

简历

姓名： 许明

性别： 男

毕业学校： 电子科技大学

学历： 本科

工作经验： 15 年

最近职位： CTO

联系方式：

Mobile: 186-2812-8964

Email: inshua@gmail.com

QQ: 595069081

BLOG: <http://www.cnblogs.com/inshua>



(期望薪酬 30,000-50,000 元。)

从业经历

本人毕业于成都电子科技大学(985、211 工程重点大学)。2001 年参加工作，就职于深圳阿斯克软件公司，参与交通银行(香港)分行人事管理系统的研发。2003 年在成都北大青鸟任培训讲师、离职时为教研组长。

2005 年，加入创业公司欣新科技(深圳)有限公司(现更名北斗时空)，从事 GIS 产品研发。先后任软件工程师、软件部经理。

主导完成产品：

- GPS 监控管理平台
 - 独立完成 Web 地图组件
 - Web 地图服务器
 - BS 客户端（前端、后端）
 - CS 客户端（.net WinForm 技术）
- 规则服务器
 - 规则设计器
 - 规则服务器
- GPS 通讯服务器
 - Erlang 编写，支持多种协议，含交通部 JT-808 JT-809 协议
- 出租车电招终端
 - WinCE .NET COMPACT FRAMEWORK
- 广告服务器

并获得相关专利一项（专利号 CN202548878U）。

公司团队最多时 15 人。

基于该平台本人领导完成的主要项目有：

- 深圳移动物流通运营平台
- 世能达（天津）公司宝洁服务车辆监控平台
- 湖南中烟工业公司 GPS 平台
- 湖南怀化金大地水泥厂 GPS 监控平台
- 湖南省渣土局渣土车监控平台
- 山东某油田 GPS 系统

- 深圳西乡街道办网格化管理政务系统（使用智能手机）

其余如出租车广告、垃圾车、物流配送、公交排班、公务车排班调度等等各运输行业，功能定制较少的小项目二次开发项目不备述。

在产品和项目中，开发有 Web 框架一套（使用到 DSL、AJAX 技术，动态语言、RESTful 理念，该框架是 D2JS 框架前身）；可用于 WinCE/Win32 的仿 iOS6 GUI 框架一套（完整的线程池、消息系统、组件树、自绘 UI、布局、窗口系统、动画）。

2013 年，任杭州软景科技技术经理，完成有：

- 医疗方面供应链整合 ETL 工具
- 药监码信息提取

2014 年，加入南京智精灵科技，完成有：

- 大量认知心理学方面 HTML5 游戏开发（cocos2d-js）
 - 游戏代码生成工具
- 北京民航总院认知心理实验室某心理测评项目

2015 年任上海某民航方面的创业公司 CTO，从 0 开始招人组队（团队规模最多时 7 人，主要阶段 3 名开发），在半年时间内以 3 个主力完成一个集 IM、eLearning、个人任务、电子书、QA 为一体的行业 APP。所有功能都是独立研发。

2016 年任成都阿凡达 CTO，同样从 0 组队，完成有：

- 功夫财经 APP
- 四川航空 E 学院
- 四川骐骥人力资源公司人格测评产品
- 四川航空 HR 系统（部分参与）

本人具备较为优秀的学习、颠覆与创新能力，是 d2js、jssp 及 molecule(2015) 框架发明人：<https://github.com/inshua/d2js>。（该框架亮点颇多，已经过数个项目检验，唯个人推广不力，尚未形成知名度。）

各项成果详见本文档附录部分。

技术经验

回顾软件开发方面，主要有三个阶段。

第一阶段。07年前，在学校编写小游戏、游戏修改器；在工作中，对 SDK 各个函数都稔熟于心，运用灵活行奇，能解决面临的各种技术难题，打字飞快，代码规范，函数名有如标准 SDK，同事视为高手、牛人，公司待之为骨干。为了引进新的理念，如 UML 等，常常与领导同事发生激烈讨论。努力学习和运用设计模式、编译原理、Win32 API、COM/DCOM、VCL 源码、Prolog、MVC、生产者消费者等等技术和理论，是为所谓的技术高手阶段。本阶段的最后成果是一个代码生成器，以及一套 UML 静态图+状态机的图形化编程工具的思路。对于很多这个阶段的人来说，汇编、操作系统、编译、破解等等，都是山峰。

第二阶段。担任管理后，对于营构项目，倾听需求，有了更深的认识。本人常与客户探讨业务问题，识别关键业务对象，梳理业务规则和目标，提出让客户满意的设计方案。值得一提的是，在探讨过程中我极少和客户谈技术与界面原型。由于自身担任管理岗位，客户也往往带有管理诉求，我非常理解代码即法律架构即政治（语出《代码 2.0》）等等理念。这个阶段对于软件质量的要求提升到很高位置，特别注重可维护性，对日志、服务续航方面，有严格的要求。招聘中能够发现潜在高手，工作中身先士卒，分享成就，与同事一起成长。虽然任管理一职数年，职务上直接对董事长汇报，经常参与公司战略讨论，但始终保持技术人本色，不摆谱，不脱离研发，和同事合作融洽，总是战斗在第一线。在研发管理中，

