

Engineering Works Scheduling for Hong Kong's Rail Network

Andy Hon Wai CHUN

Chief Information Officer
City University of Hong Kong,
Tat Chee Avenue, Kowloon Tong,
Hong Kong SAR
andy.chun@cityu.edu.hk

Ted Yiu Tat SUEN

Chief Information Officer
MTR Corporation Limited
Telford Plaza, Kowloon Bay,
Hong Kong SAR
tedsuen@mtr.com.hk

Abstract

This paper describes how AI is used to plan, schedule, and optimize nightly engineering works for both the commuter and rapid transit lines in Hong Kong. The MTR Corporation Limited operates and manages all the rail lines in Hong Kong. Its “Engineering Works and Traffic Information Management System” (ETMS) is a mission critical system that manages all information related to engineering works and their related track possessions and engineering train movements. The AI Engine described in this paper is a component of this ETMS. In Hong Kong, the maintenance, inspection, repair, or installation works along the rail lines are done during the very short non-traffic hours (NTH) of roughly 4 to 5 hours each night. These engineering works can be along the running tracks, track-side, tunnel, freight yards, sub-depots, depot maintenance tracks, etc. The proper scheduling of necessary engineering works is crucial to maintaining a reliable and safe train service during normal hours. The AI Engine optimizes resource allocation to maximize the number of engineering works that can be performed, while ensuring all safety, environment, and operational rules and constraints are met. The work described is part of a project to redesign and replace the existing ETMS, deployed in 2004, with an updated technology platform and modern IT architecture, to provide a more robust and scalable system that potentially can be deployed to other cities around the world.

Introduction

Hong Kong's rail service began in 1910 with the “KCR British Section,” which is now renamed the East Rail Line. It was operated by the Kowloon-Canton Railway Corporation (KCRC). Since then, the KCRC network expanded to include the West Rail Line and the Ma On

Shan Line as well as the Light Rail. In 2007, the KCRC merged with the MTR Corporation Limited (MTR), which ran the 6 urban rapid transit lines (Disneyland Resort Line, Island Line, Kwun Tong Line, Tseung Kwan O Line, Tsuen Wan Line, and the Tung Chung Line) and the Airport Express. (Kowloon–Canton Railway Wikipedia entry 2014, MTR homepage 2014)

After the merger, MTR now operates 10 lines (6 urban rapid transit lines, 3 commuter lines, and the Airport Express) and the Light Rail, consisting of 218.2 km (135.6 mi) of rail with 152 stations, including 84 railway stations and 68 light rail stops. (MTR Wikipedia entry 2014)



Figure 1. Map of Hong Kong's rail network.
(public domain map by Sameboat:

http://en.wikipedia.org/wiki/File:Hong_Kong_Railway_Route_Map_en.svg)

Despite being the 10th busiest subway system in the world, MTR has consistently delivered a 99.9% on-time rate while carrying over 5 million passengers on an average weekday. The passenger volume is roughly similar to the New York City, Mexico City, London, and Paris systems. Besides the on-time rate, MTR farebox recovery ratio (the percentage of operational costs covered by fares) is also the highest in the world at 185%, making it one of the most profitable systems (Padukone 2013).

Besides Hong Kong, MTR also runs individual subway lines in Beijing, Hangzhou, and Shenzhen in the Mainland of China, as well as the entire London Overground, Melbourne and Stockholm systems. MTR aims at becoming the biggest operator of metro systems in the world, sharing its management and operational best practices to other cities. For example, after MTR operated the London Overground, the on-time performance rate improved from 88.4% in 2007 to the current 96.7%. Similarly the Melbourne Metro rate rose from 84.6% in 2009 to the current 93.7% (Ng 2013).



Figure 2. Engineering work with an engineering train

Business Needs

In order for MTR to maintain its performance and quality of service, all the lines must be well-maintained. Each week over 2,600 engineering works need to be performed, with over 200 requiring possession of track segments. In total, over 10,000 maintenance personnel are involved. Deciding which engineering work should be done and when is a complex decision-making process, involving safety factors, resource availability, environmental considerations, and operational needs. Since engineering works can only be done within a short non-traffic hours (NTH) window of roughly 4 to 5 hours each night, it is crucial that engineering works are scheduled effectively and optimally. Previously, the scheduling of engineering works was a manual process that required many domain experts across different departments to meet face-to-face and collaborate together to work out a detailed action plan.

