Digital Transformation using chunks as a simple abstraction above triples and property graphs

Dave Raggett

W3C/ERCIM,
email: dsr@w3.org

Abstract: In this position paper, I propose *chunks* as a simple way to reconcile Property Graphs and Semantic Graphs, that is inspired by the cognitive sciences, and by blending symbolic and statistical information, chunks is well suited for machine learning and human-like AI. Chunks provide an attractive solution for use in web-based federated enterprise-wide knowledge graphs as a basis for digital transformation. Chunk databases can be implemented in multiple ways, e.g., as graphs, as holographic memory or as artificial neural networks. This points to the synthesis of Deep Learning and Symbolic Cognition, and a roadmap to AGI.

Keywords: chunks, reasoning, learning, memory, human-like AI, AGI, RDF.

1 Digital Transformation

Digital transformation is the reshaping of businesses for the digital age. According to McKinsey [1], the best-performing decile of digitized businesses earns as much as 80 percent of the digital revenues generated in their industries.

1.1 The challenge of digital transformation

Businesses are seeking to become more efficient, more transparent and more agile through the exploitation of digital technologies throughout the enterprise. By transparency, I mean transparency of operations for greater self-understanding. Agility relates to an organisation's ability to respond to and exploit change.

Digital transformation and open markets of information services will require integration of heterogeneous systems and data models. Moreover, to do so at scale will necessitate the need to address continuous change and inevitable variations across different communities. The world lacks an effective solution for these challenges!

- Businesses need to integrate heterogenous information systems and formats, including SQL/RDBMS, Spreadsheets, CSV files, XML, Property Graphs, Linked Data, PDF files, etc.
- Change is inevitable and on-going
- Different groups use different ways of talking about things
- Software development is generally speaking expensive, error prone and time consuming

1.2 Incremental not revolutionary change

Many analysts and developers are familiar with the use of UML and Entity-Relationship diagrams as part of the design process. Many more are familiar with spreadsheets. Relational database systems (SQL/RDBMS) are widely used. Whilst people have heard something about semantic technologies, RDF and OWL are seen as difficult and hard to adopt. We need to find ways that enable businesses to incrementally implement digital transformation without the need for abrupt and risky change. The narrative needs to couched in business terms and not in the technical depths of RDF.

Traditionally, development starts with identification of the overall goals. This is followed by gathering requirements, preparing a design for the applications and associated data models, then development, testing, deployment and dealing with bugs and improvement requests. Relational databases and their application code are tricky to design, and hard to adapt as new requirements emerge. This acts as a brake on innovation and reduces the ability of businesses to quickly adapt to changing conditions.

Businesses are thus looking for better ways to support digital integration that can boost efficiency and transparency of operations, along with increasing the organisation's agility for respond to, and to exploit change. Graph databases are promising as they can flexibly can store data along with data models and other metadata. Knowledge graphs include data and its meaning, and can be used for both declarative and procedural knowledge.

Knowledge graphs can be considered as graph-based representations of data, data models, metadata, and semantics. In principle, knowledge graphs can expose services for local or remote applications, that are subject to role-based access control. A SPARQL end-point is just one possibility.

According to Jo Stichbury [2]:

> *Knowledge graphs are able to capture diverse meta-data annotations such as provenance or versioning information, which make them ideal for working with a dynamic dataset. There is an increasing need to account for the provenance of data and include it so that the knowledge can be assessed by its consumers in terms of credibility and trustworthiness. A knowledge graph can answer what it knows, and also how and why it knows it.*

Federated knowledge graphs are where graphs are split across databases held at different locations and managed by different groups, e.g., different parts of an enterprise, or different entities within a national health service.

1.3 Enterprise-wide knowledge graphs

Enterprise-wide knowledge graphs feature:

- An integrated store for enterprise systems
- Contain data, data models, semantics and metadata

- Federated across organisational units and geographic regions as required
- Can be searched in a style closer to natural language, analogous to smart Web search
- Can include services relating to the data

In future, we can look forward to greater integration of human-machine collaboration involving cognitive agents with general purpose Human-like AI, based upon combining graphs, statistics, rules and graph algorithms.

Digital transformation can be framed in terms of the concept of enterprise-wide federated knowledge graphs. The starting point is the idea of copying data in different formats; data cleansing and transformation; and loading into target databases for use by applications. Data cleansing and quality control involves validation against integrity models and reporting on rejected data. Data transformation includes: mapping data to a standard terminology, de-duplication after merging, and calculated values where needed.

