potential exposure of product design, related engineering and technology information, as well as data that details production configurations, capabilities, deliveries and logistics, represents an increasing concern throughout the industry. An additional benefit from the use of IPC-CFX is the built-in option of encrypting data at source, using TLS 1.2. CFX messages that can then be exchanged locally, as well as using a hybrid-cloud configuration, with the knowledge that such data cannot be intercepted by a third party.

Lack of Contextualization "Gotcha" Examples

There are several "gotchas" that immediately spring to mind where failures of information integrity and interpretation arise, which occurs surprisingly often in complex commercial solutions where contextualization has not been done correctly:

Machine Maintenance

Product completions are recorded during setup and maintenance. Sensors are triggered as setup is performed, resulting in incorrect product counts, incorrect material consumption, sometimes even triggering un-needed replenishments. There are then further concerns with the compromise of individual product manufacturing history (traceability). It is necessary to know the operational mode of the machine in order to interpret messages correctly in the right context.

Station Rework

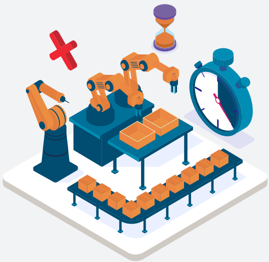
Stations are performing rework, which should normally be accounted on the factory level as non-added-value work when accounting correctly for time, work, and materials in OEE reports.

Lunch Breaks

An operator-response related issue is reported due to a longer than usual machine stoppage, when the operator was in fact, at lunch.

Overtime

Productivity levels are reported that are greater than 100%, when unplanned overtime is worked without proper consideration of the live working calendar.



Stoppages

An irate manager complains of stoppages on an automated process that require time for materials to be set up and verified.

An Example of the Contextualization Process

Let's consider a simple production line, comprising three production stations, which in principle can each be either manual or automated processes. In the case of automated stations, an interface is used to collect data through IIoT in real-time, whereas in the case of manual stations, the data is derived from the FactoryLogix Operator Cockpit. To keep this example simple, we will consider them behaving in identical ways.

Each station possesses a specific sphere of awareness, its own internal digital twin, where measurements and records of events related to operations are accumulated, stored and shared, using messages. Routine events, as part of the station's operation, make up the vast majority of the information. This is useful for display, but is rarely of additional value unless some form of exception happens, such as unexplained line slowness. Exceptions, like the failure of the pickup of a component, are also recorded as events, which hold clues that will be used in the assessment of root cause, action needed, and consequence assessment. In the vast majority of cases however, exceptions are caused by events that are external to the production station.

A production report based only on the data from a single station represents just around 20% of the potential actionable value. Simply transferring the internal data, by some mechanism, to any solution, cloud-based or otherwise, unless it is immediately contextualized, does not improve this statistic. **Understanding the relationship between events that individual stations report, considering what their work assignment is at that instant, and how their operations have been interdependent, brings opportunity for additional perspectives to be considered, significantly increasing the value from the individual data.**

Within FactoryLogix, a digital twin of the three-station configuration exists. The same internally-focused data from each of the stations, is received, sent in real-time using IIoT-based messages as the stations operate. **Such information, for illustration in this example, includes:**

Operational / non-operational time and state

Production units completed

Starved input / blocked output to the station

Operational error / issue occurred (e.g. breakdown)

Material related erroroccurred

Operation was stopped by operator for some reason

Within the FactoryLogix digital twin, the messages are set into context with each other, as well as knowledge of other factors, including:

Station configuration, specification and capability

Product routing rules between stations for this product

Product information, material requirements and work assignment

Materials setup, based on product requirements and work assignments

In addition, there is other applicable information, derived from the factory operational level, including:

- **Factory-level material availability and logistics**
- **Tools, feeders, and other equipment on which there is a dependency**
- **Maintenance plans and other exceptions, such as power or air supply**
- **Status and limitations of production unit preparation ready for these stations**
- **Status and limitations of product unit transfers following these stations**
- **Sequence of following production plans for these stations**
- **Working calendar, including overtime, operator breaks, etc.**