This was extremely time-consuming and prone to potential human oversight.

In 2002, MTR implemented an “Engineering Works and Traffic Information Management System” (ETMS) to manage the planning and execution of engineering works, the system outage and switching arrangement, and the publication of traffic notice that provides graphical presentation of railway network and the engineering works. In 2004, MTR commissioned the City University of Hong Kong (CityU) to create an AI Engine to enhance the ETMS with AI capabilities to support planning, automated scheduling and iterative rescheduling. Deployed in the same year, the AI Engine has been serving the urban rapid transit lines for close to a decade (Chun 2004, 2005).

After the 2007 merger with KCRC, MTR began to consolidate and unify the operational practices of rail lines in Hong Kong. In particular, it defined a common set of “Railway Safety Rules” with relevant guidelines and procedures for engineering works to be shared between the urban rapid transit lines and the KCRC commuter lines. At that time, KCRC had two separate IT systems to manage engineering works for the commuter lines and light rail.

In 2010, MTR decided to consolidate its 3 engineering works IT systems to create a single common one to manage all engineering works for all the lines it operates in Hong Kong. The tender for the new ETMS was awarded in 2011 to PCCW Solutions Limited (PCCW homepage 2014), with CityU (CityU homepage 2014) contracted once again to provide the AI technology.

Project Challenges

This project was challenging at many fronts. Firstly, the task of deciding which engineering works should be done and when is highly knowledge intensive. For example, certain types of engineering work may require work to be done on or near tracks. For those cases, segments of the track will need to be assigned to a specific engineering work and cannot overlap with others. On the other hand, some engineering works may require power to be on so that engineering trains can operate, while others require power to be off for safety of the workers. Engineering works may require a range of supporting resources, such as engineering wagons/locomotives and train operators. For works at exposed/open sections of above-ground running lines, if they generate noise, the total noise level within that area has to be controlled. There is also a long list of complex safety rules and government statutory requirements that must be strictly followed.

The 2011 AI Engine redesign project offered several new challenges not faced by the original 2004 implementation. In 2004, the AI Engine only needed to handle rules and constraints of the subway lines and the Airport Express. The 2011 redesign required the AI Engine

to also handle commuter lines, which are quite different in how they are operated and maintained.

Firstly, the new AI Engine will be required to manage/allocate the overhead power lines as well. For the subway lines, power is normally off during NTH and even if they are powered, the track possession area is identical to the power zone. However, for commuter lines, power is normally on and overhead power lines have different power zones than possession areas. Enhancements were needed to handle these differences, such as a new model to represent the overhead lines (OHL), algorithms to allocate sections of the power lines to different engineering works, and extending our rules to identify OHL conflicts.

Secondly, related to power, are outage documents/permits. If work is to be performed on or near OHL along the commuter rail, certain documentation will be needed, such as “Permit-To-Work” (PTW) and “Isolation Record Form” (IRF). The AI Engine will need to keep track of these permits and requests to ensure adequate level of resources are available to manage the power lines.

Thirdly, with growing complexity of the rail network, there will be multiple paths for engineering trains to travel. The new AI Engine will need to be extended to check for any potential conflict those train paths might have with other engineering works that are assigned.

Fourthly, besides scheduling engineering works and allocating necessary resources, the new AI Engine will also need to perform outstabling – the process of assigning a parking location for a passenger train to park overnight. The location assigned must not conflict with any other engineering works that might be performed in that vicinity.

Last, but not least, the AI Engine must be designed in such a way that allows MTR to easily adapt to rail operations in other cities around the world.

What AI is important?

