Nathaniel Christen

1530 Palisade Avenue #28R ♦ Fort Lee, NJ 07024
(347) · 827 · 1611 ♦ osrworkshops@gmail.com
or NathanielChristen@lingtechsys.com

EDUCATION

Simon's Rock (Bard) College *1997*
B.A. (Mathematics and Computer Science)
"summa cum laude"

George Mason University *2003*
M.A. (Linguistics)

University of Ottawa *2011*
PhD – ABD Philosophy, concentration Philosophy of Science ("All but Dissertation")
My dissertation, which I am still trying to defend, is available alongside refereed publications (see below). While in Ottawa I served on a committee that worked to develop a new publishing system for students and faculty, featuring interactive documents and open-access data sets. I expanded on this software in subsequent books and projects detailed below.

EXPERIENCE

East Manhattan School 1990 - 2003
Office Manager and Sometimes Teacher/Math Tutor *New York City*

- I also served on the school's Board of Directors.

University of Ottawa 2003 - 2007
Teaching Assistant *Ottawa, Ontario, Canada*

Route de Vin Antiques 2007 - 2012
Software Developer *Ithaca, NY*

- This was my family's business; my father was a collector of old science instruments and other antiquities. I implemented our website and custom software to initiate and manage eBay auctions, build online presentations about scientific tools (to locate patents, edit photos, and incorporate images into auction content) and digitally synced eBay auctions to the website.

Linguistic Technology Systems 2012 - Present
Software Development Engineer *Fort Lee, NJ*

- Linguistic Technology Systems is a pre-revenue startup which has worked to curate linguistics research and develop new publishing software in areas such as bioinformatics, Software Language Engineering, text mining, Computer Vision, Geographic Information Systems, video annotation, scientific computing, and database engineering.

Others

- At various times I worked as a teacher, at different levels from grade school through college (I taught a first-year calculus class, and lectured on topics such as Kant and cognitive science, lambda calculus, combinatory logic, and type theory); receptionist at a Physical Therapy office; IT consultant for a health-food store; and sales person for antique shops.

TECHNICAL SPECIALIZATIONS

Computer Languages	C++, LaTeX, Lisp, Java, C#, Clojure, Ruby, JavaScript, Haskell
Software Engineering	Front-End Development, Programming Language Implementation, Compiler Theory, Virtual Machines, NoSQL Database Design
Frameworks/Domains	Qt, wxWidgets and other front-end tools, cross-platform engineering, web/native hybrid application development, humanities computing, GIS
Tools	OpenCV (for Computer Vision), embeddable scripting languages (Clasp, ECL, AngelScript), Geographic Information Systems/Digital Cartography, NoSQL databases such as WhiteDB and HypergraphDB

PUBLICATIONS

Nathaniel Christen, "Comparing Kant and Husserl", Presented at the OISE Graduate Students' Research Conference (University of Toronto) 2008.

Nathaniel Christen, "A New Pluribus: Classification and Markup Topology for Civic Data". In *Proceedings of the Sixth Workshop on Semantics for Smarter Cities, a Workshop at the 14th International Semantic Web Conference (ISWC)* 2015, pages 85-101.

Nathaniel Christen, "Hypergraph-Based Type Theory for Software Development in a Cyber-Physical Context". In Amy Neustein, *ed.*, *Advances in Ubiquitous Computing: Cyber-Physical Systems, Smart Cities and Ecological Monitoring* (Elsevier, 2020).

Joseph Lehmann, Nathaniel Christen, Michael Barilan, and Israel Gannot, "Age-Related Hearing Loss, Speech Understanding and Cognitive Technologies". *International Journal of Speech Technology* [IJST] 24, pages 509-516 (2021).

Nathaniel Christen, From 'Naturalizing Phenomenology' to Formalizing Cognitive Linguistics (I-III). These papers were included as a supplemental data-set analysis accompanying the *Speech Technology* article on the previous line. Data-set links for the papers (part of the full archive deposited to the OpenScience Foundation) are listed alongside the titles.

Part I "Cognitive Transform Grammar". https://osf.io/m9wdt
Part II "Grounding and Center/Peripheral Relations". https://osf.io/b3jt7.
Part III "Externalism and the Interface Theory of Meaning". https://osf.io/6fd5x.