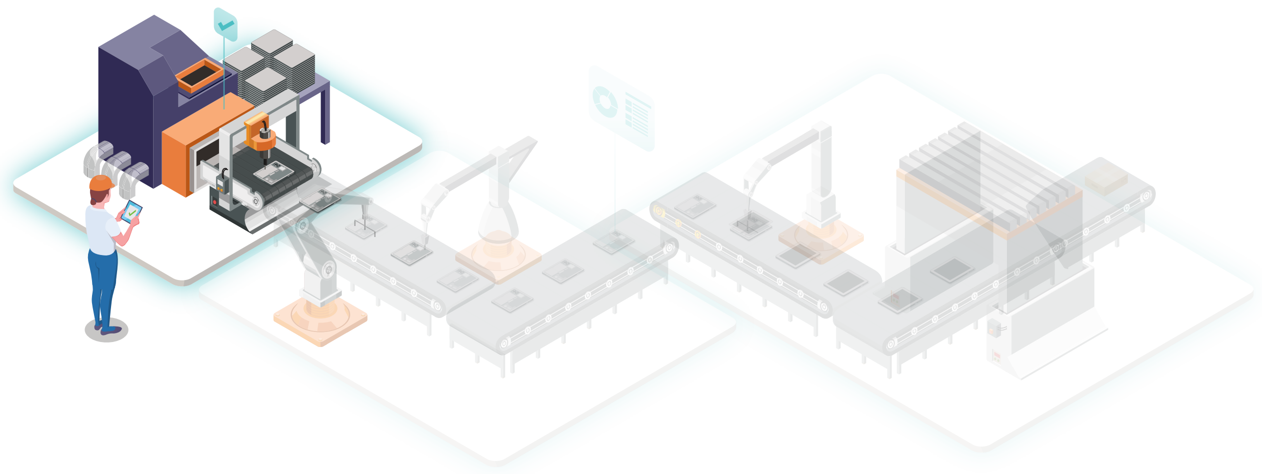
1

Contextualization Example: Stage 1 – Significance

In this example, we follow a story of the contextualization of an event that affects our three-station line. The purpose is to illustrate the activity process of an "AI" that is responding to a potential exception in the operation. At first, we assume that the initial running condition of the configuration is fine.

The first question is what do we mean by "fine"? **There are a lot of events that disturb the manufacturing flow where notifications or actions are not required, such as the replacement, setup and verification of materials within the expected time for such events to happen.** A trigger condition for an exception therefore has to be defined. In this example, we use the trigger as being the risk that there is a trend that suggests there will be a starvation of sub-assembly delivery from our three-station configuration to a main assembly line. It is imperative that main assembly is never disturbed. If any sub-assembly is late, the main line may exhaust a local stock, which can be quite small in a Lean operation.

In addition to the significant effect of a main line stop, other sub-assembly manufacturing stations may need to be stopped in order to avoid excess build-up of other sub-assemblies, until the main line is moving again. **Part of the data behind the trigger is that completions on our configuration have stopped,** even though the maximum intermediate main line inventory is below its minimum threshold.