The AI Engine is important because it provides many crucial capabilities to support MTR’s continued business growth:

- **Knowledge retention** – It takes years of experience to acquire all the necessary knowledge to perform engineering works scheduling effectively. Besides, human planners usually specialize in only one of many disciplines within railway operations. The use of AI ensures valuable knowledge, across rail functions, is retained and easily accessible by anyone using ETMS.
- **Growth** – Hong Kong’s rail network is still growing rapidly with construction on 5 rail extensions that will increase route length by over 25% to 273.8km by 2020. The AI Engine is totally data-driven, allowing new stations, new lines, new resources, new rules, etc. to be added without source code change. This gives MTR greater agility to support its continued growth.

- **Scalability** – The AI Engine is designed to be a highly scalable cloud service that provides AI scheduling as REST services (Costello 2010, Rodriguez 2010). The AI Engine is also rail line agnostic, i.e. there are no MTR-specific knowledge hardcoded into the engine. The combination of these two design elements allows MTR to conveniently scale the new system to support other rail lines that MTR operates in other cities around the world.
- **Compliance** – The most important aspect of an engineering works schedule is that it must absolutely comply with all safety rules and regulations as well as government statutory requirements. The AI Engine considers all relevant regulations as hard constraints and ensures all approved engineering works to be compliant.
- **Planning Productivity** – ETMS is used by over 3,000 people, both internal MTR staff as well as external contractors. They submit engineering work requests via Web-based forms. Prior to using AI, people submitting requests had to wait for the weekly meeting of planners to approve the engineering works. If not approved, they will have to delay the work and resubmit. With AI, conflict checking is done at the time engineering works are submitted, giving them instant feedback so that they can adjust their work plans if needed. This can save weeks of waiting time.
- **Scheduling Productivity** – Prior to using AI, generating a weekly schedule required a team of planners from various departments to sit down and work out a schedule. This is a lengthy negotiation process with numerous on-the-fly calculations and juggling of tasks. All rules and constraints had to be assessed in the minds of the individual planners. With AI, the schedule is automatically generated prior to the meeting. The weekly meeting is now used purely to vet the AI results for final approval, thus saving tremendous amount of valuable time.
- **Agility** – The AI Engine generates weekly schedules 3 weeks in advance so that there will be time to prepare. Between when the schedule is generated and execution, there will be changes, such as emergency works, unexpected problems, work cancellations, etc. AI is used to “interactively reschedule” engineering works as needed without disrupting the existing schedule. The AI Engine checks all rules and regulations, including resource availability, etc. whenever there is a change. This ability gives MTR agility in handling any unexpected engineering needs.
- **Saves Costs** – The AI Engine’s optimization algorithm maximizes resource utilization, so that more work can be done with existing resources. The AI Engine does this by combining nearby engineering

works together to share resources/equipments if they do not cause any conflicts.

- **Customer satisfaction** – Ultimately, the use of AI is to improve customer satisfaction. A well-maintained line is less prone to problems and failures that cause delays. Because the AI Engine improves planning productivity and maximizes the number of engineering works that can be done, more engineering works can be performed and as early as possible.

Application Description

The ETMS application is a web-based system that manages all engineering works at MTR. It is used during planning, scheduling, interactive rescheduling, and actual execution of engineering works. It also manages resources, such as availability and locations, as well as engineering train paths, and system outage and switching. The system is also used to produce “traffic notices” that contain graphical diagrams of railway network and the engineering work locations, as well as any corresponding safety arrangements.

AI is used during all phases. The planning phase is when details of each proposed engineering works are entered into ETMS. AI is used to check if there are any conflict between the newly entered engineering work and any other pre-approved works. The scheduling phase is when AI is used to automatically generate a weekly schedule of engineering works to be performed. AI ensures there are no hard rule violations and all the necessary resources are available. It also optimizes resource and track possession utilization, so that a maximal number of engineering works will be scheduled. The interactive rescheduling phase is used to modify a weekly schedule after it has been officially approved. This gives MTR agility to add last minute jobs or make amendments to engineering work details prior to execution. AI will check for potential conflicts and resource availability with each change.

System Architecture

AI capability is provided by the AI Engine, which is designed as a highly scalable cloud service that is decoupled from the ETMS client application. The ETMS application access the AI Engine service through a set of REST APIs. This design potentially allows AI to be accessed by other applications. We call this “AI as a Service” (AaaS). Figure 2 is a high-level diagram of the ETMS system architecture.

