DBTA 100 Companies that Matter Most in Data

Franz Inc. is proud to announce it has been named to Database Trends and Applications “100 Companies That Matter Most in Data.” .

AllegroGraph provides organizations with essential Knowledge Graph solutions, including Large Language Models (LLMs), Graph Neural Networks, Graph Virtualization, GraphQL, Apache Spark graph analytics, and Kafka streaming graph pipelines. These capabilities exemplify AllegroGraph’s leadership in empowering data analytics professionals to derive business value out of Knowledge Graphs.

“The data landscape continues to increase in size and complexity and is more distributed than ever before,” said Tom Hogan, Group Publisher, Database Trends and Applications. “Spanning the wide range of established legacy technologies from MultiValue to cutting-edge breakthroughs such as LLMs, the DBTA 100 is a list of hardware, software, and service providers working to enable their customers’ data-driven future.”

“Franz Inc. is continually innovating and we are honored to receive this acknowledgement for our efforts to deliver leading edge solutions in Data managment,” said Dr. Jans Aasman, CEO, Franz Inc. “Organizations across a range of industries are realizing the critical role that Knowledge Graphs play in creating rich, yet flexible Enterprise Data Fabrics and AI-driven applications. AllegroGraph with FedShard uniquely provides companies with the foundational environment for delivering Graph based AI solutions with the ability to continually enrich and contextualize the understanding of data.”

AllegroGraph v7.2 – Now Available (GNN, Virtual Graphs, Spark, and Kafka)

AllegroGraph 7.2, provides organizations with essential Data Fabric tools, including Graph Neural Networks, Graph Virtualization, Apache Spark graph analytics, and streaming graph pipelines. These new capabilities exemplify AllegroGraph’s leadership in empowering data analytics professionals to derive business value out of Knowledge Graphs.

Graph Neural Networks

With AllegroGraph 7.2, users can create Graph Neural Networks (GNNs) and take advantage of a mature AI approach for Knowledge Graph enrichment via text processing for news classification, question and answer, search result organization, event prediction, and more. GNNs created in AllegroGraph enhance neural network methods by processing the graph data through rounds of message passing, as such, the nodes know more about their own features as well as neighbor nodes. This creates an even more accurate representation of the entire graph network. AllegroGraph GNNs advance text classification and relationship extraction for enhancing enterprise-wide Data Fabrics.

Graph Virtualization

AllegroGraph 7.2 allows users to easily virtualize data as part of their AllegroGraph Knowledge Graph solution. When graphs are virtual, the data remains in the source system and is easily linked and queried with other data stored directly in AllegroGraph.

Any data source with a supported JDBC driver can be integrated into an AllegroGraph Knowledge Graph, including Databases (i.e. Apache Cassandra, AWS Athena, Microsoft SQL Server, MongoDB, MySQL, Oracle Database); BI Tools (i.e. IBM Cognos, Microsoft PowerBI, RapidMiner, Tableau); CRM Systems (i.e. Dynamics CRM, Netsuite, Salesforce, SugarCRM); Cloud Services (i.e. Active Directory, AWS Management, Facebook, Marketo, Microsoft Teams, SAP, ServiceNow) and Shared Data Files (i.e. Box, Gmail, Google Drive, Office365).

Streaming Graph Pipelines using Kafka

Enterprises that need real-time experiences are starting to adopt streaming pipelines to provide insights that adapt to new data in real-time rather than processing data in batches. AllegroGraph is often used as an Entity Event Knowledge Graph platform in diverse settings such as call centers, hospitals, insurance companies, aviation organizations and financial firms.

AllegroGraph 7.2 can be used seamlessly with Apache Kafka, an open-source distributed event streaming platform for high-performance data pipelines, streaming analytics, data integration and mission-critical applications. By coupling AllegroGraph with Apache Kafka, users can create a real-time decision engine that produces real-time event streams based on computations that trigger specific actions. AllegroGraph accepts incoming events, executes instant queries and analytics on the new data and then stores events and results.

Graph Analytics with Apache Spark

AllegroGraph 7.2 enables users to export data out of the Knowledge Graph and then perform graph analytics with Apache Spark, one of the most popular platforms for large-scale data processing. Users immediately gain machine learning and SQL database solutions as well as GraphX and GraphFrames, two frameworks for running graph compute operations on data.

A key benefit of using Apache Spark for graph analytics within AllegroGraph is that it is built on top of Hadoop MapReduce and extends the MapReduce model to efficiently use more types of computations. Users can access interfaces (including interactive shells) for programming entire clusters with implicit data parallelism and fault-tolerance.

Availability of AllegroGraph 7.2

AllegroGraph 7.2 is immediately available directly from Franz Inc. For more information, visit the AllegroGraph Quick Start page for cloud and download options.

Examples

Visit our Github AllegroGraph Examples page.

AllegroGraph Named to 100 Companies That Matter Most in Data

Franz Inc. Acknowledged as a Leader for Knowledge Graph Solutions

Lafayette, Calif., June 23, 2020 – Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology for Knowledge Graph Solutions, today announced that it has been named to The 100 Companies That Matter in Data by Database Trends and

Applications. The annual list reflects the urgency felt among many organizations to provide a timely flow of targeted information. Among the more prominent initiatives is the use of AI and cognitive computing, as well as related capabilities such as machine learning, natural language processing, and text analytics. This list recognizes companies based on their presence, execution, vision and innovation in delivering products and services to the marketplace.

“We’re excited to announce our eighth annual list, as the industry continues to grow and evolve,” remarked Thomas Hogan, Group Publisher at Database Trends and Applications. “Now, more than ever, businesses are looking for ways transform how they operate and deliver value to customers with greater agility, efficiency and innovation. This list seeks to highlight those companies that have been successful in establishing themselves as unique resources for data professionals and stakeholders.”

“We are honored to receive this acknowledgement for our efforts in delivering Enterprise Knowledge Graph Solutions,” said Dr. Jans Aasman, CEO, Franz Inc. “In the past year, we have seen demand for Enterprise Knowledge Graphs take off across industries along with recognition from top technology analyst firms that Knowledge Graphs provide the critical foundation for artificial intelligence applications and predictive analytics.

Our recent launch of AllegroGraph 7 with FedShard, a breakthrough that allows infinite data integration to unify all data and siloed knowledge into an Entity-Event Knowledge Graph solution will catalyze Knowledge Graph deployments across the Enterprise.”