本人秉持最小化管理成本原则（道家所追求的无为而治），少让员工为管理开会，少让员工为管理写报告，并据此设计有一套轻省的流程管理措施。

第三阶段。在任管理岗位之前，因已精通 Javascript（到 2012 年，我用 JS 开发了一个 JS 的编译器、纯手工编码），并受 Erlang 影响，深入学习和理解 SICP 后，几年来，对于以往各种编程语言、技术，已经能任意支配。宜聚合则聚合，宜封块则封块，随心所欲，更无樊篱。因为很难表述，兹举数端。例如：再也不用静态变量，凡是需要用到这个信息的地方，会将它作为参数传入，如果依赖变量多，就用名目为 Context 之类的对象将其聚合在一起；变量基本不再改变值；先写功能，能运行了再写文档，能运行了再协商接口，诸如此类。跨越 UML，我也深入理解了数据库的关系模型，熟练驾驭关系演算，当临时表则临时表，当索引则索引，不再泥于 ORM 里数据库是编程语言的对象存储工具的理论，也不纠结于 NoSQL 好还是 SQL 好。这个阶段的程序理念主要是闭包、状态机/动作机等。此阶段已将硬件目为软件的载体，作为软件系统的一部分，而不再将操作系统等纯技术问题当做山峰。

在技术方面，本人理念超前，能破能立，有较好的学习能力、创新能力、实战能力和理论功底。此外，我也具备一般开发人员欠缺的产品设计能力。

管理方面的浅见

这两年接触了几家公司后，觉察到很多企业的团队管理水平较为糟糕。回顾之前在深圳的从业经验，本人对于企业管理和企业文化有了较为深入的认识。

软件开发是一个智力工作行业，软件开发人员，尤其是高手，给人的形象是看起来文质彬彬实则难以沟通驾驭的形象。这个行业非常需要后现代化的管理方式和文化，但是现在不少创业企业甚至连现代化意识不具备。

从企业目标来说，企业应当把人尽其才，引导员工成长作为一项重要使命。软件开发行业日新月异，开发者如果只会增删改查三板斧，必将难以胜任越来越丰富的开发需求。另一方面，开发能力的上升，带来的效益也是极为可观的。一个开发能力优秀的程序员，甚至抵得上十个普通程序员，而薪酬往往仅为后者的 2-5 倍。

从企业文化来说，应当明确确立现代化的意识。例如，不少企业不懂如何展开技术讨论，讨论往往变成下达指示，讨论主持者担心自己的权威受到挑战，讨论往往脱离了完善产品的方向。甚至有的企业纠结于讨论者态度是否端正，是否在鸡蛋里挑骨头，讨论者会不会做人。在这样的企业，现代化的对事不对人的观念没有深入人心，更缺乏后现代化的幻想和鼓舞的文化。这样的企业不懂如何进行技术决策，团队围绕产品各抒己见凝聚企业文化的重要活动最终流于形式。企业领导不懂技术选型，是常见的，但是又未掌握技术讨论这种行之有效的手段，只能通过人际关系，技术人员是否有信心、有魅力来做评价，导致企业长期沿用过时技术，停滞不前。

对于这些企业来说，招聘活动也流于形式。企业不懂如何识别潜在高手，更不懂如何在非名校出身的求职者中慧眼识珠，只关心员工之前是否从事过相关领域工作，或转通过其他朋友熟人介绍“高手”。选拔新员工，纯用现代化方式，例如出题考察，都有很大的识别出错几率，何况不考察能力，只通过他人介绍。

当企业感到难以驾驭开发团队时，便订立一堆复杂的规章制度，希望制度来约束团队，靠经济手段搞研发管理。最终事与愿违，企业变成了一个不交流技术的科技企业，一个不管产品好歹的开发队伍，一个按时上下班的机关单位。

有的企业看到高科技公司氛围宽松，又接触了敏捷开发等等理论，也试着在本公司尝试，但是由于不懂现代化观念，不了解后现代化思维，最终演变成一抓就死一放就乱。

有的企业不惜重金，从 BAT 等各大企业高薪聘请高手，结果高手往往因大企业惯性使然，采用笨重技术，笨重管理手段，研发效率低下，难以迅速实现创业目标。

本人缺乏气场，并不是那种天生的管理者。唯回顾以往的管理服务经验，发现自己之前打造的队伍和培养的人员，经验足资借鉴。在选拔人才、培养人才、营造研发创新型文化，打造出色产品等方面，都有一套切实可行的办法。

综上，本人自认是一个较好的架构师、较好的开发者，具备后现代化意识的 Team Leader，是创业企业较为理想的技术加盟者。

技术能力

semantic-ui css3 Delphi HTML5 EXTJS markdown mina 任务队列 消息队列 生产者消费者模型 MVC StackOverflow Javascript 常用算法
Java word **SICP** actor-message C# 纯回调 状态机 动作机 编译器 Oracle 抓包 hook 闭包 excel Postgres ppt visio photoshop
bootstrap.js iocp 深度广度遍历 Activiti jmap MAT linux socket 包处理
cocos2d-js GDI+ 网络协议 http 协议 304 Erlang 日志 7*24 小时服务器 高并发 集群 json PowerDesigner restful fiber ajax COM 二次开发 ant
VB.NET freemind redmine jsp 关系运算 asp.net jQuery 函数式编程
nashorn http-client JMeter xml commons 数据库优化 GIS 动态语言
ADO.net uml git svn pdfbox geotools mapx mapxtreme jsoup ikvm.net
WinForm wpf winapi ActiveX 正则表达式 pl/sql developer SecureCRT
Accord.net...