Users interact with ETMS through web browsers which connect to the ETMS Web Server that passes requests to the ETMS Application Server (Fig. 3). The ETMS Application Server stores all ETMS-related data in the ETMS Database Server. This part of ETMS was coded

using a Java platform and is also architected for high-availability. It interacts with the AI Engine by making REST web service calls to a separate AI Engine system.

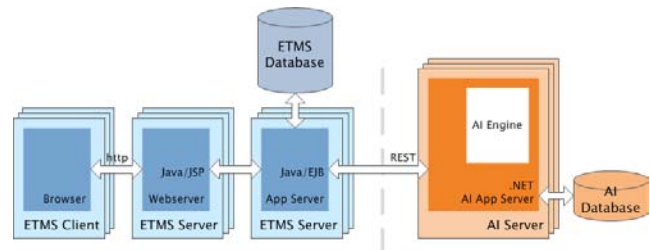


Figure 3. System architecture of the ETMS application.

The AI Engine was built using Microsoft .NET technology using AI tools/libraries that we have designed and coded over the past couple of decades of AI development work in Hong Kong. For high-availability and resilience, we used Microsoft Windows Clustering with Network Load Balancing (NLB), providing a scalable configuration of AI Web and application servers. The AI database acts as an object-persistence cache for data from the ETMS database to improve performance.

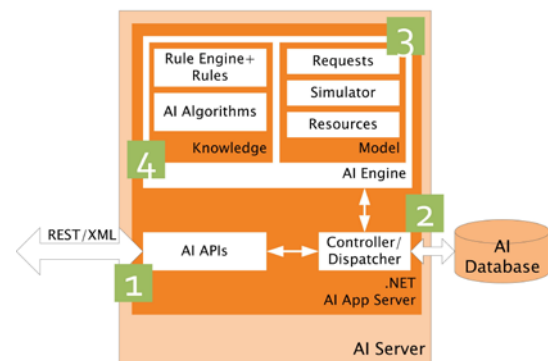


Figure 4. Detailed architecture of the AI server.

Figure 4 shows details of the AI server architecture, which basically follows an MVC design pattern (Burbeck 1992). The numbered items (i.e. 1, 2, 3, 4) in Fig. 4 are explained below:

1. **REST/XML + AI APIs** - The AI Engine provides a comprehensive set of 25 authenticated REST service calls. This includes APIs for administrative support and database support (such as adding/ removing AI rules, rail network structure, resource types, resource locations, inventories, and various code tables). For planning, there are APIs to validate (i.e. perform AI rule checking) a single proposed/granted engineering work or all proposed/granted engineering works. For scheduling, there APIs to generate optimized weekly engineering work plans. For interactive rescheduling, there are APIs to interactively add a new engineering work, cancel/ amend an existing one, combine two

existing engineering works into one, split an engineering work into two, grant/revoke an engineering work, etc. All data exchanges are done through XML. AI Engine results are delivered back to the ETMS client as XML and then displayed as the “View” in the MVC pattern.

2. **Controller/Dispatcher + AI Database** – Based on the particular Web service call, the controller/dispatcher will issue related commands to either manipulate the AI model, or call AI algorithms to process the model. This is equivalent to the “Controller” in the MVC pattern. If needed, data for AI processing is automatically loaded/saved from/to the AI object-persistent database.
3. **AI Model** – The AI Model is an in-memory object-cache of the AI database. It represents a domain model of the MTR rail network environment. The model contains relevant engineering work requests, resources (such as equipment and personnel and related information), and a “simulator” that understands rail network structure and is able to traverse and reason with the network model. This is part of the “Model” in the MVC pattern. The remaining part consists of the business logic encoded in the “AI Knowledge” component.
4. **AI Knowledge** – The AI Knowledge component consists of a rule engine and a set of rules to encode the various rules and constraints, such as safety regulations, operational guidelines, and Government statutory requirements. The rules are loaded from the AI database and can be changed any time via REST APIs. Besides the rule engine, the system provides two other key AI algorithms, including the AI Scheduling Algorithm (explained in the next section) and an Outstabling Algorithm (Chun 2013).