This process is often used in data warehousing and was first popularised in the 1970's. There are potential advantages for deferred processing based upon retention of the raw data and associated provenance information. The modern idea of knowledge graphs offers considerable benefits compared to traditional databases: e.g., being able to describe provenance, versioning, full GDPR support, granular access control, and full-scale organisation-wide models as a basis for interoperability, etc.

1.4 Web-based access to knowledge graphs

Today it is common practice to use diagrams as part of the design process, e.g., UML and Entity-Relationship diagrams. These diagrams are standalone and not integrated into subsequent stages of application lifecycles. The time has now come to switch to a Web-based framework where such diagrams are dynamically integrated into enterprise-wide knowledge graphs held on servers across the enterprise.

Here are some requirements:

- Level of detail control for complex diagrams: hierarchical structure, search queries and views
- Machine interpretable serialisation format for diagrams
- Revision control – keeping track of changes - what, why, and who
- Re-use of master and reference data models
- Validation against organisational rules

Here are some opportunities for web-based tools:

- View/edit diagrams in web browser with live synchronisation for editing by distributed teams
- Diagrams saved and centrally managed in the cloud
- Integration as part of enterprise-wide knowledge graph and business processes
- The diagramming tools need to be reversible: diagram to knowledge graph, and knowledge graph to diagram.

Spreadsheets are ubiquitous, but hard for businesses to manage, moreover, they tend to grow to the point where they become difficult to maintain. It is time to wean users to a new generation of web-based spreadsheets where cells are connected live with enterprise knowledge graphs, using names rather than letters and numbers for identifiers.

Everything is now networked, so it is time to exploit that rather than persisting with the mindset of standalone files that dates back to the pre-Internet era. The main challenge is to make it easy to transition from existing Excel spreadsheets to knowledge-based sheets. I envisage a cognitive system that acts as an expert collaborator to migrate spreadsheets to web-based "knowledge-sheets".

2 Chunks and Human-like AI

The first section has introduced the challenges and opportunities for digital transformation using web-based access to federated enterprise-wide knowledge graphs. This section explains how knowledge graphs can be implemented using "chunks" as a simple abstraction above triples and property graphs.

Chunks are based upon work in the cognitive sciences on human memory and the idea of chunking information to make it easier to recall. A chunk is a typed collection of properties whose values are literals, references to other chunks, or sequences thereof. Literals include Booleans, numbers, strings and dates.

Chunks can be serialised in a text format that is simpler than JSON. Names for chunk types, chunk IDs and property names are written without quotation marks. Properties are separated with semicolons or line breaks. A syntactic shortcut is provided for simple relationships. Reserved names are prefixed with @. Here are some examples:

```
# a single line comment
friend f34 {
  name Joan
}
friend {name Jenny; likes f34}

dog kindof mammal
cat kindof mammal
```

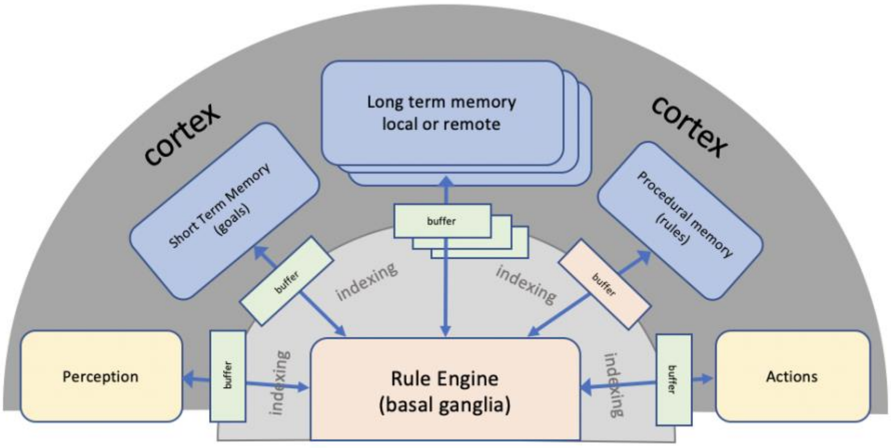
Where *friend* is a chunk type, *f34* is a chunk identifier, *name* and *likes* are property names, *Joan* and *Jenny* are also names. *likes f34* signifies that *Jenny* likes *Joan* via the link to the chunk for Joan. Missing chunk identifiers are automatically assigned when inserting a chunk into a graph. The last two examples are links.

