

自我描述

找到一份喜欢的新工作是一件愉快的事情; 遇到一位投缘的新伙伴是一段奇妙的旅程。为此, 考察一个人的经历, 资历等固然非常必要, 然而观察一个人的性格爱好, 他作出选择的内在力量及其对未来的追求更加重要。因此, 我首先就这些方面进行坦诚的自我描述, 并把它放在简历最前面。希望能够符合您的观念和标准。

本人性格目标明确, 专注执着。对于决定之事能始终保持极大热情, 甚至不惜牺牲很多人看重的功名利禄。

我与计算机领域结缘起始于1983年国防科大发明银河计算机的新闻, 它令我开始好奇计算机和人类思维的关系。因此1989年高考后我选择了国防科大计算机系, 从此一直学习和从事计算机专业。1995年硕士选题时, 我自主选择了研究自然语言编程并坚持至今。2007年我领悟到无线移动互联网将更加凶猛广阔地渗透冲击社会和生活, 而自然语言处理也迎来了巨大的发展机遇, 我开始从互联网的角度重新思考自然语言编程。

随后, 我掌握了大量web技术。于2007年开始掌握python和开源LAMP范式, 熟悉了zope, web2py等框架, 发现了nltk工具包, 实现了一个基于web的通用信息系统的原型(seer)。07年左右, jquery出现不久, 我刚一接触即惊叹于其优美设计, 认定它将大兴于世。12年, 我适时切入node.js和angular.js, 用它们完成了自然语言编程原型系统Hopen, 该系统运用了MEAN范式, 对比了多种设计方法, 试用了superfish menu, ace, x-editable, jsTree, dynatree, jqxSplitter以及部分bootstrap控件, 比如tab, modal等, 基于hopen的需求用自己的实现替代了某些不适用的控件, 比如tree, splitter, tab, editable等等。在考察试用requirejs, seajs和browserify之后, 最终采用了自己实现的twoside.js, 并开发了配套的gulpjs的插件。与此同时, 我被漂亮的coffee-script所吸引, 进一步提高了开发效率。从这些经历我发现本人已经培养出了对新的技术趋势比较强的预见性、洞察力和敏感度, 并为之自豪。

发现和创造需要机遇。我坚持自然语言编程十多年之后终于结出果实并获得了意外的收获。2011年年, 我设计出了兼具lisp和prolog特性的dao系统, 2013年发现了一种新的解析器设计和实现方法, 发布了Peasy。2012年一个偶然的机缘, 我脑海中突然如同天启一般呈现出了苦苦寻求的自然语言编程系统的整体图景。在基于这个图景利用web技术实现原型系统的过程中, 命运之神又额外赐予我另一个意外的成果: 太极语言。在实现太极语言过程中, 又非常意外地获得了一个重大的创新: 元编译。这些经历和事实令我深深体会到, 机遇的来临取决于命运之神, 个人完全难以预料和掌握, 能掌握的唯有坚持的态度和努力的程度。

大多数人随着年龄的增长逐渐失去了对技术和编程的热情, 我则正好相反。与年轻的时候相比, 我感觉到越来越多编程的快乐, 因为现在我钻研的是自己最喜欢的技术领域, 使用的是我最喜欢的语言, 工具比以前更加顺手, 技巧比以前更加熟练, 能更快实现自己的构想。一般人可能会为了追求生活而放弃理想, 而我虽然因为追求理想放弃了一般人所享受的生活, 但是很幸运生活反而回报给了我更多别样的快乐。如今, 当我创造了独到的设计或代码, 就象作家面对自己的作品, 心情非常愉悦。经过漫长的跋涉, 我能感觉到自己是一个发现了风景的旅行家, 到达了峰顶的登山客, 发现了油田的钻探师。我现在确信不但能够胜任日常的web研发工作, 而且一定能够为公司拓展新的技术空间, 开创新的前景。

我开发的项目

公开发布的项目

太极语言(taijilang): 太极语言综合了三种创新: 缩进及主流语法结合lisp的内核, 可动态定制扩展的语言, 元编译技术。太极语言破除了某些编程语言领域的流行观点和技术限制, 是高级语言领域自1958年lisp发明以来的最重大进步。虽然太极语言还刚刚问世, 但是我感觉它好象是打开了一扇门, 展示出了广阔的发展空间和巨大的发展前景。它诞生于结合web开发自然语言编程原型系统过程中意外降临的偶然机遇, 将来也会回馈和推动web开发和自然语言编程系统开发。

peasy: 演示一种简单而又强大的设计解析器的方法。

dao: 兼具lisp和prolog的特性, 综合逻辑范式和函数范式于一体的解释器和编译器, 基于continuation, 内置语法解析能力。

- **dao:** 早期的解释器实现, python版本, 从yieldProlog和pyrolog受到很多启发。
- **daonode:** node.js版本(使用coffee-script语言编写)。