Uses of AI Technology

As outlined previously, AI is used during planning, scheduling, and interactive rescheduling. The following highlights the different AI techniques used in each.

The Planning Phase

This is when proposed engineering works are first entered into ETMS via web-based forms. After the proposed engineering work has been entered, the AI Engine performs conflict checking to ensure the proposed work does not conflict with other proposed or approved engineering works. The conflict checking is performed with an AI rule engine (Chun 2005) that processes rules related to resource allocation, train formations, safety regulations, operational guidelines, Government statutory

requirements, etc. Unlike traditional rules that operate on a working memory, our rules operate on top of the AI Model (described in the previous Section). The AI Model contains objects that represent all the elements in a real world rail system, e.g. the lines, tracks, stations, landmarks, depots, power lines, trains, personnel, equipment, resources, etc. The AI Model and rules are passed to the AI Engine as XML objects through the REST service calls. This allows users to modify the proposed engineering work at the time of data entry if there is any rule violation so that it will have a better chance of getting approved and resource allocated during the Scheduling phase. For example, possibly changing the desired work day or days to other days when there is no conflict with other approved works.

The Scheduling Phase

During the scheduling phase, the AI Engine automatically generates a weekly schedule of engineering works to be performed. We modeled this as a constraint-satisfaction problem (CSP) (Cohen 1990, Van hentenryck 1989, Steele 1980, Kumar 1992, Tsang 1993). In CSP, a solution (or schedule) is found through the assignment of values to variables subjected to a set of constraints. Each engineering work request is represented by at least two variables – to approve the request or not, and whether it should be combined or not with another request. Constraints are coded as rules and validated with the rule engine (Apt 2001).

More formally, a CSP can be defined as consisting of a finite set of n variables v_1, v_2, \dots, v_n , a set of domains d_1, d_2, \dots, d_n , and a set of constraint relations c_1, c_2, \dots, c_m . Each d_i defines a finite set of values (or labels/solutions) that variable v_i may be assigned. A constraint c_j specifies the consistent or inconsistent choices among variables and is defined as a subset of the Cartesian product: $c_j \subseteq d_1 \times d_2 \times \dots \times d_n$. The goal of a CSP algorithm is to find one tuple from $d_1 \times d_2 \times \dots \times d_n$ such that n assignments of values to variables satisfy all constraints simultaneously.

When the engineering works scheduling problem is formulated as a CSP, we have a variable v_n to represent whether or not a request should be assigned, and another variable v_m to represent whether or not the request should be combined. In this case, the potential values of the variables are Boolean. For requests with “X of Y” constraints, i.e. requiring X days out of Y days, there are X variables $v_{m1}, v_{m2}, \dots, v_{mx}$ to represent which days to assign the request. The domain, or potential values, of those variables are the range of days given by Y.

The CSP constraints, c_j , are the rules, constraints, restrictions on whether or not a request can be granted. Rules/constraints are classified into hard or soft. Hard constraints are those that can never be violated.

The AI Scheduling Algorithm implements the CSP model using a heuristic-driven AI search algorithm. The

CSP “select variable” heuristic was implemented using job priorities (highest priority first) and time (earlier requests first). This mimics the same heuristic used by human planners - schedule jobs chronologically and according to priority, then do “combines” afterwards.

The AI Scheduling Algorithm was designed to handle different types of engineering works, such as possession requests, pedestrian access, special work, resource reservation, stock transfer, patrolling, etc. The scheduling algorithm determines which engineering work should be performed and when, allocates all the necessary resources, and how to combine requests if there is not enough resources.