Gartner recently released a report “How to Build Knowledge Graphs That Enable AI-Driven Enterprise Applications” and have previously stated, “The application of graph processing and graph databases will grow at 100 percent annually through 2022 to continuously accelerate data preparation and enable more complex and adaptive data science." To that end, Gartner named graph analytics as a "Top 10 Data and Analytics Trend" to solve critical business priorities. (*Source: Gartner, Top 10 Data and Analytics Trends, November 5, 2019)*.

"Graph databases and knowledge graphs are now viewed as a must-have by enterprises serious about leveraging AI and predictive analytics within their organization," said Dr. Aasman "We are working with organizations across a broad range of industries to deploy large-scale, high-performance Entity-Event Knowledge Graphs that serve as the foundation for AI-driven applications for personalized medicine, predictive call centers, digital twins for IoT, predictive supply chain management and domain-specific Q&A applications – just to name a few."

Forrester Shortlists AllegroGraph

AllegroGraph was shortlisted in the February 3, 2020 Forrester Now Tech: Graph Data Platforms, Q1 2020 report, which recommends that organizations "Use graph data platforms to accelerate connected-data initiatives." Forrester states, "You can use graph data platforms to become significantly more productive, deliver accurate customer recommendations, and quickly make connections to related data."

Bloor Research covers AllegroGraph with FedShard

Bloor Research Analyst, Daniel Howard noted "With the 7.0 release of AllegroGraph, arguably the most compelling new capability is its ability to create what Franz refers to as "Entity-Event Knowledge Graphs" (or EEKGs) via its patented FedShard technology." Mr. Howard goes on to state "Franz clearly considers this a major release for AllegroGraph. Certainly, the introduction of an explicit entity-event graph is not something I've seen before. The newly introduced text to speech capabilities also seem highly promising."

AllegroGraph Named to KMWorld’s 100 Companies That Matter in Knowledge Management

AllegroGraph was also recently named to KMWorld’s 100 Companies That Matter in Knowledge Management. The KMWorld 100 showcases organizations that are advancing their products and capabilities to meet changing requirements in Knowledge Management.

Franz Knowledge Graph Technology and Services

Franz’s Knowledge Graph Solution includes both technology and services for building industrial strength Entity-Event Knowledge Graphs based on best-of-class tools, products, knowledge, skills and experience. At the core of the solution is Franz’s graph database technology, AllegroGraph with FedShard, which is utilized by dozens of the top F500 companies worldwide and enables businesses to extract sophisticated decision insights and predictive analytics from highly complex, distributed data that cannot be uncovered with conventional databases.

Franz delivers the expertise for designing ontology and taxonomy-based solutions by utilizing standards-based development processes and tools. Franz also offers data integration services from siloed data using W3C industry standard semantics, which can then be continually integrated with information that comes from other data sources. In addition, the Franz data science team provides expertise in custom algorithms to maximize data analytics and uncover hidden knowledge.

Ubiquitous AI Demands A New Type Of Database Sharding

Forbes published the following article by Dr. Jans Aasman, Franz Inc.'s CEO.

The notion of sharding has become increasingly crucial for selecting and optimizing database architectures. In many cases, sharding is a means of horizontally distributing data; if properly implemented, it results in near-infinite scalability. This option enables database availability for business continuity, allowing organizations to replicate databases among geographic locations. It's equally useful for load balancing, in which computational necessities (like processing) shift between machines to improve IT resource allocation.

However, these use cases fail to actualize sharding's full potential to maximize database performance in today's post-big data landscape. There's an even more powerful form of sharding, called "hybrid sharding," that drastically improves the speed of query results and duly expands the complexity of the questions that can be asked and answered. Hybrid sharding is the ability to combine data that can be partitioned into shards with data that represents knowledge that is usually un-shardable.

This hybrid sharding works particularly well with the knowledge graph phenomenon leveraged by the world's top data-driven companies. Hybrid sharding also creates the enterprise scalability to query scores of internal and external sources for nuanced, detailed results, with responsiveness commensurate to that of the contemporary AI age.

Read the full article at Forbes.

Franz Inc. to Present at The Global Graph Summit and Data Day Texas

Dr. Jans Aasman, CEO, Franz Inc., will be presenting, “Creating Explainable AI with Rules” at the Global Graph Summit, a part of Data Day Texas. The abstract for Dr. Aasman’s presentation:

> *“There’s a fascinating dichotomy in artificial intelligence between statistics and rules, machine learning and expert systems. Newcomers to artificial intelligence (AI) regard machine learning as innately superior to brittle rules-based systems, while the history of this field reveals both rules and probabilistic learning are integral components of AI. This fact is perhaps nowhere truer than in establishing* explainable AI, *which is central to the long-term business value of AI front-office use cases.”*

> *"The fundamental necessity for explainable AI spans regulatory compliance, fairness, transparency, ethics and lack of bias – although this is not a complete list. For example, the effectiveness of counteracting financial crimes and increasing revenues from advanced machine learning predictions in financial services could be greatly enhanced by deploying more accurate deep learning models. But all of this would be arduous to explain to regulators. Translating those results into explainable rules is the basis for more widespread AI deployments producing a more meaningful impact on society."*

The Global Graph Summit is an independently organized vendor-neutral conference, bringing leaders from every corner of the graph and linked-data community for sessions, workshops, and its well-known before and after parties. Originally launched in January 2011 as one of the first NoSQL / Big Data conferences, Data Day Texas each year highlights the latest tools, techniques, and projects in the data space, bringing speakers and attendees from around the world to enjoy the hospitality that is uniquely Austin. Since its inception, Data Day Texas has continually been the largest independent data-centric event held within 1000 miles of Texas.

Three Necessities For Maximizing Your Digital Twins Approach

The digital twin premise is arguably the most viable means of implementing equipment asset management throughout the industrial internet. It's an exceptionally lucrative element of the internet of things (IoT), with an applicability that easily lends itself to numerous businesses. Its real-time streaming data, simulation capabilities and relationship awareness may well prove to be the “killer app” that makes the IoT mainstream.

Digital Twins Types

There are presently three types of digital twins: those for individual assets, operations and predictions. In this article, we will focus on individual assets. Examples of these assets include drilling machines in the oil and gas industry or assembly line equipment. Each type of digital twin creates a three-dimensional simulation of the real-world features it models based on relationships of IoT data. The simulated models capture and contextualize this low-latent data about each asset for vital visibility into its performance. This real-time data provides a blueprint for diminishing downtime, scheduling maintenance and monitoring other factors that impact overall asset productivity and ROI. At scale, each factor translates into significant savings, increased performance and greater chances for optimization.