Amy Neustein and Nathaniel Christen, *Innovative Data Integration and Conceptual Space Modeling for COVID, Cancer, and Cardiac Care* (Elsevier, 2022). Nine chapters:

Chapter 1 "Introduction". draft
Chapter 2 "Data Structures Associated with Biomedical Research". draft
Chapter 3 "Data Mining and Predictive Analytics for Cancer and COVID-19". draft
Chapter 4 "Modular Design, Image Biomarkers, and Radiomics". draft
Chapter 5 "Types' Internal Structure and 'Nonconstructive' ('NC4') Type Theory". draft
Chapter 6 "Using Code Models to Instantiate Data Models". draft
Chapter 7 "Multi-Aspect Modules and Image Annotation". draft
Chapter 8 "Image Annotation as a Multi-Aspect Case-Study". draft
Chapter 9 "Conceptual Spaces and Scientific Data Models". draft

Nathaniel Christen and Amy Neustein, "Software Language Implementation, Linguistics, and Virtual Machines". In *AI, IoT, Big Data and Cloud Computing for Industry 4.0* (Springer, 2023), Section 5. This covers five chapters:

Chapter 20 "Multi-Component Interoperability and Virtual Machines: Examples from Architecture, Engineering, Cyber-Physical Networks, and Geographic Information Systems". draft

Chapter 21 "Virtual Machines and Hypergraph Data/Code Models: Graph-Theoretic Representations of Lambda-Style Calculi". draft

Chapter 22 "GUI Integration and Virtual Machine Constructions for Image Processing: Phenomenological and Database-Engineering Insights into Computer Vision". draft

Chapter 23 "The Missing Links Between Computer and Human Languages: Animal Cognition and Robotics". draft

Chapter 24 "GUIs, Robots, and Language: Toward a Neo-Davidsonian Procedural Semantics". draft

Also: while not officially published, my doctoral dissertation (penultimate draft) on *Phenomenological Reductionism* can be browsed here.

CODE PROJECTS

The following represent projects we have worked on at Linguistic Technology Systems, which in general fit together into an overarching framework focused on scientific computing and data-set preparation. Some of this technology is described and demonstrated in a series of annotated videos, located at the following URLs: on Digital Cartography, Computer Vision, and Virtual Reality.

- **SledProc/ChasmVM** – This project subsumes several others. SledProc is a Software Language Engineering (SLE) franework specifically designed for Data Publishing (the acronym is SLE for Data Publishing – Prototype Compiler/Runtime). SledProc is not conceived as a general-purpose scripting language, but (by intent) has more features than a DSL. Primary use cases would be implementing code libraries for deserializing raw data files – parsing non-standard formats as necessary, and/or traversing an XML infoset, JSON document hierarchy (or analogous processing steps for other serializations) – plus supplying algorithms for data-set analysis, and GUI components for maintenance/visualization. SledProc is paired with, and compiled to, ChasmVM (Channel/Syntagm Virtual Machine), which in turn is interpreted by an engine that can be included in source code fashion into C++ projects, such as science applications, or custom software built specifically for individual data sets. Targeting an embedded ChasmVM engine, libraries written in SledProc (or, conceivably, other front-ends) could be distributed as bytecode and provide consistent deserialization capabilities in a cross-platform manner, thus promoting data sets' "FAIR" accessibility and reusability. Meanwhile, SledProc has features such as dispatching Design by Contract, (SI) unit-decorated types/variables, "virtual calling conventions" that allow user-defined procedure-call protocols based on runtime reflection or compile-time declarations, and template metaprogramming with constructs that enumate dependent type systems. The goal of these features is that SledProc code is to some degree self-documenting, and, in general, helps to clarify the theoretical background, mathematical conventions, data-acquisition methods, and other assumptions and paradigms informing the research within which a given data set is contextualized. I have worked on migrating other projects I am describing here to the overall SledProc environment, where they could function as C++ modules with a bidirectional interface to SledProc scripts.
- **GIS, Video, and 360° Photography Modules** – Intended (as a primary use-case) for data sets including audio/video, GIS, and/or panoramic-photography content, these modules provide Qt-based GUI components. In the case of digital cartography: displaying interactive GIS maps and superimposing data layers indexed by latitude/longitude pairs (or other geospatial coordinate systems), as well as rending custom GIS "tiles" (the small pictures that form the basis of composite digital maps) which could be styled for constrained geographic areas (deferring to generic tile servers elsewhere). A video-player window in provisioned with graphical and textual annotation capabilities (implemented on top of QGraphicsScenes) and scriptability for controling video content (e.g., pausing on particular frames based on their annotations). The 360° Photography components can intercept web navigation requests from tags/pins in a "virtual tour", thereby enabling communication between embedded WebGL windows and host applications. In general, these modules support cross-referencing (e.g., geotagging video frames) and are interoperable, employing data from one module to guide the state of another (e.g., centering a digital map based on a video's geotag annotations).