2 Contextualization Example: Stage 2 – Determine Root Cause

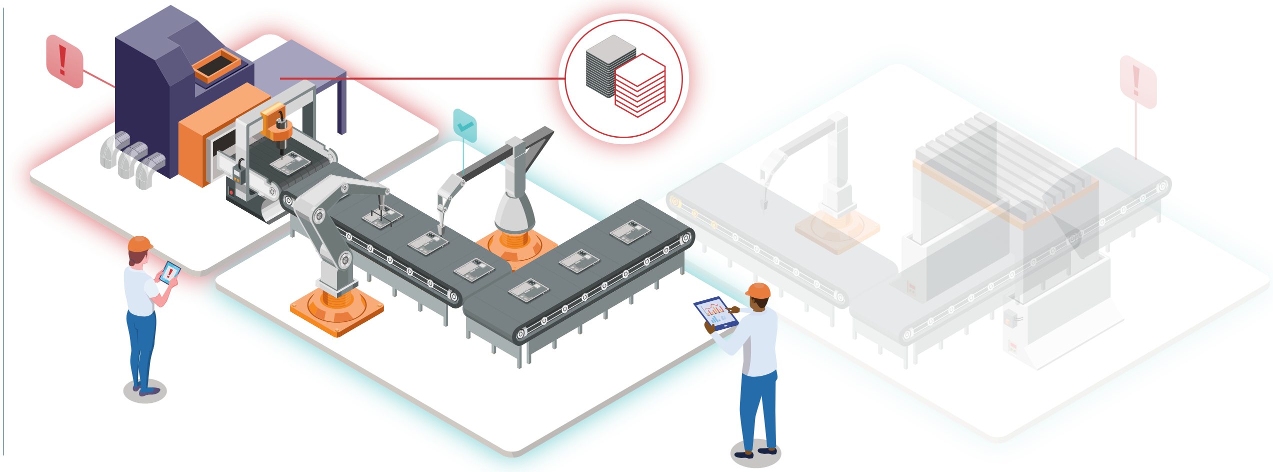
The third and final station in the line has reported that it is stopped. The AI knows that the scheduled work order is still active, that the station is still in production mode, but has not received a production unit from the preceding second station.

If the AI sees that the second station is running, then it would attempt to find out whether there is an issue affecting the general performance of the configuration. **Immediate cycle times of each station would be analyzed in case there is an unexpected bottleneck.** In this case however, the second machine is also in an operational condition, but has not received a production unit for some time from the first station in the line.

The first station has reported a stop condition, which is related to a specific material that it was expecting to use, but was not present. A production unit is waiting for completion, but without the material, the station is unable to proceed.

The AI will look at the status of the missing material to discover, for example, whether the material has been exhausted. If the material had only been partly used, the problem could be that there was an issue with the setting or feeding of the material on to the station, and perhaps an operator had not responded where action was needed. In this example however, the material had been exhausted, but no replenishment material had been logged as having arrived at the station. The AI then turns to the material logistics area to find out why there was no material ready. The discovery is that there had been a material prepared, but the material had been found to have been damaged so could not be used.

This had been the case for some time; sufficient time in which usually the next available material would have been set up. The warehouse area is then questioned, and the AI finds that there is no further availability of the specific material, and that an alternate material is being assessed for use in place of the original, but this requires engineering signoff. As it is the evening shift, the engineering manager is not available to authorize this change of material, so the process is "stuck." The person in charge has **failed to take the appropriate action to escalate this incident,** as they did not think that the seriousness of the situation demanded it.