本人自己开发有 d2js jssp molecule 框架, <https://github.com/inshua/d2js>。其中 d2js 原型从 06 年起在项目中实用(替代 struts2); JSSP 13 年实用, molecule 15 年发明并实用, 这套框架个人认为可以取代 angular.js、react、spring + hibernate + struts 等框架。

最近自学过机器学习, 大致了解分类、聚类、回归、推荐中的常用算法, TensorFlow 也在了解中。该部分还不能投入工作中, 相信设计周边工具没有问题。

面试建议

希望不要考察 HashMap 和 HashTable 有什么区别之类幼稚老套且实际开发中毫无意义的问题。

建议的考察方法：

1. 我可以讲解 d2js 框架或其它产品成果，借此可以评定我对技术的认知深度
2. 可以出一些常用算法、架构、BUG 排查的试卷
3. 可以提出某个产品设想或已经成功的产品，让我分析产品如何设计，技术上如何实现
4. 可以提出你在某次开发中切身经历的问题，观察我对该问题如何认识和解决，该活动可以邀请成长中的其他员工来做

附：产品成果摘要

(以下成果公开渠道均可见到，不涉及公司机密。由于技术话题较多，**最好与知晓技术的人员阅读。**)

需要说明的是，各种软件之间存在极大的相通性。本人从事 GPS 长达 7 年，近几年因为行业规避先后从事过游戏、医疗 ERP、民航 APP、内容商城 APP 等各种领域开发，都可从容应对。尤其是在 APP 开发中，我对 iOS 编程所知不多，但同样能给出巧妙的 APP 架构、制定技术方案选型，并指导 iOS 开发人员提升其编程水平，如状态机、回调、并发、双缓冲、MVC、日志等等。又如下面 GPS 平台中提到的 Erlang 通讯服务器，同样的技术也可用于 IM 服务器、游戏服务器，Erlang 的 Actor-Message 并发模型，在 Scala 和 Python 语言中也有相应实现，在 Java 中也有类似的 Akka 框架。

以下欢迎从产品的设计和技术角度观察。

功夫财经 APP（2016）

“功夫财经”打造了一个“最强网红经济学家”的江湖联盟，由王牧笛、匡澜联合创始，集结了李大霄、马光远、王福重、时寒冰四大掌门，以及琢磨先生、胡润等数位财经界著名人士。

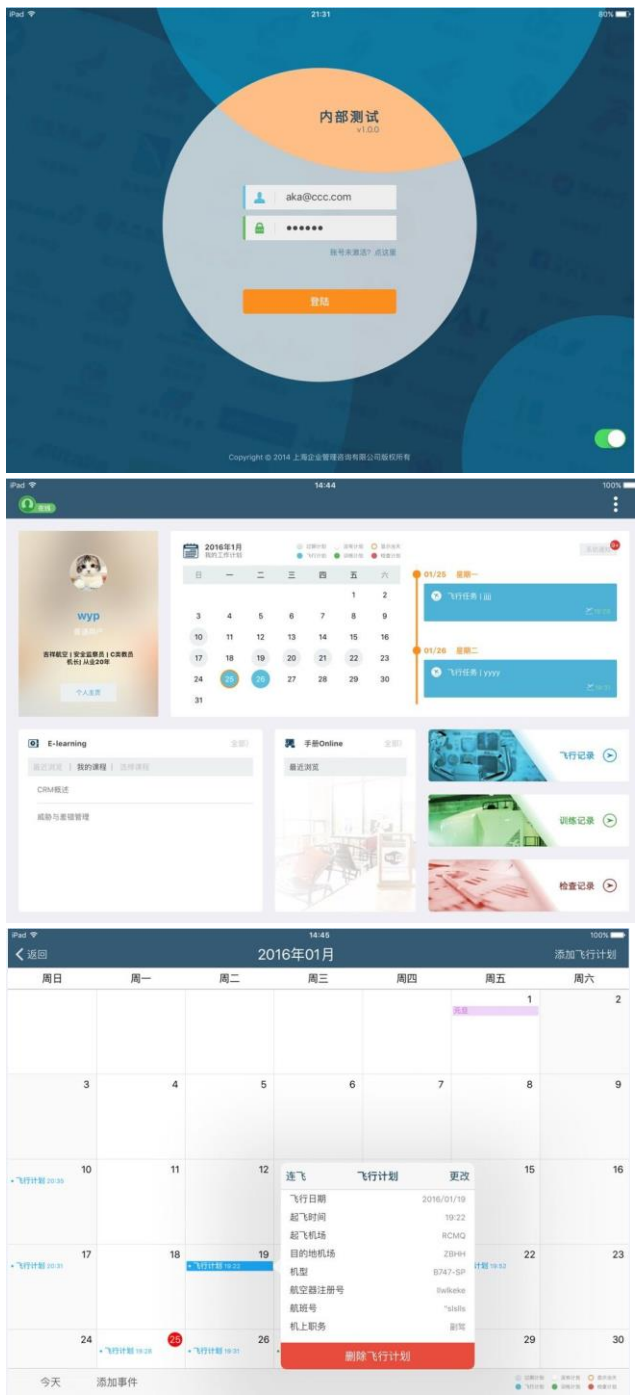
新闻参考：功夫财经获合一集团 1500 万人民币投资，视频网站入局财经内容市场_36 氪
<https://36kr.com/p/5047038.html>