- **HTXN –** A "Hypergraph Text Encoding" system. HTXN supports PDF and SVG document generation as well as publishing machine-readable versions of document text suitable for text-mining and interactive PDF viewers or e-readers. HTXN employs a binary coding scheme that compresses file size while allowing special characters as needed, and offers specialized code-points to represent text landmarks such as sentence and paragraph boundaries; as a result, an encoded byte-stream can offer unambiguous representation of natural-language text content, often without needing to examine associated standoff markup. Text landmarks (e.g., the positional indexes framing a single sentence) are mapped to PDF page coordinates for the benefit of interactive viewers (including a core SledProc module) for functionality such as mouseover highlights, or convenient copy-to-clipboard, of content such as single sentences. Multimedia cross-references also allow HTXN-encoded documents to microcite content such as videos, data set elements (e.g., tabular columns, or internal data types), or source code objects (such as procedures, classes, or code annotations).

- **XCSD –** A database-oriented image format. XCSD is a format for storing images and representing image data optimized for persisting image series in database instances. In particular, XCSD constructs pixel data in blocks organized from the image-center outward, attempting to minimize buffer-copy operations and maximize the likelihood that in-memory pixel runs are geometrically useful. XCSD furthermore supports non-chromatic channels for region/superpixel labeling, texture detection, and precomputed format conversions, as well as offering a variety of color-space models with an emphasis on textural segmentation requirements.

- **ConceptsDB –** A lightweight NoSQL data-persistence module. ConceptsDB implements a hypergraph database via a key-value store where keys are structured designations of hypergraph sites (nodes inside hypernodes, edge-labels, properties, and subgraphs) and values are pointers to raw-memory segments, with an API that can work either through key-indirection or directly on raw memory.

- **Social/Civic Informatics GIS, and CSV tooling for C++** This code was developed in conjunction with data sets related to environmental contamination in the New York metro area. There are several state and federal initiatives for tracking contaminant; a major US government program is the Toxic Release Inventory (TRI), which is updated anually. TRI is used as a data source for the New York City "environmental justice" program (EJNYC), and Linguistic Technology Systems worked on a prototype to extend this analysis, alongside the NYC "Zoning and Land Use" (ZOLA) GIS software, to the full metro region. The State of New Jersey had passed a law in 2009 mandating the formation of a "site remediation priority" formula to rank locations based on levels of polution hazards. Unfortunately, government agencies and private industry delayed the implementation, claiming that data aggregation needed to seed a ranking algorithm was too difficult. Examining the files contributing raw data – especially the TRI sources – I discovered numerous encoding and documentation errors that would corrupt any data sets incorporating TRI. I also found that the TRI data profile was more complex and hierarchical than expected for tabular/CSV data, and accordingly built new C++ code designed to process CSV sources in a rigorous fashion. Data for individual columns are dispatched to distinct C++ methods via function pointers, but preliminary code is inserted to handle situations where multiple columns are interrelated so as to serialize vectors or associative arrays; to dispatch based on preconditions such as nonempty or nonzero fields; to handle text content constrained to controlled vocabularies (suitable an enumeration type in the deserialized objects), and other functionality targeted at properly interpreting complex CSV files. Outside the scope of the original project, I will adopt this CSV library as a module for SledProc deserialization features.

- **An SVG framework for viewing annotated legal documents** Recently Linguistic Technology Systems was contract to assist with a legal case, which involved collating and annotating numerous documents. Building off of the PDF moduled mentioned above, I implemented code that parsed PDF annotations along with document titles, deriving a classification of the documents into thematic categories. After aggregating all files into chronological order, the code split the composite document into individual pages, translating them to a hybrid HTML/SVG format to be hosted online. The rationale for switching from PDF to SVG was to ensure support for the annotations: PDF viewers are inconsistent in their ability to render annotations, and the quality of popup text boxes can be poor. By contrast, after conversion to SVG – actually two superimposed SVG files with an HTML backing, for navigation – the annotations could be displayed in an enlarged format, with additional graphics clearly delineating the text span to which the annotation applied, and coding mouseover effects. Another benefit of the SVG approach is that each individual page acquired its own web URL, which was convenient for citing single pages within legal arguments. Also, an online page was generated that linked to the start pages of the original documents, along with checkboxes showing the topical categories and other document metadata. We found this to be an effective presentation format leveraging multiple file types, so the associcated generative and transpile code will yield a useful SledProc module.