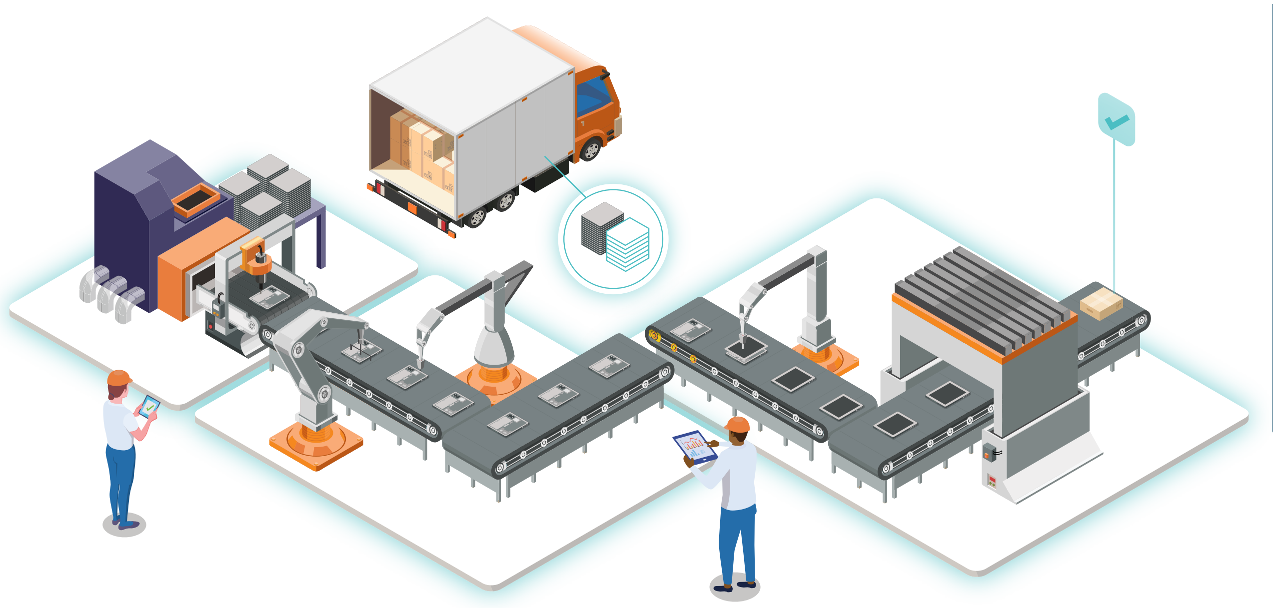


3 Contextualization Example: Stage 3 – Action

With the AI seeing the entire chain of past events, the present status, and extrapolated consequences into the future, the **justification to call an engineering manager can be immediately made.** Rather than having the manager come to the factory, remote access allows the AI to provide information about the original material and alternative suggested, which may not have been on the approved vendor list as the material is of greater quality or cost than required. Information about physical attributes of each material is also shared. The manager can then be sure that allowing this part to be used will not affect other production orders, nor impact the quality of the product, or performance of production. Even though the suggested alternative material is more expensive, the cost saving in avoiding any main assembly line disruption makes sense. The substitution is approved.

In this example, **AI and humans work together to resolve the issue,** based on many different pieces of information gathered from machines about the product and the factory operational condition. In the future, the AI itself may be able to make more and more of these decisions, based on a judgement using the same factors as were considered by the human manager. The reason that a human is performing the decision-making today is because the decision is complex, for which responsibility needs to be taken.

Though illustrating the AI contextualization "thought process," this example is a little unrealistic in the real world, as mini-triggers would already have been created within the logistics area, the warehouse area, and around the production stations long before there was a threat to the main line production. Each mini-trigger contributes to the overall significance of action being needed, driving the escalation, with **each algorithm providing context from each perspective.**



The Relationship Between Contextualization and Provenance

The use and benefit of contextualized data goes far beyond uses related to Industry 4.0 and the live manufacturing operation. Provenance within manufacturing, covering materials, processing and quality, is an essential element of modern, secure, and trusted manufacturing. Provenance can only be reliably created by associating contextualized relevant information with discrete physical entities. Overall manufacturing provenance breaks down to include:

MATERIAL PROVENANCE

The record of the exact materials used in a product is an essential element in Material Provenance throughout the supply chain. The physical material used against each design-driven component may differ, even during a single production lot. This can be due to, for example, a change in the lot or batch of parts in use from a single manufacturer, the use of parts with the same specification but from different manufacturers, or the use of alternative parts as allowed by the Approved Vendor List (AVL). Information related to the performance of any aspect of materials must be contextualized against the actual material used, so as to differentiate performance in terms of quality and reliability, as well as cost. Material Provenance within manufacturing is a key contributor to the overall holistic supply chain product provenance, which in many cases may be subject to environmental and ethical-sourcing control, as well as location-based taxation.

PROCESS PROVENANCE

This is the assurance that manufacturing processes have been operated according to specified procedures, which include the sequence and setup of processes, recording of operational measurements and results, and in many cases, operator credentials and environmental data. Process provenance also includes the assurance that in the case of any deviations occurring from the specified sequence of operations, such that additional inspection, test or repair is needed, the specified recovery rules are followed. Facts that form the complete Process Provenance data-set associated with each product manufactured, are contextualized from the many different micro-facts and associated perspectives, which provide the assurance of operational compliance without the need for manual checks and reports.