<https://itunes.apple.com/cn/app/gong-fu-cai-jing/id1166858886?mt=8>



阿凡达公司承接了该 APP 开发，本人主要负责架构设计和部分后台功能开发、开发人员招聘、技术指导等。系统后台使用 PostgreSQL 数据库，其中大量应用 JSONB 数据类型，抓大放小，动静交织，使数据库架构大为精简，达到关系型数据库与 NoSQL 的优雅融合。由于该公司影响较大，此处不多介绍技术方案。

飞行员 APP (2015-2016)

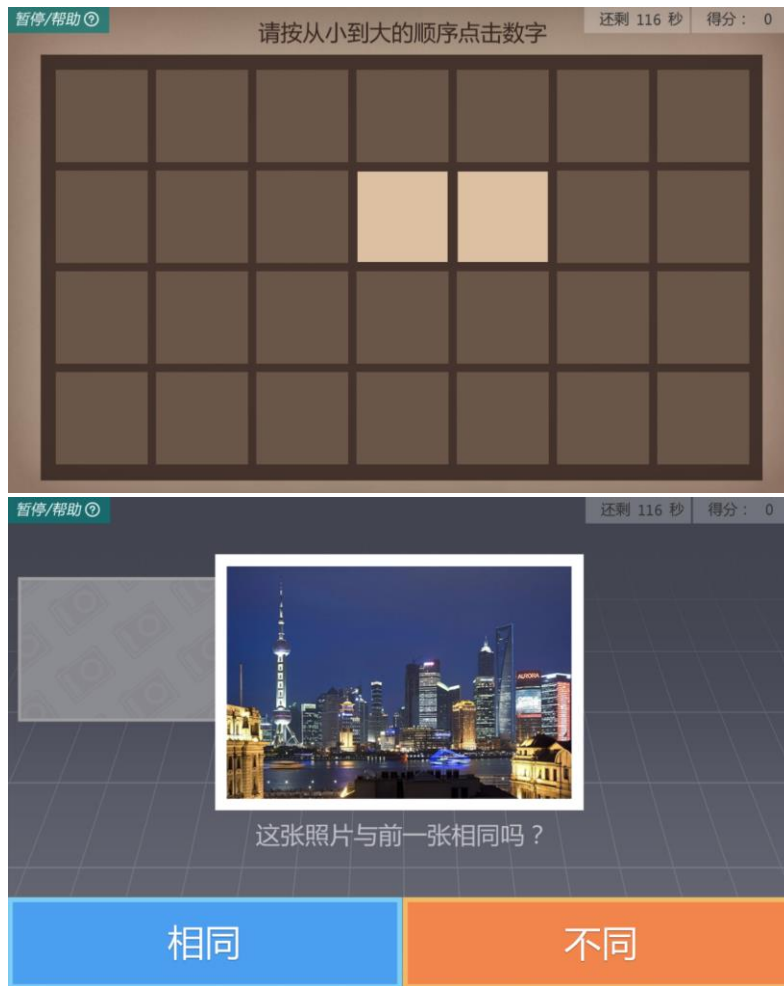




该项目从 0 开始，主体工期从 5 月到 12 月，主力长期 3 个人，最多 5 名开发人员，完成 IM、Elearning、电子手册、QA、飞行员工作计划及飞行员相关业务功能。各业务模块互相穿插，IM、Elearning 中可以查看手册，手册中可以访问 IM、QA 等等。