The crux of the digital twin’s expansive capabilities is almost entirely predicated on solving one of the more time-honored data management difficulties: data modeling. But the schema issues complicating downstream data modeling processes such as transformation, integration and predictive analytics can be swiftly redressed by knowledge graphs that simplify this vital prerequisite. The standards-based data models of semantic knowledge graphs deliver unparalleled flexibility, interoperability and low latency for which IoT deployments are renowned. *(Full disclosure: My company specializes in semantic knowledge graphs.)*

Read the Full Article at Forbes.

The Importance of FAIR Data in Earth Science

Franz’s CEO, Jans Aasman’s recent Marine Technology News:

Dr. Jans Aasman, CEO, Franz Inc.

Data’s valuation as an enterprise asset is most acutely realized over time. When properly managed, the same dataset supports a plurality of use cases, becomes almost instantly available upon request, and is exchangeable between departments or organizations to systematically increase its yield with each deployment.

These boons of leveraging data as an enterprise asset are the foundation of GO FAIR’s Findable Accessible Interoperable Reusable (FAIR) principles profoundly impacting the data management rigors of geological science. Numerous organizations in this space have embraced these tenets to swiftly share information among a diversity of disciplines to safely guide the stewardship of the earth.

According to Dr. Annie Burgess, Lab Director of Earth Science Information Partners (ESIP), the “most pressing global challenges cannot be solved by a single organization. Scientists require data collected across multiple disciplines, which are often managed by many different agencies and institutions.” As numerous members of the earth science community are realizing, the most effectual means of managing those disparate data according to FAIR principles is by utilizing the semantic standards underpinning knowledge graphs.

Read the full article at Marine Technology News

SHACL – Shapes Constraint Language in AllegroGraph

SHACL is a SHApe Constraint Language. It specifies a vocabulary (using triples) to describe the shape that data should have. The *shape* specifies things like the following simple requirements:

- How many triples with a specified subject and predicate should be in the repository (e.g. at least 1, at most 1, exactly 1).
- What the nature of the object of a triple with a specified subject and predicate should be (e.g. a string, an integer, etc.)

See the specification for more examples.

SHACL allows you to validate that your data is conforming to desired requirements.

For a given validation, the shapes are in the *Shapes Graph* (where *graph* means a collection of triples) and the data to be validated is in the *Data Graph* (again, a collection of triples). The SHACL vocabularly describes how a given shape is linked to *targets* in the data and also provides a way for a Data Graph to specify the Shapes Graph that should be used for validatation. The result of a SHACL validation describes whether the Data Graph conforms to the Shapes Graph and, if it does not, describes each of the failures.

Namespaces Used in this Document

Along with standard predefined namespaces (such as rdf: for <http://www.w3.org/1999/02/22-rdf-syntax-ns#> and rdfs: for <http://www.w3.org/2000/01/rdf-schema#>), the following are used in code and examples below:

```
prefix fr: <https://franz.com#>
prefix sh: <http://www.w3.org/ns/shacl#>
prefix franz: <https://franz.com/ns/allegrograph/6.6.0/>
```

A Simple Example

Suppose we have a *Employee* class and for each Employee instance, there must be exactly one triple of the form

```
emp001 hasID "000-12-3456"
```

where the object is the employee's ID Number, which has the format is [3 digits]-[2 digits]-[4 digits].

This TriG file encapsulates the constraints above:

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://franz.com#Shapes> {
 <https://franz.com#EmployeeShape>
  a sh:NodeShape ;
  sh:targetClass <https://franz.com#Employee> ;
  sh:property [
    sh:path <https://franz.com#hasID> ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
           sh:pattern "^[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]$" ;
  ] .
}
```

It says that for instances of fr:Employee (sh:targetClass

<https://franz.com#Employee>), there must be exactly 1 triple with predicate (path) fr:hasID and the object of that triple must be a string with pattern [3 digits]-[2 digits]-[4 digits] (sh:pattern "^[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]$").

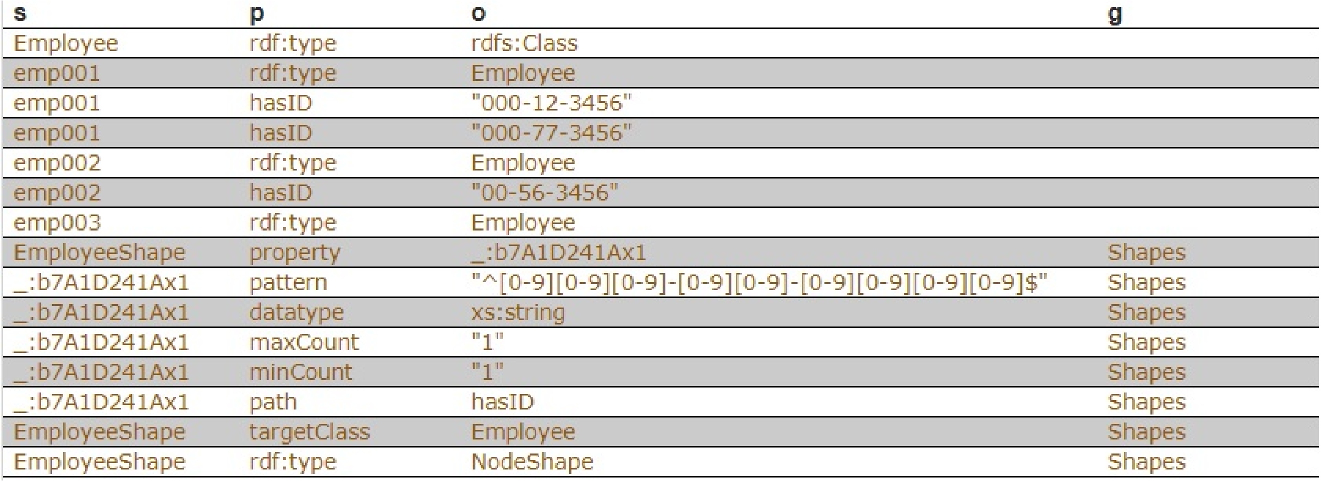
This TriG file defines the Employee class and some employee instances:

```
@prefix fr: <https://franz.com#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