- Data for the three above-referenced papers ("From 'Naturalizing Phenomenology' to Formalizing Cognitive Linguistics (I-III)") listed earlier, developed as analytic/supplemental materials for the article by Joseph Lehmann, Nathaniel Christen, Michael Barilan, and Israel Gannot (cited above). This data set includes a collection of linguistic samples drawn from a combination of linguistics literature (including such authors as Ronald Langacker, Jens Allwood, and James Pustejovsky) and published corpora, particularly a CoNLL (Congress on Natural Language Learning) corpus for Dependency Grammars. This data set introduced code for syncing PDF-based text documents with structured corpus data (in the form of language-sample collections) and also developed a markup formal for annotating language samples with speech/prosody markup, discussed in the original article on "Age-Related Hearing Loss, Speech Understanding and Cognitive Technologies". Currently the repo is stored at https://github.com/scignscape/ntxh/.
- A republished version of a data set originally developed by Hakob Avetisyan and Jan Holub in conjunction with their article "Subjective Speech Quality Measurement With and Without Parallel Task: Laboratory test results comparison" (International Journal of Speech Technology, 2018). The newer data set including additional code to create a self-contained application for viewing/hearing samples. This data set was later incorporated into the repository for the Cognitive-Linguistic content outlined in the previous item.
- Republished data concerning avian communications (bird songs) discussed in the *Ubiquitous Computing* chapter cited above. The new archive includes additional code for parsing and accessing the data; it was also bundled into the Cognitive-Linguistics and prosody/annotation data set.
- Code samples developed and analyzed for the paper "Building Queryable Data Sets: Combining Object-Oriented and Functional Type Systems for Dynamic Query Evaluation", which was originally submitted to a conference on Software Language Engineering (later the code base was folded into the prior repository).
- An archive of language samples, curated along with text-mining and analytic code, extracted from the Blackwell *Handbook of Pragmatics*. Currently stored at https://github.com/ScignScape-RZ/HandbookPragmatics.

Areas of Research

In my graduate school/PhD research, I was particularly interested in the connections between Cognitive Linguistics and Philosophy of Science. This includes rigorous examination of the scientific methodology and argumentational warrants for Cognitive Linguistic theory, but it also includes (in the reverse direction) how Cognitive Linguistics illustrates conceptual patterns and cognitive schemata which are a foundation for human scientific reasoning. I have focused in particular on the role of phenomenology (in the sense of Edmund Husserl and his followers) as a bridge between Cognitive Science (or Cognitive Humanities) and the Philosophy of Science.

More recently, I have explored approaches to grammar and semantics that could be applied to both natural and computer languages, considering programming language implementation as a sandbox for exploring (on a greatly simplified level) cognitive themes in natural language. On an empirical level, I am engaged in an ongoing project to develop corpora by extracting language-samples analyzed by linguists in research literature, using computer code which (in an early version) was published alongside a collection pulled from Blackwell's *Handbook of Pragmatics*, referenced earlier. In conjunction with co-authors from Johns Hopkins and Tel Aviv Universities (the IJST paper cited earlier) I also worked on developing new annotation tools adapted for Dependency Grammar (for corpora such as those from CoNLL, the Conference on Natural Language Learning) and intonation/prosody patterns.

With Linguistic Technology Systems, I was the sole programmer working on a diverse set of tools related to document and data-set curation/publication, including components for linguistics, text mining, bioimaging, Civil Informatics, Geographic Information Systems, Computational Humanities, and other facets/areas within scientific computing.

Lastly, in recent papers (such as Chapter 23 of the *Industry 4.0* text cited above) I have begun to address the overlap between Cognitive Linguistics and Computer Vision, trying to establish structural parallels between cognitive schemata relevant for linguistic understanding (and underlying situational/contextual perception) and visual processing, with applications to such topics and probabilistic object-proposals and textural segmentation.