Title of Invention

Integration Platform to enable Operational Intelligence and User Journeys for Smart Cities and the Internet of Things

Background of the Invention

Analogy – evolution of the Internet

Originally, the Internet operated in "silos". The Internet was used primarily to exchange data between backend systems – such as email servers and news servers. In short, 'people' were not users of the Internet.

Beginning in the early 1990s, the advent of search engines provided a reason for people to become Internet users. From that time, the Internet has grown exponentially.

When they first became available "online," search engines indexed 'metadata' of websites and enabled users to search across websites (similar to the way a catalogue was once used to find a book in the library).

As Internet usage increased, the metadata indices became too big to browse through. Additionally, casual users (who became primary users of the Internet) demanded 'smarter search functions' to simplify search.

As a response, search engines started leveraging on 'relationships' between raw metadata to filter search results based on popularity and geo location – to produce more relevant search results.

Today, many Internet-based services operate by leveraging those 'metadata' and 'metadata relationships' – beyond just search.

Infrastructure management – following same path

The same trend is taking place in infrastructure management.

As with the Internet, the field of infrastructure management started with "silo based" applications and operations. Data captured (through devices and human data entry) stayed in applications – silos. Where integration across systems was required, custom solutions were made by solution integrators. These custom solutions were designed to serve 'trained operators' who often played dedicated infrastructure management roles – called by names such as "facility manager" or "building engineer".

IoT and IoE

In another dimension, city system infrastructures began to grow in sophistication and variety. During last decade, many internet-based devices made their way to marketplace – ushering in the era of the Internet of Things (IoT). System integration is no longer merely between 'systems' -- it must include IoT as well.

In short, infrastructure is evolving evolved from 'systems and devices' to "Internet of Everything" (IoE) –networks of computers, devices, people and cloud services. Any person with a smartphone is now a node of the IoE.

Summary of the Invention

Smart City – smart management of infrastructure

Widely growing adoption of digital technologies to manage cities, a.k.a Smart Cities, with the goal of enhancing performance of city operations, reducing cost and improving service to citizens (end-users of systems), by having a closer engagement with them, demands a sophisticated approach to infrastructure management. These infrastructures include both physical systems such as energy and water supply, transportation, etc., and software systems such as supervisory and data acquisition (SCADA), maintenance management, lease management, etc.

Managing infrastructure systems in isolation to make them 'run' is no longer adequate. Infrastructures should be 'aligned' with one-another to serve the needs of the city and its citizens. This demands that data and system silos be broken down, and that their operations be integrated.

The complexity of smart city implementation comes from two fundamental challenges: first, a smart city is a 'living system', which must constantly evolve to serve city needs and to reflect constantly changing and moving citizenry/population; and secondly, people and skills involved in modeling city services are different from people and skills involved in system implementation and integration.

Engagement of end users – hence 'user journey'

Further, mobile users are growing their expectation of user experience when engaging with smart city solutions. Solutions must be highly intuitive, as nobody is expected to 'learn' to use new software. Mobile users have a very short operation/attention timespan. They expect system to provide ONLY what they need to carry out their operation.

Smart City solutions need to address this by modeling 'user journeys' – providing a template containing the anticipated exact steps that a particular type of user carries to interact with the system to achieve an objective.

Neat solution – metadata model of IoE for the city

The present invention solves the problem by separating the problem into two tasks: implementing a "metadata model of IoE" of the city; and using it to implement citizen engagement through 'user journeys' and system intelligence through automated response logic.

More specifically, the present invention enables system integrators to build a metadata model of IoE, which enables city managers to implement user journeys and system intelligence, in incremental fashion, as business needs of the city evolves.

Summary of the solution:

a) Extract metadata from IoE (subsystems, applications and devices) including both data and methods;
b) Setup relationships between individual metadata elements to form a metadata model for the city;
c) Processes events, apply analytics to filter and qualify events (operational intelligence);
d) Set up automated orchestrations of actions to 'self manage' known situations;
e) Set up 'user journeys'; and
f) Connect with end-users (citizens) through unified communication.