{
 fr:Employee
  a rdfs:Class .
 fr:emp001
  a fr:Employee ;
  fr:hasID "000-12-3456" ;
  fr:hasID "000-77-3456" .
 fr:emp002
  a fr:Employee ;
  fr:hasID "00-56-3456" .
 fr:emp003
  a fr:Employee .
 }
```

Recalling the requirements above, we immediately see these problems with these triples:

1. *emp001* has two *hasID* triples.
2. The value of *emp002*'s ID has the wrong format (two leading digits rather than 3).
3. *emp003* does not have a *hasID* triple.

We load the two TriG files into our repository, and end up with the following triple set. Note that all the employee triples use the default graph and the SHACL-related triples use the graph <https://franz.com#Shapes> specified in the TriG file.



s	p	o	g
Employee	rdf:type	rdfs:Class	
emp001	rdf:type	Employee	
emp001	hasID	"000-12-3456"	
emp001	hasID	"000-77-3456"	
emp002	rdf:type	Employee	
emp002	hasID	"00-56-3456"	
emp003	rdf:type	Employee	
EmployeeShape	property	_:b7A1D241Ax1	Shapes
_:b7A1D241Ax1	pattern	"^[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]$"	Shapes
_:b7A1D241Ax1	datatype	xs:string	Shapes
_:b7A1D241Ax1	maxCount	"1"	Shapes
_:b7A1D241Ax1	minCount	"1"	Shapes
_:b7A1D241Ax1	path	hasID	Shapes
EmployeeShape	targetClass	Employee	Shapes
EmployeeShape	rdf:type	NodeShape	Shapes

Now we use **agtool shacl-validate** to validate our data:

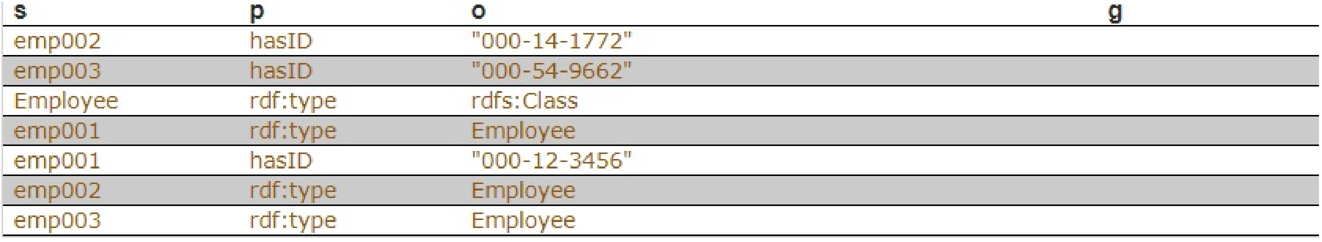
```
bin/agtool shacl-validate --data-graph default  --shapes-graph
https://franz.com#Shapes shacl-repo-1
Validation report:              Does not conform
Created:                        2019-06-27T10:24:10
Number of shapes graphs:        1
Number of data graphs:          1
Number of NodeShapes:           1
Number of focus nodes checked: 3

3 validation results:
Result:
 Focus node:                <https://franz.com#emp001>
 Path:                      <https://franz.com#hasID>
 Source Shape:              _:b7A1D241Ax1
                   Constraint                 Component:
<http://www.w3.org/ns/shacl#MaxCountConstraintComponent>
 Severity:                  <http://www.w3.org/ns/shacl#Violation>

Result:
 Focus node:                <https://franz.com#emp002>
 Path:                      <https://franz.com#hasID>
 Value:                     "00-56-3456"
 Source Shape:              _:b7A1D241Ax1
                   Constraint                 Component:
<http://www.w3.org/ns/shacl#PatternConstraintComponent>
 Severity:                  <http://www.w3.org/ns/shacl#Violation>

Result:
 Focus node:           <https://franz.com#emp003>
 Path:                 <https://franz.com#hasID>
 Source Shape:         _:b7A1D241Ax1
                  Constraint               Component:
<http://www.w3.org/ns/shacl#MinCountConstraintComponent>
 Severity:             <http://www.w3.org/ns/shacl#Violation>
```

The validation fails with the problems listed above. The **Focus node** is the subject of a triple that did not conform. **Path** is the predicate or a property path (predicates in this example). **Value** is the offending value. **Source Shape** is the shape that established the constraint (you must look at the shape triples to see exactly what **Source Shape** is requiring).

We revise our employee data with the following SPARQL expresssion, deleting one of the emp001 triples, deleting the emp002 triple and adding a new one with the correct format, and adding an emp003 triple.

```
prefix fr: <https://franz.com#>

DELETE DATA {fr:emp002 fr:hasID "00-56-3456" } ;

INSERT DATA {fr:emp002 fr:hasID "000-14-1772" } ;

DELETE DATA {fr:emp001 fr:hasID "000-77-3456" } ;

INSERT DATA {fr:emp003 fr:hasID "000-54-9662" } ;
```

Now our employee triples are



s	p	o	g
emp002	hasID	"000-14-1772"	
emp003	hasID	"000-54-9662"	
Employee	rdf:type	rdfs:Class	
emp001	rdf:type	Employee	
emp001	hasID	"000-12-3456"	
emp002	rdf:type	Employee	
emp003	rdf:type	Employee	

We run the validation again and are told our data conforms:

```
% bin/agtool shacl-validate --data-graph default  --shapes-graph https://franz.com#Shapes shacl-repo-1
```

```
Validation report:            Conforms
Created:                      2019-06-27T10:32:19
Number of shapes graphs:      1
Number of data graphs:        1
Number of NodeShapes:         1
Number of focus nodes checked: 3
```

When we refer to this example in the remainder of this document, it is to the un-updated (incorrect) triples.

SHACL API

The example above illustrates the SHACL steps:

1. Have a data set with triples that should conform to a shape
2. Have SHACL triples that express the desired shape
3. Run SHACL validation to determine if the data conforms

Note that SHACL validation does not modify the data being validated. Once you have the conformance report, you must modify the data to fix the conformance problems and then rerun the validation test.

The main entry point to the API is **agtool shacl-validate**. It takes various options and has several output choices. Online help for **agtool shacl-validate** is displayed by running agtool shacl-validate --help.

In order to validate triples, the system must know:

1. What tripes to examine
2. What rules (SHACL triples) to use
3. What to do with the results

Specifying what triples to examine

Two arguments to **agtool shacl-validate** specify the triples to evaluate: --data-graph and --focus-node. Each can be specified multiple times.

- The --data-graph argument specifies the graph value for triples to be examined. Its value must be an IRI or default. Only triples in the specified graphs will be examined. default specifies the default graph. It is also the default value of the --data-graph argument. If no value is specified for --data-graph, only triples in the default graph will be examined. If a value for --data-graph is specified, triples in the default graph will only be examined if --data-graph default is also specified.
- The --focus-node argument specifies IRIs which are subjects of triples. If this argument is specified, only triples with these subjects will be examined. To be examined, triples must also have graph values specified by --data-graph arguments. --focus-node does not have a default value. If unspecified, all triples in the specified data graphs will be examined. This argument can be specified multiple times.

The --data-graph argument was used in the simple example above. Here is how the --focus-node argument can be used to restrict validation to triples with subjects <https://franz.com#emp002>and <https://franz.com#emp003> and to ignore triples with subject <https://franz.com#emp001> (applying **agtool shacl-validate** to the orignal non-conformant data):

