Andrea Martinelli

andmartinelli@gmail.com | github.com/antiufo | +39 3402431264
Vicolo San Francesco 5, 38056 Levico Terme, Italy

Languages: C♯, Java, TypeScript, JavaScript, HTML, C, SQL, PowerShell, some experience with F# and C++.
Frameworks/tools: .NET, WPF, Mercurial/git, Node.js. Some experience with Win32 and Android development.

Education

2013	Exchange student at Technical University of Denmark (1 semester)	DTU
2009-2012	Bachelor of Science degree in Computer Science (110/110 with honors) Thesis: Detection and extraction of articles from web pages and partial RSS feeds with full feed generation	University of Trento
2004-2009	Diploma perito informatico (90/100)	"Marie Curie" High School – Pergine Valsugana

Past and current experience

Sorted by end date

2017-01 – 2017-12	Shaman.RequestMaker: a command line version of Shaman, extracts a table/sequence of data from a webpage and its columns/fields (along with pagination). It also learns to simplify a sample HTTP request by trying to remove one potentially unnecessary header/POST/GET/json field at a time (so that it can be scripted). It also allows conversion of tables from {HTML/CSV/SQLdump/JSON/protobuf/sqlite/Win32 listviews} to {CSV/JSON/Excel/plain text}. Custom table extractions can be performed by specifying list and field selectors manually.
2017-02 – 2017-12	Shaman.Cecil: tree-shaking and symbol renaming for .NET assemblies.
2017-02 – 2017-11	Shaman.Scraping: scrapes web sites, blogs, subreddits and facebook pages (along with their full history) to WARC files
2010-06 – 2017-11	Shaman, a set of tools/libraries for extracting data from the web. The main component provides both an infrastructure for writing and using strongly typed connectors to third party web sites using declarative annotations and CSS selectors, and a tool for automatically inferring the structure and the data model/schema of a website, which also generates the CSS selectors to extract the individual fields. A declarative C# class is generated for each entity type, which can be further fine-tuned. A data viewer can display data as tables, grids, lists or JSON, and custom queries can be written with an autocomplete menu, which are then converted to SQL and run against the database of dumped data.
2012-03 – 2017-11	Shaman.Backup: stores encrypted and compressed blobs on Azure, keeping track of duplicates and previous file versions.
2017-02 – 2017-04	Shaman.Dokan.Archive: Mounts a 7z/zip/rar/WARC archive as a file system.
2017-03 – 2017-03	Shaman.DomEmulation: emulates a browser-like environment and allows execution of JavaScript through Chakra and Shaman.Dom (HtmlAgilityPack fork), with the DOM implemented in C#
2017-03 – 2017-03	Notepad2-mod: forked to support autosave and autorecovery.
2015-09 – 2017-03	Shaman.BlobStore: provides System.IO-like APIs optimized for sequential reading and writing of large amounts of blobs, stored as batched packages.
2009-02 – 2016-12	Songr: a music player that aggregates a dozen of music search engines, providing playback and download capabilities. It automatically picks the best results based on quality, speed and reliability of the server. Performs metadata cleanup and entity resolution, and also acts as a player for local files. It includes an opt-in system for submitting anonymous data and stats for improving the application. Includes support for retrieving lyrics from various websites
2016-10 – 2016-11	Legislative text annotator/hyperlinker: I developed a library that parses plain text looking for legislative references, and determines for each of them the kind of law/date/law number(s)/section(s)/paragraph(s) it's referring to (taking into account things like abbreviations, different syntaxes, ordinal numbers, letters, ranges, first/last…) – along with the text range it comes from.

	The grammar was written in ANTLR, with some parts of the grammar generated from a higher-level custom format. The library can provide strongly typed object(s) representing each reference, or output the decorated text as XML. Regression tests were used to monitor the correctness of parsing results as the syntax support was extended and improved.
2015-08 – 2015-10	RoslynLinqRewrite: modified C# compiler that rewrites LINQ expressions as fast, procedural code with minimal allocations.
2015-03 – 2015-07	Software Engineer Internship at Google (Munich, Germany): I worked on the ads settings page of My Account. I don't remember all the details but among the other things I remember adding tristate support for the toggle button, implementing the sample ads carousel and implementing the new WhyThisAd (web, RPC boq nodes, soy template…), although I couldn't complete it because the underlying APIs were not ready.
2014-05 – 2014-08	Reviewed book "TypeScript Essentials" for Packt Publishing
2011-06 – 2013-10	Reddit Full Feeds: retrieves the pages of the most popular links on Reddit, detects the actual content (ignoring menus, navbars and ads), and builds a complete feed.
2011-05 – 2013-05	JSON C♯ Class Generator: identifies patterns inside a JSON example file and generates the corresponding C♯ classes.
2012-04 – 2012-04	Hackathon participation at University of Trento
2010-12 – 2012-02	Blogger at WindowsBlogItalia.com
2011-08 – 2011-09	WinForms components development at Xenos Srl
2011-10 – 2011-11	Track Folder Changes: displays a tree of files on the disk in real time, restricted to the changes that occurred since the application started up, highlighting changes, deletions and creations.
2011-01 – 2011-06	Windows Exports Comparer: lists the differences between two builds of Windows by comparing exported functions and localized strings in DLL files.