QUALITY ASSURANCE & TRACEABILITY

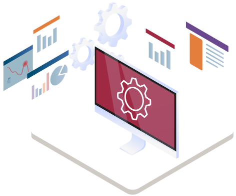
Beyond the confirmation of operations, traceability data, which includes that of Material and Process Provenance, exists as a superset of contextualized facts. This typically related to a group of products made and details exact manufacturing conditions of every product. In the event of a defect occurring in one product within a work-order or lot, full traceability allows the discovery of the exact unique circumstance that led to the defect occurring, and will also indicate any products made which may have a related weakness. In order to use traceability data for quality control to its full potential, the exact context needs to be in place so as to understand the precise material, operation, process, event sequence, etc.

Applications of Digital Twins

As we saw in the above example, the application of the term "digital twin" is associated with complex scenarios for which software is specifically developed, where the scope has evolved to become far greater than trivial. **There are four basic types of digital twin applications:**

VISUAL

The focus within the data is the description of a physical entity, for example a car, or a circuit board. Shapes, materials, movement, finishes, including those of a multitude of individual components, make up the final model. The objective is to provide a visual human interface, creating a format whereby the human mind can instantly see the results of ideas, and perform changes that further enhance a process while it is being performed.



COMPUTATIONAL

The simplest form of a digital twin is a piece of software that processes data in order to find a solution. A good example of this is software that creates the sequence of instructions for an automated machine or robot. The components of this digital twin usually consist of the product and material data, a representation of machine operational mechanisms, a timing model, and in advanced cases, data associated with other operations in the line, as well as consideration of multiple product variants that are to be made as part of a group.

OPERATIONAL

In this digital twin, software is used to make real-time decisions as to how to control an operation. An analogy is air-traffic control, where accurate positioning and speed measurements are digitalized from radar as well as taken from the aircraft itself. Information about each flight is known, as is the capacity of the runways and taxi areas at the airport. The job is to get all of the planes to where they need to go quickly and safely. As with an assembly manufacturing operation, air-traffic control has been gradually automated, an evolving digital twin, but the final oversight remains with the human. In assembly production, we also see an increasing number of decisions becoming automated, for example, the pull of the next material replenishment to the machine immediately prior to material exhaust, taking the opportunity of an unrelated down-time event to perform a maintenance task, and more.

ANALYTICAL

Enabled by the sharp increase in the volume and range of data being collected, analytics provide the ability to find patterns in the data, which are associated to a particular outcome. Outside of the digital twin, analytics will select, sort, filter, cross-reference, pivot, and summarize complex datasets, such that patterns and trends are exposed. Many analyses are pre-determined based on known KPIs. In the digital twin world, software contextualizes the meaning of the combined data, knowing the operation of each station. Should, for example, a defect at an assembly test station be noticed, the digital twin will compare the results of all prior operations across all similar products, discovering what were the unique factors that contributed to this defect. The software is then able to not only identify other products where the factors were similarly close to creating a defect, but can also monitor future assembly, recognizing trends towards those patterns that have caused defects, adjusting parameters that compensate for the trend, and automatically avoiding further such defects being created.