```
% bin/agtool shacl-validate --data-graph default  \
  --shapes-graph https://franz.com#Shapes \
  --focus-node https://franz.com#emp003 \
  --focus-node https://franz.com#emp002 shacl-repo-1
Validation report:            Does not conform
Created:                      2019-06-27T11:37:49
Number of shapes graphs:      1
Number of data graphs:        1
Number of NodeShapes:         1
Number of focus nodes checked: 2
```

```
2 validation results:
Result:
 Focus node:          <https://franz.com#emp003>
 Path:                <https://franz.com#hasID>
 Source Shape:        _:b7A1D241Ax2
 Constraint Component:
<http://www.w3.org/ns/shacl#MinCountConstraintComponent>
 Severity:            <http://www.w3.org/ns/shacl#Violation>

Result:
 Focus node:          <https://franz.com#emp002>
 Path:                <https://franz.com#hasID>
 Value:               "00-56-3456"
 Source Shape:        _:b7A1D241Ax2
 Constraint Component:
<http://www.w3.org/ns/shacl#PatternConstraintComponent>
 Severity:            <http://www.w3.org/ns/shacl#Violation>
```

Specifying What Shape Triples to Use

Two arguments to **agtool shacl-validate**, analogous to the two arguments for data described above, specify Shape triples to use. Further, following the SHACL spec, data triples with predicate <http://www.w3.org/ns/shacl#shapeGraph> also specify graphs containing Shape triples to be used.

The arguments to **agtool shacl-validate** are the following. Each may be specified multiple times.

- The --shapes-graph argument specifies the graph value for shape triples to be used for SHACL validation. Its value must be an IRI or default. default specifies the default graph. The --shapes-graph argument has no default value. If unspecified, graphs specified by data triples with the <http://www.w3.org/ns/shacl#shapeGraph> predicate will be used (they are used whether or not --shapes-graph has a value). If --shapes-graph has no value and there are no data triples with

the <http://www.w3.org/ns/shacl#shapeGraph> predicate, the data graphs are used for shape graphs. (Shape triples have a known format and so can be identified among the data triples.)
- The --shape argument specifies IRIs which are subjects of shape nodes. If this argument is specified, only shape triples with these subjects and subsiduary triples to these will be used for validation. To be included, the triples must also have graph values specified by the --shapes-graph arguments or specified by a data triple with the <http://www.w3.org/ns/shacl#shapeGraph> predicate. --shape does not have a default value. If unspecified, all shapes in the shapes graphs will be used.

Other APIs

There is a lisp API using the function validate-data-graph, defined next:

validate-data-graphdb &key data-graph-iri/s shapes-graph-iri/s shape/s focus-node/s verbose conformance-only?
function

Perform SHACL validation and return a validation-report structure.

The validation uses data-graph-iri/s to construct the dataGraph. This can be a single IRI, a list of IRIs or NIL, in which case the default graph will be used. The shapesGraph can be specified using the shapes-graph-iri/s parameter which can also be a single IRI or a list of IRIs. If shape-graph-iri/s is not specified, the SHACL processor will first look to create the shapesGraph by finding triples with the predicate sh:shapeGraph in the dataGraph. If there are no such triples, then the shapesGraph will be assumed to be the same as the dataGraph.

Validation can be restricted to particular shapes and focus nodes using the shape/s and focus-node/s parameters. Each of these can be an IRI or list of IRIs.

If conformance-only? is true, then validation will stop as soon as any validation failures are detected.

You can use validation-report-conforms-p to see whether or not the dataGraph conforms to the shapesGraph (possibly restricted to just particular shape/s and focus-node/s).

The function validation-report-conforms-p returns t or nil as the validation struct returned by validate-data-graph does or does not conform.

validation-report-conforms-preport
function

Returns t or nil to indicate whether or not REPORT (a validation-report struct) indicates that validation conformed.
There is also a REST API. See HTTP reference.

Validation Output

The simple example above and the SHACL examples below show output from **agtool validate-shacl**. There are various output formats, specified by the --output option. Those examples use the plain format, which means printing results descriptively. Other choices include json, trig, trix, turtle, nquads, rdf-n3, rdf/xml, and ntriples. Here are the simple example (uncorrected) results using ntriples output:

```
% bin/agtool shacl-validate --output ntriples --data-graph
default --shapes-graph https://franz.com#Shapes shacl-repo-1

_:b271983AAx1
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/ns/shacl#ValidationReport> .
```

```
_:b271983AAx1 <http://www.w3.org/ns/shacl#conforms>
"false"^^<http://www.w3.org/2001/XMLSchema#boolean> .
_:b271983AAx1 <http://purl.org/dc/terms/created>
"2019-07-01T18:26:03"^^<http://www.w3.org/2001/XMLSchema#dateT
ime> .
_:b271983AAx1 <http://www.w3.org/ns/shacl#result>
_:b271983AAx2 .
_:b271983AAx2
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/ns/shacl#ValidationResult> .
_:b271983AAx2 <http://www.w3.org/ns/shacl#focusNode>
<https://franz.com#emp001> .
_:b271983AAx2 <http://www.w3.org/ns/shacl#resultPath>
<https://franz.com#hasID> .
_:b271983AAx2 <http://www.w3.org/ns/shacl#resultSeverity>
<http://www.w3.org/ns/shacl#Violation> .
_:b271983AAx2
<http://www.w3.org/ns/shacl#sourceConstraintComponent>
<http://www.w3.org/ns/shacl#MaxCountConstraintComponent> .
_:b271983AAx2 <http://www.w3.org/ns/shacl#sourceShape>
_:b271983AAx3 .
_:b271983AAx1 <http://www.w3.org/ns/shacl#result>
_:b271983AAx4 .
_:b271983AAx4
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/ns/shacl#ValidationResult> .
_:b271983AAx4 <http://www.w3.org/ns/shacl#focusNode>
<https://franz.com#emp002> .
_:b271983AAx4 <http://www.w3.org/ns/shacl#resultPath>
<https://franz.com#hasID> .
_:b271983AAx4 <http://www.w3.org/ns/shacl#resultSeverity>
<http://www.w3.org/ns/shacl#Violation> .
_:b271983AAx4
<http://www.w3.org/ns/shacl#sourceConstraintComponent>
<http://www.w3.org/ns/shacl#PatternConstraintComponent> .
_:b271983AAx4 <http://www.w3.org/ns/shacl#sourceShape>
_:b271983AAx3 .
_:b271983AAx4 <http://www.w3.org/ns/shacl#value> "00-56-3456"
.
_:b271983AAx1 <http://www.w3.org/ns/shacl#result>
_:b271983AAx5 .
```