twoside: 100行1kb使模块能同时运行在后端（node.js）和前端（浏览器）并支持模块化编程的工具。

gulp-twoside: 支持twoside的gulpjs插件。

splitter: 用于angularjs的一个简单的多窗格分隔条, 基于**bg-splitter**。

pyshin: 基于python的命令行shell。

killsola: 恢复被sola病毒打包成 exe 的文件。

玩具领域多项国家专利: 2011年, 我获得了很多玩具设计方面的创意, 一年内不依赖专利界人士和机构几乎以零成本个人独立申请了五项实用新型国家专利。

未公开的项目

seer: 一个基于网络的通用信息查询管理系统。

hopen: 自然语言编程解决方案的实现原型。在十多年的探索之后, 我确信找到了解决这一问题的方案。这个解决方案不但实际可行, 而且非常实用, 能发展出合适的商业模式, 存在着广阔的发展空间, 一定可以超过github, stackoverflow, csdn这些网站的流量和规模, 成为一个巨大的社区。

学习和使用过的web开发技术

英语能力: 英语通过了六级, 但那神马只是浮云。以下这些技能才是硬道理: 使用google（而不是百度）, stackoverflow, google groups, 直接从各种英文官网获取资料, 阅读英文文档、API, 源码、书籍, 博客....., 我已经完全体会到这种能力对于快速接触、进入和提高技术有多么重要。将它放在web开发技术第一位应该也不算太离谱。

git和github: 很久以前使用svn, 现在当然是git。

node.js: 关注了很久, 亲眼看到到一门技术成熟和兴旺起来。hopen的服务器语言环境。

expressjs: hopen中用它作为服务器端框架。

angularjs: hopen的客户端框架。

mongodb+mongoose: MEAN web开发范式的一环,hopen的数据库解决方案。

passport: hopen用passport做用户认证中间件。

jquery: web开发者都知道。

websocket, socket.io: 开发太极语言之前有一个关于web框架前后端双向绑定的思想, 为了了解它们与实现我的想法的关系而考察试用了这两种技术。

jade: 最喜欢的模板语言。hopen就用它。可能有人注意到了: 我喜欢缩进风格的东西:

py, coffee, jade, taijilang。感谢Guido, Jeremy, TJ。

css, sass, compass: hopen使用sass+compass。本简历用jade产生html, css排版, wkhtmltopdf工具转换, 用Ghostscript向pdf嵌入字体。

web相关测试技术

- **qunit**: 在seer项目中使用了qunit测试框架。
- **karma, mocha, jasmine, chaijs, selenium+protractor**: hopen项目使用了这些测试技术。
- **代码测试覆盖技术**: 现在还只知皮毛, 未曾实用。但是, 我相信如果发现有足够的驱动需求, 我能很快上手这类兵器。我认为这种态度和自信比实际掌握了多少种技术更为重要。不知道您的看法如何?

工作流自动化

- **gruntjs**: hopen项目使用grunt实现流程的自动化。
- **gulpjs**: 因为gulpjs速度更快, 更简单, 内置watch等优点, 在peasy, twoside等项目中转换到了gulpjs。

travis-ci: peasy项目使用travis-ci.com实现了持续集成。

bower, npm: 各种包管理工具。

Meteor: 看了一些文档, 读了一些代码, 好复杂, 目前还没看懂。

Derbyjs: 这个要简单多了。但就是比Meteor发展慢, 流行度低。

zope: python web框架的鼻祖, 重量级; 几年前还曾经用它和plone实现过一个系统。很遗憾地看到它夕阳西下...

web2py: 很好学好用的框架, 遵循pythong带电池的哲学, 用它部分地开发了seer。

学习过的编程语言

长期使用的语言

C/C++: 1993年-2007年主要使用的语言。

python: 2007年-2011年主要使用的语言。

javascript: 2011年至今主要使用的语言。taijilang以javascript作为目标语言。

coffee-script: 我通过coffee-script提高javascript开发效率。taijilang的预定义语法与coffee非常类似。

taijilang: 好比Chris Lattner自称唯一4年Swift经验的专家，也可以肯定我是使用太极语言编程时间最久的专家:)

熟悉的语言

lisp: 2000年精心研究过一年，非常喜欢它的优雅和强大。

scheme: 研读过HTDP和SICP,从中对于编程得到了最多的启发。

haskell: 学习过三次，至今没有掌握monad.....

rebol: 欣赏它的优雅精简。受它影响，太极语言最开始使用[]作为代码块定界符，发布前才改成更通用的{}。另外，太极语言有比rebol的parse功能更方便的定制dsl的能力。