The AI Scheduling Algorithm is coded as a “2-pass” algorithm. The first pass sweeps through all the requests, satisfying as many as possible and then combining those with prefer combine. The second pass tries to “squeeze” in more requests using resources saved from the first pass optimization. At the end it tries to add even more requests through additional combines. The following is a simplified pseudo-code for the AI Scheduling Algorithm:

- First pass through all requests:
 1. Execute CSP algorithm on each engineering work request using search heuristics.
 2. Optimize by combining allocated requests to save resources, based on a set of optimization rules.
- Second pass through remaining requests which cannot be scheduled in the first pass:
 1. Execute CSP algorithm on remaining requests
 2. Perform optimization on newly allocated requests

The final step is to perform train outstabling. This is a separate algorithm that allocates outstabling locations to passenger trains. From time to time railway trains may need to be outstabled to temporary locations, such as stations, sidings, depots, etc., until they are needed for regular operations. During the night, trains are outstabled to various locations along the rail network so that when operations start again next day, the trains will be nearby their originating station or conveniently located so that they can be put into service whenever needed. The train outstabling problem is also modeled as a CSP. Details of this algorithm are documented in (Chun 2012).

Although the design of the scheduling algorithm is similar to a classical single-thread algorithm, the REST APIs are stateless and hence allows all AI services, including scheduling, to run on different servers to improve performance. The AaaS load balancing and cloud implementation allows AI to scale up easily when needed.

The Interactive Rescheduling Phase

The interactive rescheduling phase is used to modify a schedule after it has been officially approved. This is needed to accommodate any last minute changes or urgent jobs. The algorithm used is a traditional AI “iterative repair

algorithm” (Zweben 1994, Rabideau 1999). The main objective is to determine if a change can be accommodated without any impact to the existing schedule. If not, it then tries to find a way to satisfy the request through various different types of “combine” operations. This algorithm is an “iterative repair” algorithm as it iteratively finds problems and repairs the schedule. It uses the same set of conflict checking rules and search heuristics as that for the Scheduling Phase.

Application Use and Payoff

The original version of the AI Engine was deployed in July 2004 and has been responsible for scheduling engineering works for all the rapid transit subway lines and the Airport Express. The newly redesigned and enhanced AI Engine was deployed in July 2013 to also include the 3 commuter rails. The new ETMS replaces the existing ETMS as well as two other legacy systems used by commuter rails. The use of AI has been contributing to MTR’s 99.9% on-time performance and benefiting over 5 million passengers daily. The ETMS application is currently used by several thousand people, including both internal MTR staff as well as external contractors. The AI Engine is used in all phases of engineering works planning, scheduling, and interactive rescheduling. Application payoff includes:

- **Improved productivity** – with schedule automatically generated by AI, human planners can now focus on resolving difficult operational issues and resource contentions that necessitates human negotiation; saving several person-days of effort each week
- **Ensured operational safety** – since the AI Engine checks all rules all the time, it eliminates any potentials of human oversight
- **Maximized resource utilization** – different engineering work combinations are explored to maximize resource and track possession utilization; allowing MTR to do more with the same set of resources
- **Streamlined workflows** – the human aspect of scheduling is streamlined for efficiency
- **Streamlined decision making & problem solving** – changes/modifications are automatically resolved
- **Reduced nightly dispatch time** – With ETMS installed in the operations control center, the nightly dispatching of engineering works is streamlined, saving over 30 minutes of valuable time each night that can now be used for actual work.
- **Improved quality of service** – by providing better schedules for engineering works, quality of service for passengers during daytime operations will improve; helping MTR achieve 99.9% on-time performance.

- **Well-positioned for business growth** – the new “AI as a Service” (AaaS) model allows MTR to easily scale its rail management business and share its engineering works scheduling best practices with rail operations that MTR manages in other cities around the world.

Considering just tangible benefits alone, the use of ETMS will provide over US\$1 million productivity gain annually.