认知心理学游戏（2014）



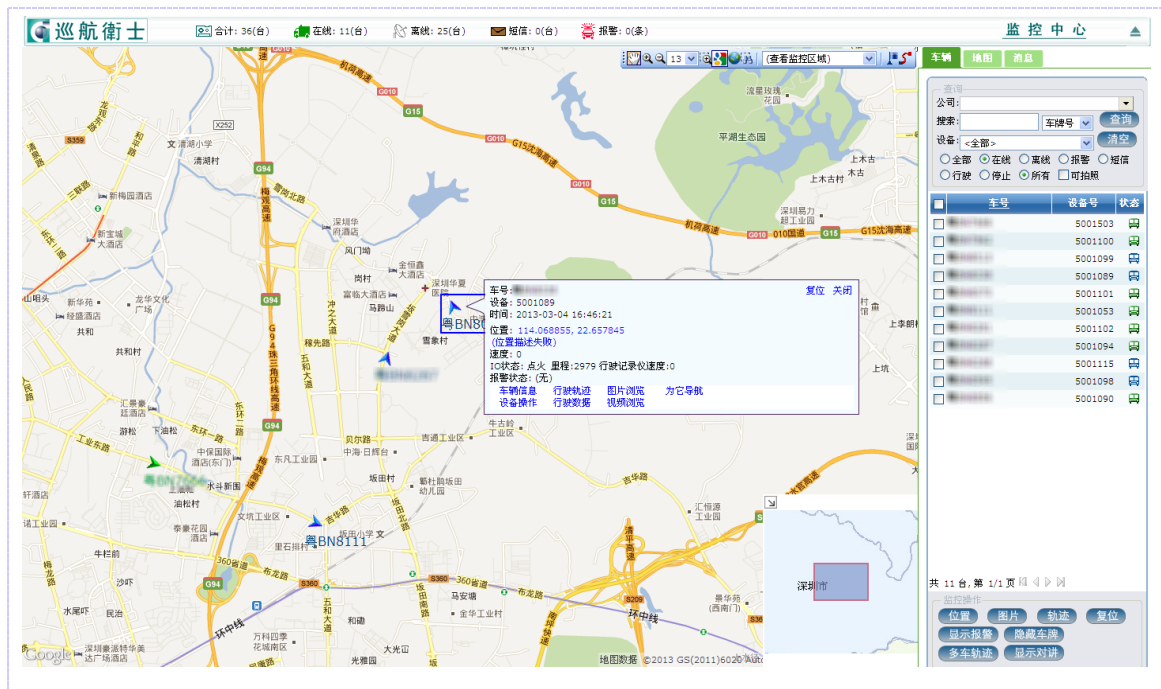


以上为认知科学健脑游戏。使用 HTML5 cocos2d-js 开发，适合平板电脑、大触摸屏。这些游戏看起来简单，每个游戏设计思路都不同，在我加入公司之前，一款游戏需要制作一周。有 Android 合作公司想改为 native 方式，花了一个月也未成功。

经过对产品开发周期的深入分析，本人设计了一个从 **PSD** 直接生成游戏源码的工具，可以生成字体、PLIST、精灵坐标、场景等等代码；以及一个从 **FLASH** 生成 **Cocos2D** 动画代码的工具。这两项工具将工期大大缩短，实现在 20 个工作日内完成 40 款小游戏。

GPS 运营平台、出租车终端

BS 网站（2005）

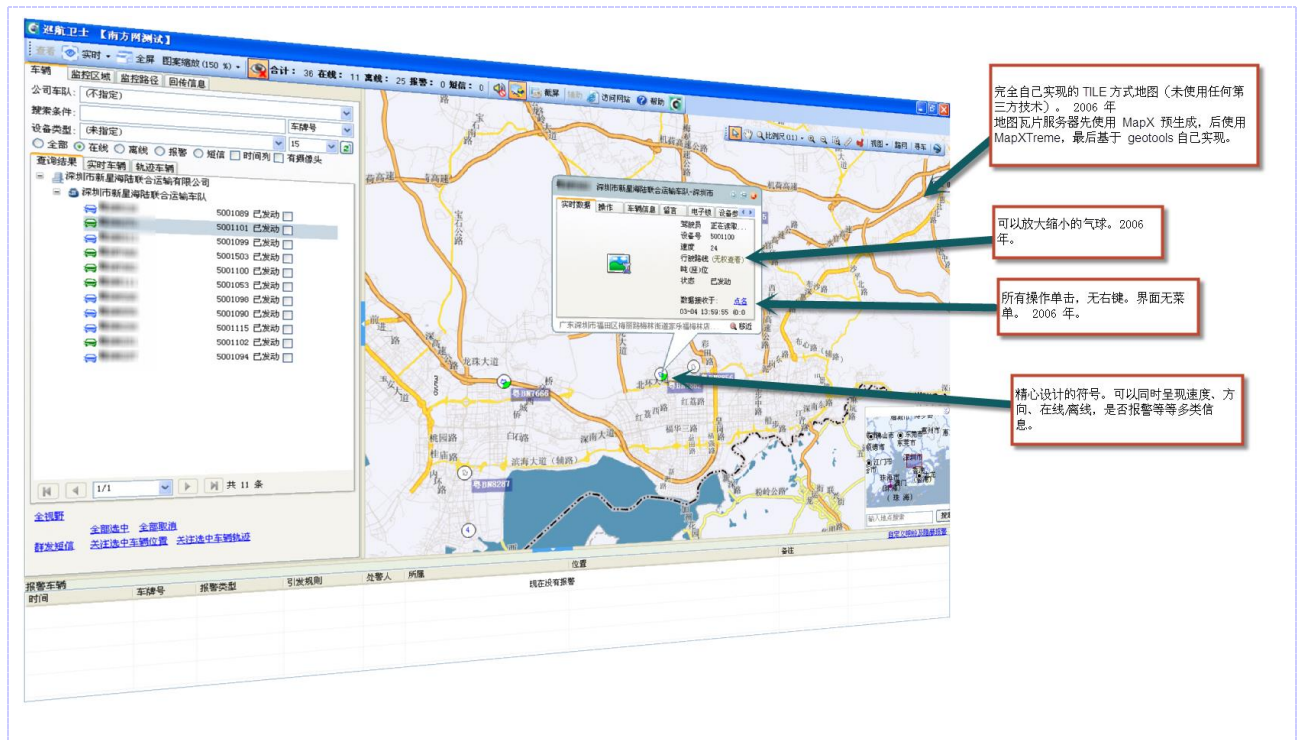


BS 所用地图为本人独立开发的 Web 地图引擎，在业界还在为 MapX 好还是 MapXtreme 好而纠结的时候（2006），本人即已独立研发出类似 Google Map 可以实用的 Web 地图技术。后来又通过包装加入了 Google 地图、图吧地图、四维地图等其它第三方 WEB 地图。