Example

You (as a FM manager of a building) could receive a SMS [unified communication] indicating that your building is warm and occupants feel uncomfortable.

In a normal situation, a control system reporting 26 degrees [metadata] is not considered a problem. However, according to the present invention, the system is monitoring metadata related to user-devices (mobile phones) of building occupants [metadata model] who are sending out Tweets indicating 'office is hot'. By correlating the control system metadata concerning room temperature with metadata from other systems, i.e., mobile phones on the IoE [event processing & analytics], the system can qualify the situation as "important to respond".

Having received the SMS concerning the situation, you(the FM manager) open your mobile app, and it will show a single page with the details of the problem, and also what system thinks the root cause is - "energy saving" mode has automatically kicked in due to a sudden rise of dynamic energy price [automated orchestration].

The same mobile app page will give you the option to turn off the energy saving mode, and will show you how much extra it will cost the company, without you having to navigate to various systems to gather such knowledge [modeling of user journey].

Detail Description of Invention

Introduction

This invention applies to the field of smart infrastructure management – also known as Smart City solutions, and comprises the features and elements described below.

Dynamic metadata map

In the case of a smart city, the system of the invention would include a metadata model that is made up of metadata nodes that represent each internet-aware device/entity in the city. Since this comprehensive metadata model includes people with smartphones, who are constantly moving around and going in and

out of the city (solution boundary), the metadata model is adapted to be constantly changing.

Therefore, metadata model of IoE cannot be manually built. It must be automatically built, dynamically as metadata changes.

Metadata nodes

The metadata model is made up of metadata nodes. Each node is a collection of metadata. For example, 'Meeting Room One' is a metadata node, and it carries metadata such as 'Location', 'Seating Capacity', 'Amenities', etc.

Events and event engine

Constructing and modifying of metadata nodes is done by an 'event engine'. It receives metadata in the form of 'events'. An exemplary event might be 'Visitor V001 arrived at Gate G001'. Upon receiving new or modified metadata, the event engine either creates new metadata nodes or modifies existing nodes to absorb newly received information into metadata model.

Metadata node template

How a metadata node should represent an entity is defined in a 'metadata node template', which is defined by the system integrator. For example, the 'Visitor in Building' node template would identify 'Arrived Gate', 'Meeting Location' and 'Host' as metadata placeholders. Upon receiving 'visitor arrived' event, new 'Visitor in Building' node would be created by event engine to represent Visitor V001.

Forming metadata model

The 'metadata model' is formed by connecting metadata nodes through common metadata that exists in different nodes. This is done by the event engine as part of processing of received events. In the above example, upon receiving 'Visitor V001 arrived at gate G001' event, two embedded metadata (visitor's ID and gate's ID) are extracted and stored in metadata placeholders in the node. Further, the metadata node template (Visitor in Building) also requests the event engine to identify 'meeting location' and 'host', to complete its structure. The event engine explores existing metadata models to find matching nodes and to extract missing metadata to complete the node as specified by node template.

Please refer to figure 1 in Drawings section.

Extraction of metadata from subsystems

This requires identifying data elements and methods of the silo-system. A "Connector," a software executable, identifies metadata from the subsystem and sends them to 'event engine' as 'events'. Alternatively, subsystems could be programmed to 'push' their metadata to the event engine directly.

Examples of subsystems according to the invention include: air conditioning and heating systems, elevator systems, card access systems, CCTV systems and BMS (building management systems).

Please refer to Figure 2 in Drawings section.

Correlation of events

In the above example, Connector may later find that Visitor V001 is leaving the building. At that point, Connector will raise 'Visitor V001 left' event. The event engine will correlate this event to existing 'Visitor In Building' node for V001 and make the necessary updates to reflect the current status.

Navigation through metadata model

As shown in figure 3 in Drawings section, the metadata model (collection of nodes) behaves similar to an 'object collection' in object oriented programming. It forms a hierarchy (such as parent-child relationship in Locations), and it carries object attributes (such as node's attributes representing metadata) and methods (actions represented by metadata).