golang: 正在研究将它作为taijilang编译的一个目标语言选项以及用它重写taijilang编译器的方法。

了解过但是没有深入的语言

当前最流行也是web开发最常用的三门语言，曾经试图进入，但是纯粹因为个人喜好没有继续。

java: 一方面是觉得它太罗嗦（语法繁琐，文件夹嵌套太多层次），另一方面可能是我感觉java生态太巨大。

php: 一直没有吸引我。

ruby: 同上。

虽然这些语言我未曾使用，但是我有自信只要给我一个月时间，我足以掌握任何指定的语言、框架。

其它了解过的语言：

smalltalk: 安装过squeak, 感受过它独特的UI和OO的风格。

J语言: 一门精巧的面向数学的语言。

对各种语言的广泛了解主要源自于我对自然语言编程的研究兴趣，同时，通过了解不同的语言，我也对软件系统的设计有了更深刻的理解，对于编程的理解上升到了一个新的层次。

读过的一些书

《道德经》：我从道德经得到了最多的软件设计和编程开发方面的心得感悟。

《测试驱动开发》：Kent Beck，对于提高我的编程效率帮助最大的一本书。

《拥抱极限编程》：Kent Beck，里面确实有很多东西都已经成为软件界的日常实践，我也算采用了一部分。

《人工智能手册》：人工智能专业学生的枕边书，我只有第三册。

《unix编程艺术》：Eric S.Raymond 很开眼界，很受启发。

《编译原理（龙书）》：第四版，我估计只能理解其中的百分之三十，太极语言大概只用到了其中的百分之十。

《语言学教程》：北京大学 胡壮麟等。

《自然语言处理综论》：Daniel Jurafsky,James H.Martin等。

《语法哲学》：丹麦，叶斯帕森，英语历史语法的巨著。蛮有意思。

《代码大全》：经典。

《咨询的奥秘：寻求和提出建议的智慧》：温伯格，虽然说的是咨询，好象软件设计也能从中得到借鉴。

《你的灯亮着吗？发现问题的真正所在》：温伯格的总是能令人更新观念。

《C++设计新思维-泛型编程与设计模式之应用》看的时候很震撼：原来C++还可以这样玩。后来用python的时候又很感叹：在一种环境下需要殚精竭虑的问题换个环境居然都不再成为问题。

《TCP/IP 详解卷一：协议》：协议总是最重要的，然而我看得还很不够。

《重构--改善既有代码的设计》：Martin Flower, 我很少照搬这本书里面的重构处方，但是非常赞同这本书体现的思想。值得写很多遍代码的问题才是有价值的问题，越多次重构的代码越是有价值的代码。这充分体现在Dao, Peasy和taijilang系列项目中

Lisp In Small Pieces: 设计dao的时候，有什么lisp原语不会实现了，就到这本书里查代码。

on lisp: Paul Graham，在这本书里YCombinator的创立者对于宏有很深入的解说，

HTML & XHTML_ The Definitive Guide

The Definitive Guide to javascript

CSS: The Definitive Guide: 要开发web应用，o'reilly的这几本书应该是必读书目。

求职期望

行业：互联网/移动互联网/电子商务。

职位：软件开发工程师，高级软件工程师，系统设计师，前端开发工程师，自然语言处理工程师，搜索工程师。

薪资：面谈。相比于薪资，我更关注工作环境的发展空间和学习机会。

学历和工作经历

1989-1993：国防科技大学计算机系本科

1994-1997：空军雷达学院硕士研究生（人工智能）

1993-1994，1997-2009: 就职于空军第一航空学院

2009-2013：全职研究自然语言编程

空军第一航空学院期间的工作经历说明：

在此期间，我业余时间一直在坚持探索研究自然语言编程，工作时间除了少量计算机课程教学，主要担负科研工作，涉及各种检测设备的研究生产，在大多数项目中我负责软件设计工作，主要使用C/C++语言，特别是LabWindows/CVI开发环境，也用到LabView，有些项目需要数据库技术。其中我参与的第一项科研任务是研制一台大型综合检测设备，使用了VXI及虚拟仪器技术，系军内首创，获2000年军队科技进步一等奖。因为此奖项我获得了当年评定副教授的机会，但是因为不喜欢做论文剪贴搬运工，向期刊杂志写些没有创见和实用价值的理论文章，事实上相当于放弃了这个机会。这可能算是体现我执着的事例之一吧。后来我又参与了很多小型装备检测项目的研制生产，有多项获得军队科技奖二三等奖。期间还参与了一个基于OpenGL的仿真项目，并曾代表单位到北京某研究所作为主要软件设计者参与一个大型的网络信息管理项目(使用Oracle和Visual C++开发)。因为我对待工作认真负责，专注技术，领导、同事和合作者都很信任我。他们后来还为我提供了不少参与其它项目的机会，老领导创业时也曾经邀我加入他的团队，但是因为测控领域不是我的研究兴趣和发展方向，我逐渐淡出了这个领域。部队转业安排时，我不希望工作在按部就班的政府部门或事业单位，开始全力专注于研究自然语言编程。

这段经历在周围人看来算是安逸和顺利，有些甚至称得上幸运。我自己也确实知道这是很多人所追求和喜欢的生活，但是并不符合我的个性。上述经历基本只能说明我的过去，与互联网领域没有太大的关系。我把它们放在这里以保持简历完整透明。

最后的话

非常感谢您能花时间看完这份篇幅较长且不合常规的简历。IT行业是一个高竞争、高风险的领域，不管是投资、创业、技术、人才，莫不如此。招聘就象是玉石行业中的赌石。我努力完整而充分地把一个真实的我展示在您的面前。他山之石，能否攻玉，我想您自有定评。谢谢。