值得一提的是，该 BS 系统早在 2005 年即开始使用 AJAX 技术。

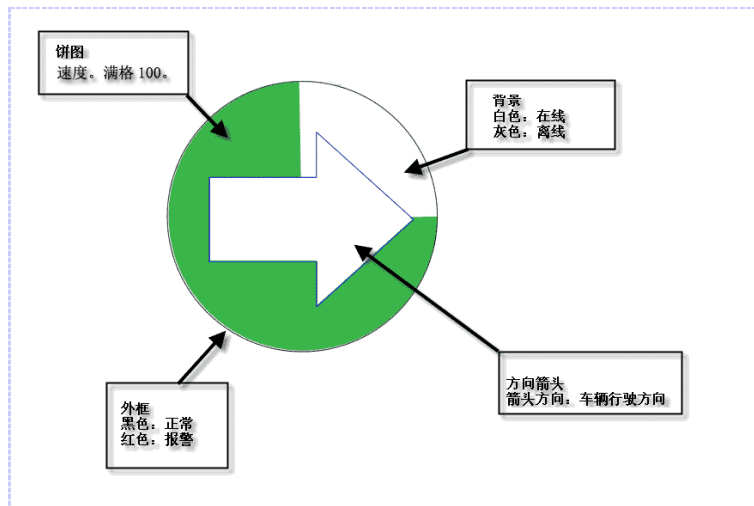
CS 客户端（2006-2008）

2006 年第一版，2007 年稳定版，后一直升级完善、做二次开发。



地图控件、树控件、气球控件都是自己编写。其中地图需要快速移动，需要达到较高的 FPS 指标，对图像动画编程有较高的要求；监控系统使用 MVC 思想设计，车辆数据每 5 秒轮询一次以更新数据。在更新画面时用户往往还在操作地图，需要对.NET 并发编程有较好理解，不然极易崩溃；整个界面设计不采用当时流行的菜单+MDI+对话框，全系统几乎没有子窗口，完全没有模式对话框，界面和操作都集中在一个监控窗口完成，简洁干净，操作集中方便，尤其单车操作分解清晰，使用气球的表达方式，生动活泼，本套界面设计除了对业务需要有较深的理解，更重要的是对 UI 表达能力的考验。

以车辆圆符号为例：



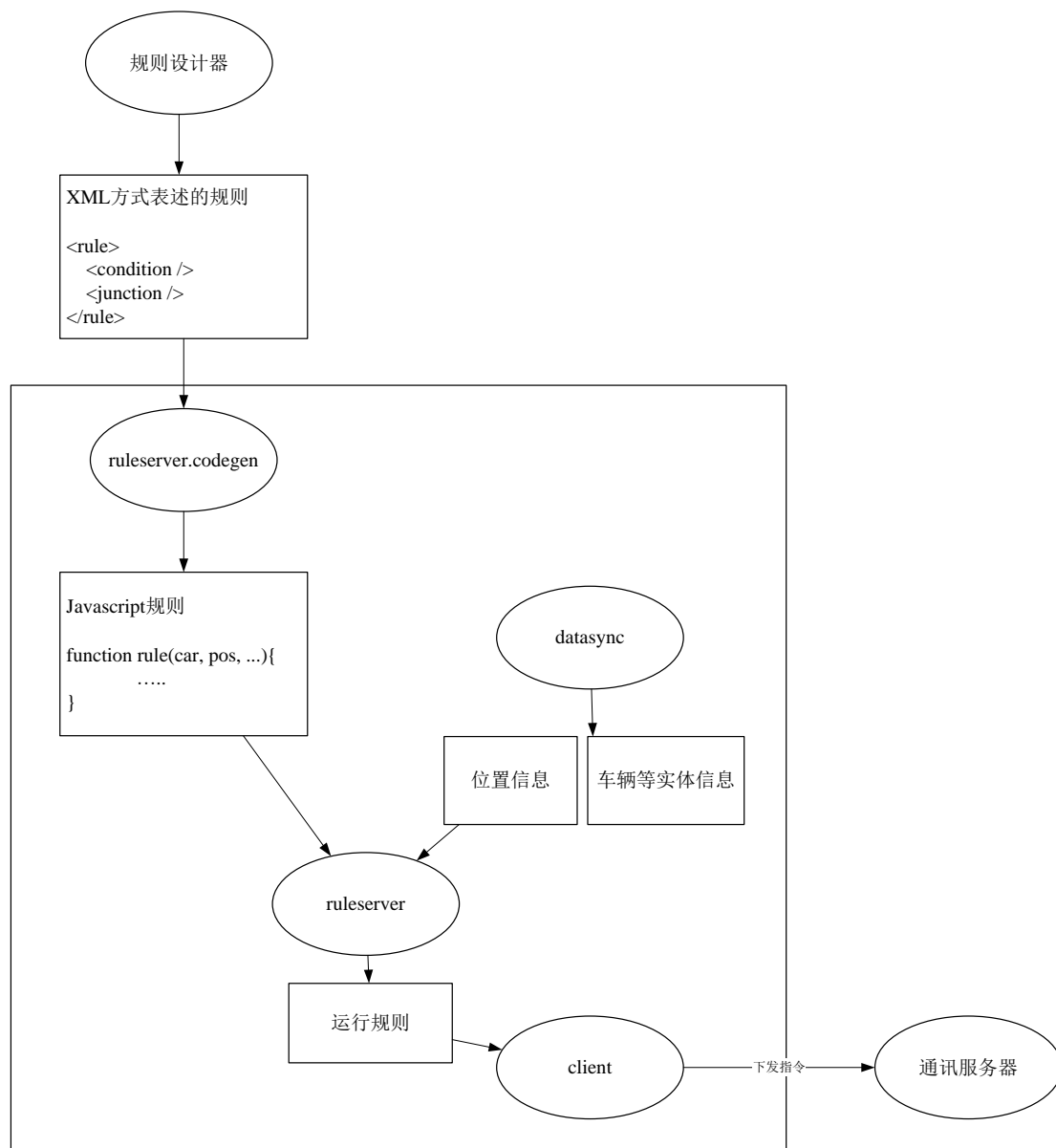
在一个 16*16 的圆中，集中展示出以上众多的信息。有些用户使用过用车辆卡通图标来表现车辆的系统，接触过这种设计后，就对表达内容不够丰富的车辆卡通图标失去了兴趣。