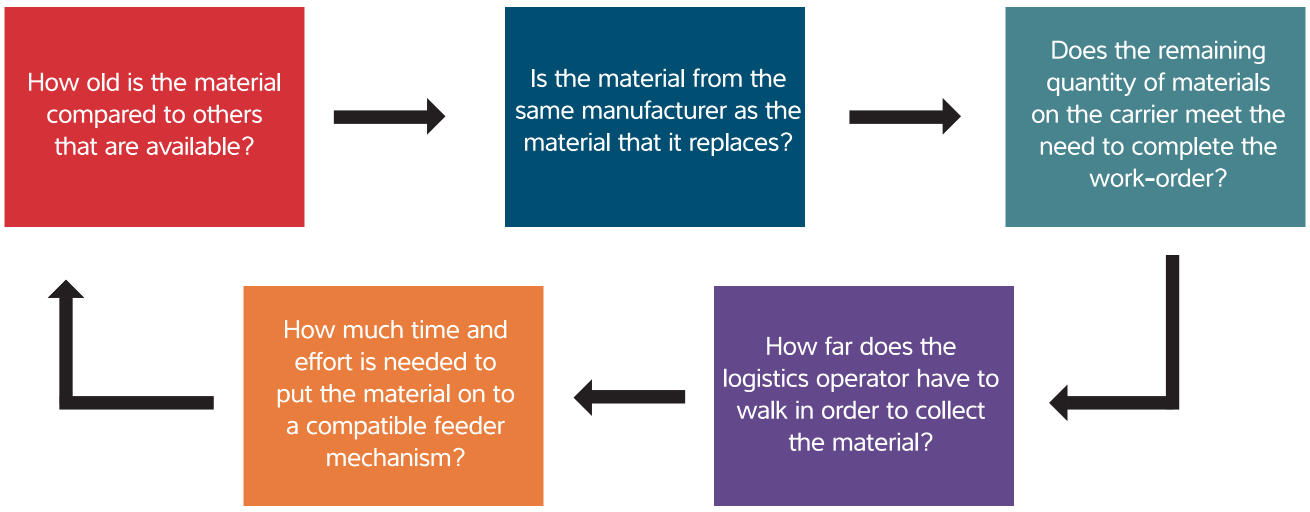
Artificial Intelligence Algorithms

As the availability and interoperability of data is one key component of the digital twin, the other is the software that contains the algorithms, or intelligence, behind calculations and decisions that are made.

The majority of common software algorithms are heuristic, following a set of rules that define how data is combined to reach an outcome. The challenge with these algorithms when used for complex problems is that a great deal must be known or created by the programmer about how to find the result, which requires a deep understanding of the problem itself, as well as assumptions about how it should be solved. In this way, we can say that the resultant digital twin based on these algorithms is intelligent in that it includes knowledge of how to find the solution. **The challenge however is that the algorithm is dependent on the capability and experience of the programmer, and so may not represent the best solution.** Significant reengineering of the algorithm is also necessary should there be any significant change in the method needed. A key benefit of heuristic algorithms, however, is that results are obtained very quickly as compared with other forms of algorithms. Digital Twins that need to react or make decisions quickly, based on relatively simple rules, will include many examples of these rather traditional heuristic algorithms.

Another type of algorithm is referred to as "fuzzy logic." In this case, the software splits the problem into its logical factors, calculates a result for each, and then uses additional logic, based on the relative importance and interactive effect of each factor, in order to find the overall solution. One example for the use of this type of algorithm is the selection of a replenishment SMT material reel from the warehouse. **Individual factors would include:**

- **How much time is there to have this material delivered?**
- **For each instance of available material that meets the operational requirement:**



How old is the material compared to others that are available?

Is the material from the same manufacturer as the material that it replaces?

Does the remaining quantity of materials on the carrier meet the need to complete the work-order?

How far does the logistics operator have to walk in order to collect the material?

How much time and effort is needed to put the material on to a compatible feeder mechanism?

The fuzzy logic algorithm avoids potential problems that simplistic fixed logic would have. For example, if the "FiFo" (First-In, First-Out) rule were applied in isolation, there would be the risk that the selected material carrier had too few components remaining and so would require an additional needless replenishment, or perhaps that too much time would be needed to move the material from one feeder type to another, missing the delivery deadline.

Fuzzy logic algorithms assess and balance all factors based on the context of the situation, involving quite complex interactive calculations, often yielding results that may be unexpected from the point of view of initial expectation, hence the name "fuzzy" logic. As the relative priorities for each factor can be altered through the use of parameters, and new factors added or removed over time, fuzzy logic algorithms are a very important and flexible tool within the modern AI, even though they have been used in software development for decades. As more data becomes available through extended interoperability, it is relatively easy to expand the "intelligence" of the fuzzy-logic-based solution.