```
_:b271983AAx5
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/ns/shacl#ValidationResult> .
_:b271983AAx5     <http://www.w3.org/ns/shacl#focusNode>
<https://franz.com#emp003> .
_:b271983AAx5     <http://www.w3.org/ns/shacl#resultPath>
<https://franz.com#hasID> .
_:b271983AAx5   <http://www.w3.org/ns/shacl#resultSeverity>
<http://www.w3.org/ns/shacl#Violation> .
_:b271983AAx5
<http://www.w3.org/ns/shacl#sourceConstraintComponent>
<http://www.w3.org/ns/shacl#MinCountConstraintComponent> .
_:b271983AAx5    <http://www.w3.org/ns/shacl#sourceShape>
_:b271983AAx3 .
```

You can have the triples added to the repository by specifying the --add-to-repo option true.

In the plain output information is provided about how many data graphs are examined, how many shape graphs were specified and node shapes are found, and how many focus nodes are checked. If zero focus nodes are checked, that is likely not what you want and something has gone wrong. Here we mis-spell the name of the shape graph (https://franz.com#shapes instead of https://franz.com#Shapes) and get 0 focus nodes checked:

```
% bin/agtool shacl-validate --data-graph default --shapes-graph https://franz.com#shapes shacl-repo-1
Validation report:             Conforms
Created:                       2019-06-28T10:34:22
Number of shapes graphs:       1
Number of data graphs:         1
Number of NodeShapes:          0
Number of focus nodes checked: 0
```

SPARQL integration

There are two sets of magic properties defined: one checks for basic conformance and the other produces validation reports as triples:

- ?valid franz:shaclConforms (?dataGraph [?shapesGraph])
- ?valid franz:shaclFocusNodeConforms1 (?dataGraph ?nodeOrNodeCollection)
- ?valid franz:shaclFocusNodeConforms2 (?dataGraph ?shapesGraph ?nodeOrNodeCollection)
- ?valid franz:shaclShapeConforms1 (?dataGraph ?shapeOrShapeCollection [?nodeOrNodeCollection])
- ?valid franz:shaclShapeConforms2 (?dataGraph ?shapesGraph ?shapeOrShapeCollection [?nodeOrNodeCollection])
- (?s ?p ?o) franz:shaclValidationReport (?dataGraph [?shapesGraph])
- (?s ?p ?o) franz:shaclFocusNodeValidationReport1 (?dataGraph ?nodeOrNodeCollection) .
- (?s ?p ?o) franz:shaclFocusNodeValidationReport2 (?dataGraph ?shapesGraph ?nodeOrNodeCollection) .
- (?s ?p ?o) franz:shaclShapeValidationReport1 (?dataGraph ?shapeOrShapeCollection [?nodeOrNodeCollection]) .
- (?s ?p ?o) franz:shaclShapeValidationReport2 (?dataGraph ?shapesGraph ?shapeOrShapeCollection [?nodeOrNodeCollection]) .

In all of the above ?dataGraph and ?shapesGraph can be IRIs, the literal ‘default’, or a variable that is bound to a SPARQL collection (list or set) that was previously created with a function like https://franz.com/ns/allegrograph/6.5.0/fn#makeSPARQLList or https://franz.com/ns/allegrograph/6.5.0/fn#lookupRdfList. If a collection is used, then the SHACL processor will create a temporary RDF merge of all of the graphs in it to produce the data graph or the shapes graph.

Similarly, ?shapeOrShapeCollection and ?nodeOrNodeCollection can be bound to an IRI or a SPARQL collection. If a collection is used, then it must be bound to a list of IRIs. The SHACL processor will restrict validation to the shape(s) and focus node(s) (i.e. nodes that should be validated) specified.

The shapesGraph argument is optional in both of the shaclConforms and shaclValidationReport magic properties. If the shapesGraph is not specified, then the shapesGraph will be created by following triples in the dataGraph that use the sh:shapesGraph predicate. If there are no such triples, then the shapesGraph will be the same as the dataGraph.

For example, the following SPARQL expression

```
construct { ?s ?p ?o } where {
  # form a collection of focusNodes
bind(<https://franz.com/ns/allegrograph/6.6.0/fn#makeSPARQLList>(
    <http://Journal1/1942/Article25>,
    <http://Journal1/1943>) as ?nodes)
                          (?s ?p ?o)
<https://franz.com/ns/allegrograph/6.6.0/shaclShapeValidationReport1>
    ('default' <ex://franz.com/documentShape1> ?nodes) .
}
```

would use the default graph as the Data Graph and the Shapes Graph and then validate two focus nodes against the shape <ex://franz.com/documentShape1>.

SHACL Example

We build on our simple example above. Start with a fresh repository so triples from the simple example do not interfere with this example.

We start with a TriG file with various shapes defined on some classes.

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix fr: <https://franz.com#> .
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<https://franz.com#ShapesGraph> {
fr:EmployeeShape
   a sh:NodeShape ;
   sh:targetClass fr:Employee ;
   sh:property [
     ## Every employee must have exactly one ID
     sh:path fr:hasID ;
     sh:minCount 1 ;
     sh:maxCount 1 ;
     sh:datatype xsd:string ;
            sh:pattern "^[0-9][0-9][0-9]-[0-9][0-9]-
[0-9][0-9][0-9][0-9]$" ;
    ] ;
   sh:property [
     ## Every employee is a manager or a worker
     sh:path fr:employeeType ;
     sh:minCount 1 ;
     sh:maxCount 1 ;
     sh:datatype xsd:string ;
     sh:in ("Manager" "Worker") ;
    ] ;
    sh:property [
      ## If birthyear supplied, must be 2001 or before
      sh:path fr:birthYear ;
      sh:maxInclusive 2001 ;
      sh:datatype xsd:integer ;
    ] ;
    sh:property [
      ## Must have a title, may have more than one
      sh:path fr:hasTitle ;
      sh:datatype xsd:string ;
      sh:minCount 1 ;
    ] ;

    sh:or (
      ## The President does not have a supervisor
      [
        sh:path fr:hasTitle ;
        sh:hasValue "President" ;
      ]
      [
       ## Must have a supervisor
         sh:path fr:hasSupervisor ;
         sh:minCount 1 ;
         sh:maxCount 1 ;
         sh:class fr:Employee ;
      ]
      ) ;

    sh:or (
      # Every employee must either have a wage or a salary
      [
       sh:path fr:hasSalary ;
       sh:datatype xsd:integer ;
       sh:minInclusive 3000 ;
       sh:minCount 1 ;
       sh:maxCount 1 ;
      ]
      [
       sh:path fr:hasWage ;
       sh:datatype xsd:decimal ;
       sh:minExclusive 15.00 ;
       sh:minCount 1 ;
       sh:maxCount 1 ;
      ]
    )
    .
   }
```

This file says the following about instances of the class fr:Employee:

1. Every employee must have exactly one ID (object of fr:hasID), a string of the form NNN-NN-NNNN where the Ns are digits (this is the simple example requirement).
2. Every employee must have exactly one fr:employeeType triple with value either “Manager” or “Worker”.