Application Development and Deployment

The Tender was issued mid-2010 and the Prime Contract was awarded to PCCW Solutions Ltd. end of 2010. Subsequently, PCCW sub-contracted the AI work to the CityU Professional Services Ltd., a wholly-owned subsidiary of the City University of Hong Kong. PCCW designed and developed the ETMS application, while CityU designed and developed the AI Engine. The AI Engine operated independently of the ETMS application. ETMS accesses the AI Engine through simple REST service calls. The development spanned over two years and was divided into several phases:

Phase 1

The first phase focused on re-engineering the existing AI Engine to fit the new AI requirements of the commuter rails and the light rail, namely the West Rail Line (WRL), East Rail Line (EAL), Light Rail Line (LRL), and Ma On Shan Line (MOL); collectively called the WELM lines. This work consists of several key tasks:

- **Redesigned application architecture** – the AI Engine was redesigned to be a highly scalable “AI as a Service” (AaaS) cloud architecture with clustering and load-balancing; giving it ability to grow as needed.
- **Redesigned AI architecture** – all rail-specific data and knowledge are stored outside of the AI Engine in the ETMS database; making the AI Engine totally data-driven and allowing it to be easily deployed to other cities and rail lines, by providing appropriate data through REST service calls.
- **Enhanced AI Model** – the model was generalized and enhanced to fit the needs, practices and requirements of not only subway lines but also traditional rail lines.
- **Enhanced AI Knowledge** – AI rules/constraints and various scheduling/optimization algorithms were enhanced to cover additional types of knowledge and processing required of commuter rail lines.

Phase 2

While Phase 1 focused on the WELM lines, Phase 2 involves retrofitting the data for the existing subway lines to fit the format of the new AI Engine design. This included the Urban Railway Lines (URL), Airport Express

Line (AEL), Tung Chung Line (TCL), and the Disneyland Resort Line (DRL); collectively called the DUAT lines.

User Acceptance Testing (UAT) and Deployment

Because of the mission critical nature of the system and the large volume of AI knowledge encoded, a comprehensive and extensive testing period was needed to collect enough real operational data and to ensure all the necessary rules and constraints were coded properly. The ETMS application without the AI was deployed in Nov 2012 for operational use. At the same time, the AI Engine underwent extensive UAT of the knowledge-base using actual operational data that was accumulated after the launch. The extended UAT helped uncover “gaps” in the knowledge-base that needed to be filled or parameters that needed to be fine-tuned. The AI Engine was soft launched for parallel run in May 2013, and was fully deployed in July 2013 for all lines, i.e. both WELM and DUAT lines.

ETMS is now accessed by over 3,000 users and is also installed in MTR’s “Super OCC” (Operations Control Center) to manage over 2,600 engineering works each week, performed by over 10,000 maintenance personnel. Since ETMS release in Nov 2012, the system had a perfect 100% availability record.



Figure 5. MTR “Super OCC” (Operations Control Center)

Future Plans

The network in Hong Kong continues to be expanded, with several extensions under construction. With the AaaS design of the AI Engine, new lines/extensions and consequently new rule/constraint instances can be easily accommodated without any change to the AI Engine itself. The AaaS design also provides MTR with the ability to quickly deploy the system to other cities. For example, besides Hong Kong, MTR also operates lines in Beijing, Hangzhou, and Shenzhen in China, as well as the London Overground and the Melbourne and Stockholm systems. MTR is exploring the possibility of deploying ETMS at those cities as well. Since MTR aims to become one of the biggest operators of metro systems in the world, the ability

to quickly deploy this mission critical system at other locations gives them greater business leverage.

Maintenance

As any railway grows, there will be changes in the rail network, equipments, operational parameters, rules, constraints, etc. Since all rail-specific data reside outside the AI Engine, they can easily be changed anytime by MTR and passed to the AI via REST calls. AI algorithms, such as conflict checking, scheduling, and optimization, rarely need to be modified. Since AI deployment in July 2013, there has not been a single AI maintenance task.

Conclusion

This paper described how AI is used to plan, schedule, and optimize engineering works for railway lines in Hong Kong. Besides streamlining the scheduling/rescheduling processes for engineering works, AI optimizes resource allocation to maximize the number of works that can be performed. Rules are used to ensure all safety, operational, and statutory constraints are met. AI is also used to schedule nightly outstabling locations for passenger trains such that they do not interfere with engineering works. The software architecture design is also special. A cloud-based “AI as a Service” (AaaS) model was used; AI services are provided via client-agnostic and rail-line agnostic REST APIs. This AaaS model allows AI to be easily deployed to potentially serve other rail lines outside of Hong Kong through straightforward REST calls. We believe MTR is currently the only rail operator in the world to use AI for engineering works management. In addition, ETMS is most likely the world’s first engineering works management system to support all the main rail types – rapid transit, commuter, and light rail.