当时行业还不流行 UI 设计师, CS 系统 UI 系由本人使用 Photoshop 设计出效果后编程实现。

规则设计器及规则服务器（2007-2008）

规则服务器于 2008 年完成，该产品可以充分满足灵活复杂的业务需要。在不少项目中，例如对物流车的全程配送监控，都是使用规则实现的。这套 GIS 平台接入了多款常见厂商的设备，这些设备功能不一，例如越界，有的支持多边形检测，有的仅支持矩形，还有的无越界功能。通过规则服务器，这些终端纷纷具备了高级终端才具备的无上限个复杂多边形区域功能，且区域可以和各种条件进行组合，表达更复杂业务。

规则、条件、动作等等均系本人独立构想和实现。

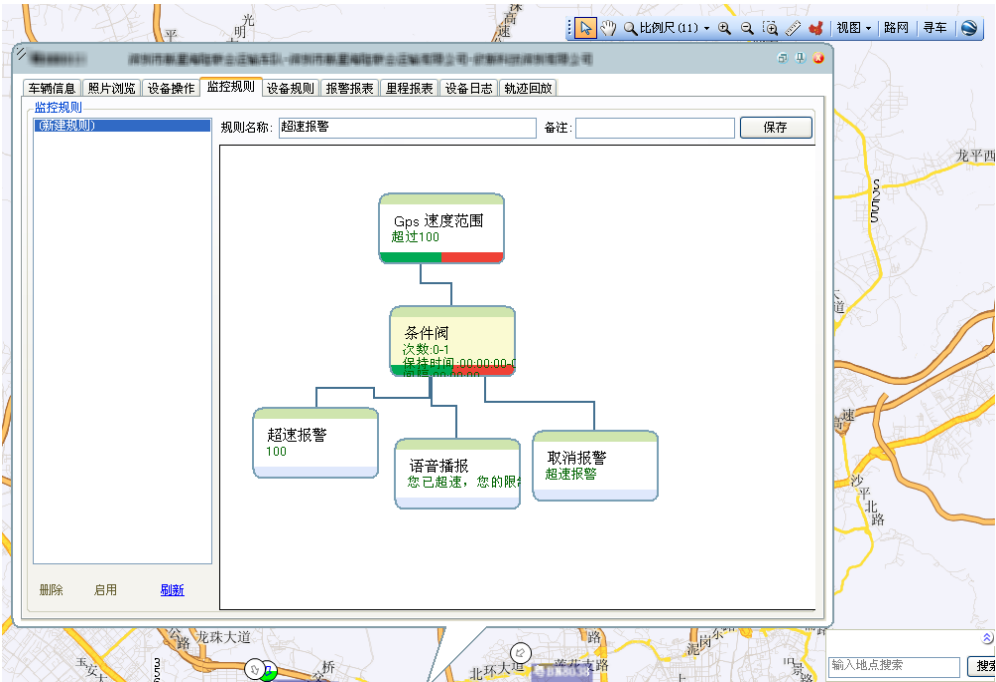


规则服务器主要逻辑过程

规则服务器使用 Java 编写，按设计器设计好的规则定义生成 Javascript 规则，并作为规则的容器运行这些规则。隔离是并发的秘诀。每条规则均生成一个 Javascript 函数，车辆、位置等作为函数参数传入，没有参数栈外的全局变量。函数调用，也即函数作用于参数，形成一个密闭域。各个调用栈互相隔离、无共享数据、无锁，保证了并发高可靠性。