3. Employees may have a fr:birthYear triple, and if so, the value must be 2001 or earlier.
4. Employees must have a fr:hasTitle and may have more than one.
5. All employees except the one with title “President” must have a supervisor (specified with fr:hasSupervisor).
6. Every employee must either have a wage (a decimal specifying hourly pay, greater than 15.00) or a salary (an integer specifying monthly pay, greater than or equal to 3000).

Here is some employee data:

```
@prefix fr: <https://franz.com#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

{
 fr:Employee
  a rdfs:Class .

 fr:emp001
  a fr:Employee ;
  fr:hasID "000-12-3456" ;
  fr:hasTitle "President" ;
  fr:employeeType "Manager" ;
  fr:birthYear "1953"^^xsd:integer ;
  fr:hasSalary "10000"^^xsd:integer .

 fr:emp002
  a fr:Employee ;
  fr:hasID "000-56-3456" ;
  fr:hasTitle "Foreman" ;
  fr:employeeType "Worker" ;
  fr:birthYear "1966"^^xsd:integer ;
  fr:hasSupervisor fr:emp003 ;
  fr:hasWage "20.20"^^xsd:decimal .

 fr:emp003
  a fr:Employee ;
  fr:hasID "000-77-3232" ;
  fr:hasTitle "Production Manager" ;
  fr:employeeType "Manager" ;
  fr:birthYear "1968"^^xsd:integer ;
  fr:hasSupervisor fr:emp001 ;
  fr:hasSalary "4000"^^xsd:integer .

 fr:emp004
  a fr:Employee ;
  fr:hasID "000-88-3456" ;
  fr:hasTitle "Fitter" ;
  fr:employeeType "Worker" ;
  fr:birthYear "1979"^^xsd:integer ;
  fr:hasSupervisor fr:emp002 ;
  fr:hasWage "17.20"^^xsd:decimal .

 fr:emp005
  a fr:Employee ;
  fr:hasID "000-99-3492" ;
  fr:hasTitle "Fitter" ;
  fr:employeeType "Worker" ;
  fr:birthYear "2000"^^xsd:integer ;
  fr:hasWage "17.20"^^xsd:decimal .

 fr:emp006
  a fr:Employee ;
  fr:hasID "000-78-5592" ;
  fr:hasTitle "Filer" ;
  fr:employeeType "Intern" ;
  fr:birthYear "2003"^^xsd:integer ;
  fr:hasSupervisor fr:emp002 ;
  fr:hasWage "14.20"^^xsd:decimal .

 fr:emp007
  a fr:Employee ;
  fr:hasID "000-77-3232" ;
  fr:hasTitle "Sales Manager" ;
  fr:hasTitle "Vice President" ;
  fr:employeeType "Manager" ;
  fr:birthYear "1962"^^xsd:integer ;
  fr:hasSupervisor fr:emp001 ;
```

```
  fr:hasSalary "7000"^^xsd:integer .
 }
```

Comparing these data with the requirements, we see these problems:

1. emp005 does not have a supervisor.
2. emp006 is pretty messed up, with (1) employeeType “Intern”, not an allowed value, (2) a birthYear (2003) later than the required maximum of 2001, and (3) a wage (14.40) less than the minimum (15.00).

Otherwise the data seems OK.

We load these two TriG files into an emply repository (which we have named **shacl-repo-2**). We specify the default graph for the data and the https://franz.com#ShapesGraph for the shapes. (Though not required, it is a good idea to specify a graph for shape data as it makes it easy to delete and reload shapes while developing.) We have 101 triples, 49 data and 52 shape. Then we run **agtool shacl-validate**:

```
% bin/agtool shacl-validate --shapes-graph https://franz.com#ShapesGraph --data-graph default shacl-repo-2
```

There are four violations, as expected, one for emp005 and three for emp006.

```
Validation report:             Does not conform
Created:                       2019-07-03T11:35:27
Number of shapes graphs:       1
Number of data graphs:         1
Number of NodeShapes:          1
Number of focus nodes checked: 7

4 validation results:
Result:
 Focus node:           <https://franz.com#emp005>
 Value:                <https://franz.com#emp005>
 Source Shape:         <https://franz.com#EmployeeShape>
```

```
                   Constraint                     Component:
<https://www.w3.org/ns/shacl#OrConstraintComponent>
 Severity:             <https://www.w3.org/ns/shacl#Violation>

Result:
 Focus node:           <https://franz.com#emp006>
 Path:                 <https://franz.com#employeeType>
 Value:                "Intern"
 Source Shape:         _:b19D062B9x221
                   Constraint                     Component:
<http://www.w3.org/ns/shacl#InConstraintComponent>
 Severity:             <http://www.w3.org/ns/shacl#Violation>

Result:
 Focus node:           <https://franz.com#emp006>
 Path:                 <https://franz.com#birthYear>
                                                      Value:
"2003"^^<http://www.w3.org/2001/XMLSchema#integer>
 Source Shape:         _:b19D062B9x225
                   Constraint                     Component:
<http://www.w3.org/ns/shacl#MaxInclusiveConstraintComponent>
 Severity:             <http://www.w3.org/ns/shacl#Violation>

Result:
 Focus node:           <https://franz.com#emp006>
 Value:                <https://franz.com#emp006>
 Source Shape:         <https://franz.com#EmployeeShape>
                   Constraint                     Component:
<http://www.w3.org/ns/shacl#OrConstraintComponent>
 Severity:             <http://www.w3.org/ns/shacl#Violation>
```

Fixing the data is left as an exercise for the reader.

Earth Day – Franz Inc. and

Geoscience Experts Recognize the Growing Importance of Semantic Knowledge Graphs for Earth Science

Semantically Linking Earth Observation Data Makes it FAIR for the Global Community of Geoscientists

In celebration of Earth Day, Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology for Knowledge Graphs, today recognized how AllegroGraph, its semantic knowledge graph technology, is playing an essential part in making data FAIR (Findable, Accessible, Interoperable and Reusable) for the geoscience community. Since the current understanding of earth science processes is largely based on earth observation and numerical model data, making this data FAIR for all geoscientists and technologists is critical to facilitate future knowledge discovery about planet Earth.