References

Apt, K.R., Monfroy, E. 2001, “Constraint programming viewed as rule-based programming,” *Theory and Practice of Logic Programming*, 1(6):713-750.

Burbeck, S. 1992. “Application Programming in Smalltalk-80: How to use Model-View-Controller (MVC).” University of Illinois in Urbana-Champaign (UIUC) Smalltalk Archive. Available at st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html

Chun, H.W. and Yeung, Wai Ming, 2004. “Rule-based Approach to the Validation of Subway Engineering Work Allocation Plans,” In *the Proceeding of the International Conference on Computing, Communications and Control Technologies*, August 14-17, 2004, Austin, Texas, USA.

Chun, H.W., Yeung, W.M., Lam, P.S., Lai, D., Keefe, R., Lam, J., and Chan, H., 2005. “Scheduling Engineering Works for the MTR Corporation in Hong Kong,” In *Proceedings of the 17th*

Conference on Innovative Applications of Artificial Intelligence, Pittsburgh, July, 2005.

Chun, H.W., 2013. “Train Outstable Scheduling as Constraint Satisfaction,” In *Proceedings of the 25th Conference on Innovative Applications of Artificial Intelligence*, Bellevue, July, 2013.

CityU homepage (n.d.), <http://www.cityu.edu.hk/> (accessed Jan 8, 2014).

Cohen, J. 1990. Constraint Logic Programming. *Communications of the ACM*. 33(7):52-68.

Costello, R.L. 2010. “Building Web Services the REST Way” Available at: <http://www.xfront.com/REST-Web-Services.html>

Kumar, V. 1992. “Algorithms for Constraint Satisfaction Problems: A Survey,” In *AI Magazine*, 13(1):32-44.

MTR homepage (n.d.), http://www.mtr.com.hk/eng/homepage/cust_index.html (accessed Jan 8, 2014).

Ng, J. “Hong Kong’s Subway System Wants to Run the World,” In *The Wall Street Journal*, <http://online.wsj.com/news/articles/SB10001424127887323342404579080682338101534>, 18 Sep 2013.

PCCW Solutions Limited homepage (n.d.), <http://www.pccwsolutions.com/public/Home> (accessed Jan 8, 2014)

Padukone, N. 2013. “The Unique Genius of Hong Kong’s Public Transportation System,” In *The Atlantic*, <http://www.theatlantic.com/china/archive/2013/09/the-unique-genius-of-hong-kongs-public-transportation-system/279528/>, 10 Sep 2013.

Rabideau, G., Knight, R., Chien, S., Fukunaga, A., & Govindjee, A. 1999. “Iterative Repair Planning for Spacecraft Operations Using the ASPEN System,” In *Proceedings of the Fifth International Symposium, ISAIRAS ’99*, 1-3 June, 1999.

Rodriguez, A. 2010. “RESTful Web services: The basics.” Available at: <http://www.ibm.com/developerworks/webservices/library/ws-restful/index.html>

Steele, G.L. Jr. 1980. The Definition and Implementation of a Computer Programming Language Based on Constraints, Ph.D. Thesis, MIT.

Tsang, E. 1993, *Foundations of Constraint Satisfaction*, Academic Press.

Van Hentenryck, P. 1989. *Constraint Satisfaction in Logic Programming*, MIT Press.

Wikipedia contributors (n.d.), “Kowloon–Canton Railway,” Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Kowloon%E2%80%93Canton_Railway (accessed Jan 8, 2014).

Wikipedia contributors (n.d.), “MTR,” Wikipedia, The Free Encyclopedia, <http://en.wikipedia.org/wiki/MTR> (accessed Jan 8, 2013).

Zweben M., Daun B., Davis E., and Deale M. 1994. “Scheduling and Rescheduling with Iterative Repair,” in *Intelligent Scheduling*, Morgan Kaufman, San Francisco.