Genetic Algorithms (GA) are another, more modern method of finding solutions, with many variations and derivations in how they work. In theory, genetic algorithms are able to find the best solutions to problems, without dependency on assumptions from a heuristic perspective. An example of the use of genetic algorithms would be the assignment of manual assembly work to a line of operators. Rules governing the sequence of assembly would be defined against which a score would be calculated. Such rules may include, sequential operation dependencies, visibility and access for the operator, the avoidance of similar materials within the same cell, etc.

The goal is to assign the work according to the requirements and rules, while also balancing the work between the operators so that there is no wasted time between them. The genetic algorithm starts by assigning all work items at random between the operators, assessing the result then altering the random assignment, creating another sequence to be assessed, and the process repeats. The best assessment is taken as a potential best solution. There are various differing methods for the splitting and recombining of the sequence, with the original method following the same premise as the genetic inheritance of DNA from parents to their child. The challenge with genetic algorithms is that they can take a considerable time and computing power to process, such that a defined time-limit or period of no improvement must be specified so that the algorithm comes to an end. With the creation of quantum computing, which is orientated towards genetic algorithmic problems, these results could be sped up immensely in the future, widening the scope of application.

> As more data becomes available through extended interoperability, **it's relatively easy to expand the 'intelligence' of the fuzzy-logic-based solution.**

In the cases of the fuzzy-logic and genetic algorithms, there remains the challenge that the assessment ruleset is in fact still based on the engineering knowledge of the task. There is the potential to allow AI to alter the way in which the scoring is done, perhaps even allowing the discovery of new parameters to monitor and score. This is essentially, at a very basic level, how humans learn, driven by motivation. In the manufacturing world, we would teach the AI what is desired and what should be avoided throughout the whole manufacturing operation in order to trigger the process of determining what works and what does not. **Algorithms and computing technologies themselves, such as neural networks and quantum computing, are evolving alongside the development of this aspect of AI.**

Conclusion: Putting Analytics Up Front is Essential

Contextualization is at the root of intelligence. Within FactoryLogix, there are many simultaneous and intertwining contextualization processes going on in real-time. As manufacturing executes, more jobs are allocated to the work-load, and new patterns are discovered and analyzed in historical data.

Without contextualization—as is the case in many cloud-based manufacturing analytics solutions, where raw data from production stations has to be reverse engineered to find the significance and relationships from literally millions of discrete events each time—calculations need to be repeated over and over again in just slightly different ways.

With contextualization, micro-facts are determined in real-time, while information is "fresh" and meaningful. Micro-facts are considered from many different perspectives, used in combination to identify actions, recommendations, and measurements of performance. Historical contextualized data is meaningful. AI algorithms can be formulated to make complex decisions in the merest fraction of the time that would otherwise be needed, with the simultaneous processing of many routine factors and extraordinary condition triggers, presenting complex situations in the form of digital twins.

Industry 4.0 continues to evolve away from isolated app-based "point solutions" that struggle to perform in complex situations, towards the holistic approach of intelligent software-driven manufacturing.

The unique FactoryLogix single IIoT-based platform is the perfect environment in which AIs can provide value for manufacturing today, and increasingly in to the future.

Learn More

AEGIS
SOFTWARE

Email: info@aiscorp.com
Visit: www.aiscorp.com
@FactoryLogix
linkedin.com/company/aegis-industrial-software

Corporate Headquarters
220 Gibraltar Road, Suite 300
Horsham, PA 19044
Phone: +1.215.773.3571

European Headquarters
Wetterkreuz 27
91058 Erlangen, Germany
Phone: +49.9131.7778.10

Asia Headquarters
Rm. 809, Dahua Hucheng Business Center
No 6, Lane 239, Dahua No. 1 Road
Putuo District, Shanghai, 200442, P.R. China
Phone: +86 2 1 5882 4882