Collecting, storing, monitoring and analyzing data from the core of the Earth up to the atmosphere provides critical knowledge about the planet and how living things interact with it. Scientists and technologists gather information about Earth from a range of sources, including: satellites, air- ground- and ocean-based sensors, physical sample data, etc., which are all recorded at a variety of temporal and spatial resolutions and need to be represented on the web for the global scientific community to access and use. AllegroGraph’s unique semantic graph capabilities allow diverse and complex data sources to be easily integrated with full search and cross-dataset queries possible.

“Our most pressing global environmental challenges cannot be solved by a single organization,” said Dr. Annie Burgess, Lab Director, Earth Science Information Partners (ESIP). “Scientists require data collected across multiple disciplines, which are often managed by many different agencies and institutions. ESIP is a community of data and information technology professionals dedicated to ensuring those data are FAIR. To assist with that goal, the unique semantic graph capabilities of AllegroGraph are leveraged with the ESIP Community Ontology Repository, a community platform to manage and exchange terms and vocabularies that assists scientists to publish, discover and reuse data.”

“To address important marine research, there is a critical need for ocean observatories to share data in a way that is easy to discover, use and integrate,” said Carlos Rueda, Senior Software Engineer, Monterey Bay Aquarium Research Institute. “With this goal in mind, the Marine Metadata Interoperability Project developed the MMI Ontology Registry and Repository (ORR), which leverages AllegroGraph to provide powerful interoperable semantic services that make the content on the web interconnected in a meaningful way for both humans and machines to consume.”

“We are at an exciting stage where there is a critical mass of experts and organizations around the globe with similar goals as well as the realization that we need knowledge-intensive applications,” said Dr. Lewis McGibbney, Data Scientist, Jet Propulsion Laboratory, California Institute of Technology and Co-Chair of the NASA ESDSWG Search Relevance Working Group. “The semantic technology stack is a crucial piece for building intelligent apps for knowledge-intensive use cases within the geoscience area.”

“Semantic graph technology is particularly well-suited to address the complex data integration, data access and analysis challenges surrounding Earth data science,” said Dr. Jans Aasman, CEO of Franz Inc. “We are thrilled that leading geoscience organizations are tapping into the power of AllegroGraph to share Earth science ontologies and data. We look forward to continuing to work with the community and help forward their important projects.”

A recent Gartner report explains the importance of using semantic technology to drive value out of data and included AllegroGraph as a graph database to consider for semantic technology solutions. “Unprecedented levels of data scale and distribution are making it almost impossible for organizations to effectively exploit their data assets. Data and analytics leaders must adopt a semantic approach to their enterprise data assets or face losing the battle for competitive advantage." (Source: *Gartner, How to Use Semantics to Drive the Business Value of Your Data, Guido De Simoni, November 27, 2018.)* To view a summary of the report, go to https://www.gartner.com/doc/3894095/use-semantics-drive-business-value.

About ESIP
The Earth Science Information Partners (ESIP) is a community of innovative science, data and information technology practitioners. ESIP members catalyze connections across traditional institutional and domain boundaries to solve critical Earth science data stewardship, information technology and interoperability issues. Through this work, ESIP improves Earth science data management practices and makes Earth science data more discoverable, accessible and useful to researchers, policy makers and the public. Learn more at esipfed.org or follow @ESIPfed on Twitter.

About Monterey Bay Aquarium Research Institute

Monterey Bay Aquarium Research Institute (MBARI) encompass the entire ocean, from the surface waters to the dee seafloor, and from the coastal zone to the open sea. The need to understand the ocean in all its complexity and variability drives MBARI's research and development efforts.

About JPL

The Jet Propulsion Laboratory is a unique national research facility that carries out robotic space and Earth science missions. JPL helped open the Space Age by developing America’s first Earth-orbiting science satellite, creating the first successful interplanetary spacecraft, and sending robotic missions to study all the planets in the solar system as well as asteroids, comets and Earth’s moon. In addition to its missions, JPL developed and manages NASA’s Deep Space Network, a worldwide system of antennas that communicates with interplanetary spacecraft. JPL is a federally funded research and development center managed for NASA by Caltech. From the long history of leaders drawn from the university’s faculty to joint programs and appointments, JPL’s intellectual environment and identity are profoundly shaped by its role as part of Caltech.

About AllegroGraph

AllegroGraph is a database technology that enables businesses to extract sophisticated decision insights and predictive analytics from highly complex, distributed data that cannot be uncovered with conventional databases. Unlike traditional relational databases or other NoSQL databases, AllegroGraph employs semantic graph technologies that process data with contextual and conceptual intelligence. AllegroGraph is able run queries of unprecedented complexity to support predictive analytics that help organizations make more informed, real-time decisions. AllegroGraph is utilized by dozens of the top F500 companies worldwide

Semantic Knowledge Graphs are the Foundation for Artificial Intelligence

The foundation for Knowledge Graphs and AI lies in the facets of semantic technology provided by AllegroGraph. Semantic Graph databases provide the core technology environment to enrich and contextualized the understanding of data. The ability to rapidly integrate new knowledge is the crux of the Knowledge Graph and depends entirely on semantic technologies.

About Franz Inc.

Franz Inc. is an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology with expert knowledge in developing and deploying Knowledge Graph solutions. The foundation for Knowledge Graphs and AI lies in the facets of semantic technology provided by AllegroGraph and Allegro CL. The ability to rapidly integrate new knowledge is the crux of the Knowledge Graph and Franz Inc. provides the key technologies and services to address your complex challenges. Franz Inc. is your Knowledge Graph technology partner.

Webcast – Speech Recognition, Knowledge Graphs, and AI for Intelligent Customer Operations – April 3, 2019

Presenters – Burt Smith, N3 Results and Jans Aasman, Franz Inc.

In the typical sales organization the contents of the actual chat or voice conversation between agent and customer is a black hole. In the modern Intelligent Customer Operations center (e.g. N3 Results – www.n3results.com) the interactions between agent and customer are a source of rich information that helps agents to improve the quality of the interaction in real time, creates more sales, and provides far better analytics for management.

Join us for this Webinar where we describe a real world Intelligent Customer Operations center that uses graph based technology for taxonomy driven entity extraction, speech recognition, machine learning and predictive analytics to improve quality of conversations, increase sales and improve business visibility.

View the recorded webinar.