Chunks combine symbolic information with sub-symbolic statistics. This can be used to mimic human memory when needed. This reflects the need to find the most relevant information in very large knowledge graphs, analogous to web search engines.

Chunks form part of a cognitive architecture for human-like AI [3]. The cortex is modelled as a set of chunk databases with associated graph algorithms. These are used by a number of modules for perception, emotion, cognition and action, reflecting a functional model of the human brain. The on-going research seeks to enable cognitive agents for human-machine collaboration, that:

- are knowledgeable, general purpose, creative, collaborative, empathic, sociable and trustworthy
- can apply metacognition and past experience to reason about new situations
- support continuous learning based upon curiosity about the unexpected
- have a level of self-awareness in respect to current state, goals and actions
- have an awareness of others in respect to their beliefs, desires and intents
- are multilingual and can interact with people using their own language

Chunk rules are inspired by John R. Anderson's work on ACT-R [4]. Rule conditions are evaluated on chunk buffers, whilst rule actions either update the buffers directly, or indirectly by invoking asynchronous operations exposed by cortical modules. Built-in operations include creating, recalling, updating, and deleting chunks, analogous to CRUD operations with HTTP. Applications can define additional operations using a graph API.



Cognitive Buffers hold single chunk
Analogy with HTTP client-server model

The work is being done under the aegis of the W3C Cognitive AI Community Group [5]. There is extensive documentation, an open-source library and an expanding suite of web-based demos. For a formal specification of chunks data and rules, see [6].

2.1 Relationship to triples and property graphs

Each chunk corresponds to an *n*-ary term in RDF, where a set of triples share the same subject node. Chunks can be mapped to RDF in a similar manner to JSON-LD. Links between concepts can be represented either as properties whose values reference other chunks, or as chunks that name both the subject and the object for the link as properties. This latter form corresponds to how RDF uses reification to support annotation of links.

Property Graphs consist of nodes and links between them, where both nodes and links can have properties. Both nodes and links map directly to chunks. When designing a semantic graph, you decide whether to represent a relationship as a property or a link.

Chunks provide a convenient framework for knowledge graphs that is easier to work with than existing RDF serialisations. This is appealing when it comes to digital transformation and winning over existing developers who perceive RDF and OWL as hard to understand and to work with, and would prefer to keep using the kinds of diagrams they are used to.

Whilst the Semantic Web focuses on formal semantics and logical deduction, in many industry use cases, developers are more interested in using applications to manipulate graph representations of information. The semantics are implicit in the code and the description of the concepts and relationships.

Human-reasoning itself is not based upon formal semantics or Bayesian statistics, but rather on working with examples, and the use of analogies for insights based upon past experience, see [7]. Human-reasoning is much more flexible than logical deduction, and embodies many different forms of reasoning where statistical evidence is important. Examples include induction, abduction, planning and causal reasoning.

2.2 Chunks and machine-learning

Classical AI uses manually developed knowledge and has difficulties for scaling up. By contrast, deep learning with artificial neural networks has been very successful on the basis of huge amounts of data. Deep learning is highly effective on deep statistical correlations, but lacks an understanding of salience and semantics. We now need to find ways to support machine learning with deep semantics in order to break free of the bottleneck of manual knowledge engineering. Hybrid approaches using symbolic graphs and sub-symbolic distributed representations presents exciting opportunities for machine learning where you want to derive concepts and relationships from the evidence you are presented with along with prior knowledge and past experience.

References

1. McKinsey Digital, May 2019, “Twenty-five years of digitization: Ten insights into how to play it right”, https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/twenty-five-years-of-digitization-ten-insights-into-how-to-play-it-right
2. Jo Stichbury. “WTF is a knowledge graph?”, April 2017, https://hackernoon.com/wtf-is-a-knowledge-graph-a16603a1a25f
3. Dave Raggett, “Human-like AI”, ERCIM News 125, April 2021, https://ercim-news.ercim.eu/
4. J. R. Anderson, “How Can the Human Mind Occur in the Physical Universe?”, Oxford University Press, 2007, https://doi.org/ 10.1093/acprof:oso/9780195324259.001.0001
5. W3C Cognitive AI Community Group: https://www.w3.org/community/cogai/
6. Chunks and Rules, Cognitive AI CG report: https://w3c.github.io/cogai/
7. P. Johnson-Laird, “How We Reason”, Oxford University Press, 2012, https://doi.org/10.1093/acprof:oso/9780199551330.001. 0001