For example, Location metadata ('Room 001') of equipment ('A001') gets attached to Location ('Room 001') attribute (metadata). Location hierarchy makes it immediately inherit parent location ('Floor 001') of Room 001.

Any service that is looking for 'photocopy machines' (equipment of category 'photocopy') in Location 'Floor 001' will now find A001.

The metadata model is a machine readable structure that does not require a human to browse through it.

Cross references between metadata nodes

This invention allows system implementation to take place progressively. When the system is expanded by adding new metadata node templates and connectors, it is possible to build cross references between new nodes and existing nodes, without having to modify existing nodes.

As shown in figure 4 in Drawings section, originally implemented 'people count' map node (representing floor F001) maintains list of employees in F001, collected from Card Access system, which is required in case of emergency evacuation.

However, people count node is not aware of newly introduced event 'Visitor V001 arrived', collected from QR Code scanner. Therefore, when a new visitor arrives, it doesn't reflect on people count.

This problem is solved by having newly introduced Visitor node raise a 'secondary' event, 'Person V001 entered F001', which is recognized by original people count node (without having to modify it).

Consolidated actions – orchestrations

As explained in the example above, 'people count' node has a list of people (both employees and visitors) who are currently in floor F001. The present invention allows system integrator to define a single action, e.g., "notify emergency evacuation" to everyone in F001, regardless of the silo that captured their presence, whether it was Room Booking systems managing visitors, Work Order system managing contractors, or Space Management system managing seating locations for internal staff.

Automated Orchestrations

In some cases, it is required to automatically execute a series of actions, upon receiving an event. For example, People Count node may also be programmed to send SMS notification to all 'persons' in the particular floor when evacuation is ordered.

More importantly, additional metadata that would identify a person as 'handicap' could enable people count node to notify FM supervisors about locations of handicap persons in the building for special handling during the evacuation.

Complex event processing (analytics)

This invention includes an analytic engine that is able to process events and event-data to identify patterns and correlations. Examples are as follows:

a) Generate extended event based on repeated basic events within a given timespan (repeated "card rejected" event may generate "card reader faulty" event)
b) Correlate repeated events into a single event (repeated smoke-detected events indicates the single fire-incident event)
c) Transform data by consolidation of events, and include them as event-data ("power consumption" data in a series of events can be consolidated into "energy usage" data)
d) Predict secondary events by analyzing heuristics of primary events (event history may indicate that sustained 'car park full' event is generally followed by 'dirty toilet' feedback event for a particular restroom located next to car park. This is not a configured correlation but identified automatically through heuristics event data)

Enabling Operational Intelligence

Processing events through analytics allows identifying meaningful events – both threats and opportunities. It enables "operational intelligence".

For example, when one smoke detector reports a fire in a particular location, a new metadata node may be initiated to build necessary connections between the city's metadata elements and to orchestrate certain actions such as notification to first responders.

A few minutes later, more smoke detectors (in that Location) will likely report a fire. The system's response to 2nd and 3rd smoke detected events should be different from the first event. They act as "confirmation" to first event, rather than initiating new fire-incident scenarios.

Similarly, additional metadata can be appended to an event to add 'more sense'. In the same example above, if the first smoke detected alarm came from a detector that has produced a 'false alarm' a week ago, it would be wise to look for a 2nd confirmation from another detector, or a human confirmation, before the orchestration is carried out to respond to a fire.

Enabling Predictive Operations

Operational intelligence produced through this invention is 'machine readable'. As a result, it allows predetermined orchestrations to be executed based on

complex event conditions. This allows operators to manage the infrastructure predictively.

For example, in a large space such as the lobby of a shopping mall, it takes a long time for the air-conditioning system to react when more shoppers come into the mall. This is due to the natural delay in warmed air reaching air-conditioning ducts on the roof, where sensors are placed to recognize the rising temperature.

This invention allows system integrators to setup an orchestration that acts on 'people count' that is monitored through an independent system to influence air-conditioning system to produce more air-conditioning, predictively.