一个精良的后台服务也需要一个优秀的 UI 呈现。

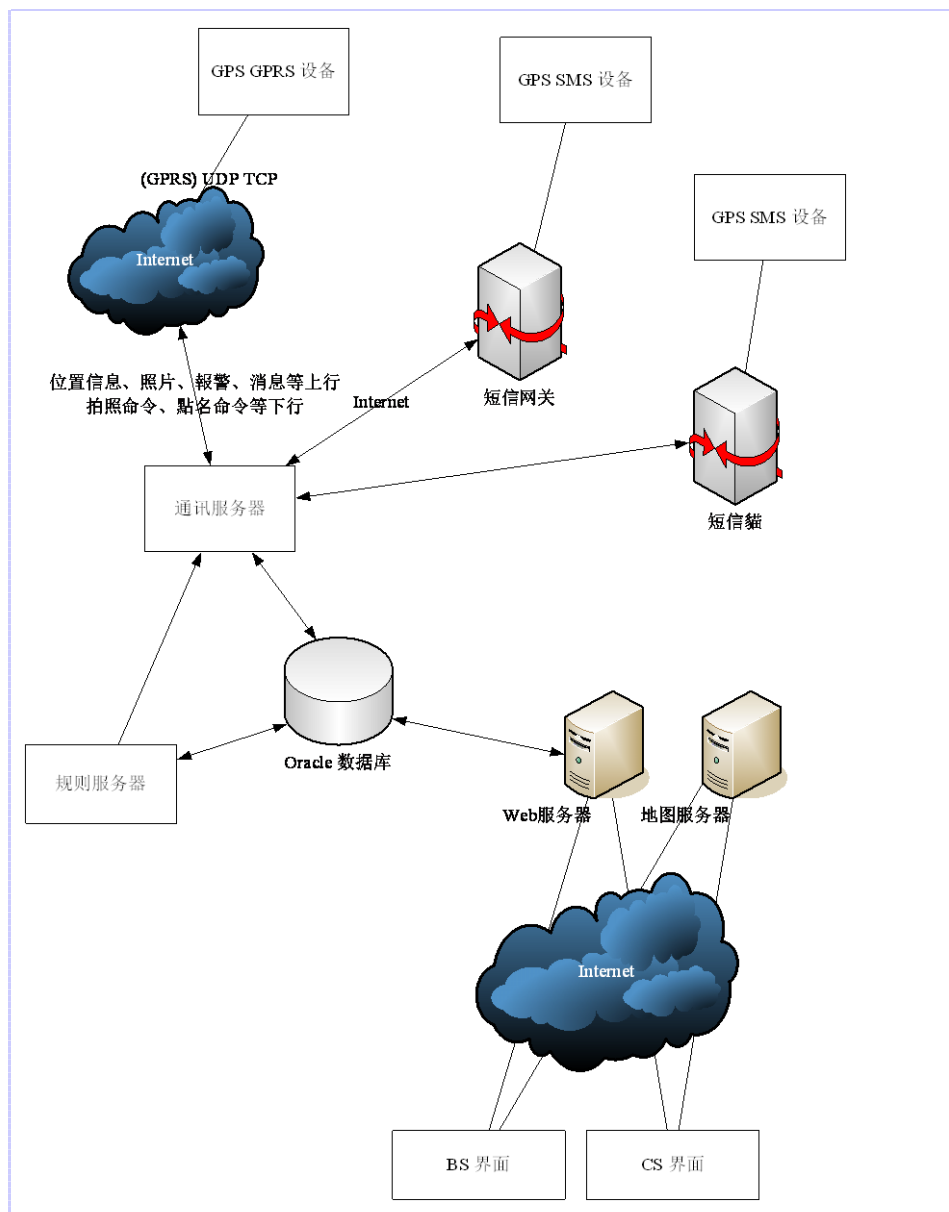
本人独立设计了如下规则设计器：



上图规则设计器使用.NET 开发。

除了生成 Javascript，针对一款 ARM9 车机还可以生成 C 代码编译为嵌入式 Linux 程序后下发到设备。

GPS 平台系统结构（2006）



通讯服务器原采用 Java 开发，支持接入多款厂商的 GPS 设备。2010 年开始采用更适合大规模并发的 Erlang 语言，现一直稳定运行。

CS、BS 均使用 **Web 服务器** 作为后台，一个功能两个界面。Web 服务器使用自主研发的框架。

通常，每个车每 15 秒产生一条记录，**数据库**存储数据量较大。通过优化，单个 Oracle 实例已能满足 5 万台车在线的运行需要，再大规模可通过 RAC 阵列扩充。所有以上服务器程序都可以按集群方式扩容。

目前，采用该套系统的运营商或公司企业在全国已有大大小小上百家。目前该平台已通过部标 JT-808 JT-809 协议（GPS 监控平台的部颁推荐性标准）认证。

WinCE 出租车电招终端（2011）



（图为深圳卫视深视新闻关于深圳物流博览会的新闻录像截屏）

该产品基于 WinCE 系统，从 0 开始搭建 GUI 框架，实现了仿 iOS 风格 UI 的效果，包括滑动、半透明、不规则元素、拖曳式 ScrollBox 等等各种 GUI 组件。这些组件在运用时代码流利清晰。下图为某界面的代码片段：

```

protected override Window CreateMainForm(Canvas canvas1) {
    Window form = null;
    form = new Window(canvas1, PadLayoutManager.Instance) {
        WindowState = Com.NewGlobe.Wince.iPhoneUi.FormWindowState.Normal,
        TextAlign = Div.TextAlignment.Center,
        Font = new Font(FontFamily.GenericSansSerif, 18, FontStyle.Bold),
        Color = Color.White,
        PaddingY = 12,
        BackgroundImage = @"images\form_bg.png",
        Text = "电招",
        Elements = new Div[]{
            new VBox(canvas){
                AutoHeight = true,
                Solid = true,
                //PaddingX = 22,
                BackgroundColor = ColorUtil.FromArgb(0xffdcdfe5),
                Rectangle = new Rectangle(20,45,300, 73),
                Elements = new Div[]{
                    new Div(canvas){
                        Opacity = 0,
                        Text = "乘客位置:", Location = new Point(20,20), AutoSizeWithText = true
                    },
                    new Div(canvas){
                        Opacity = 0,
                        Id = "txHailLocation", Text = "-", Location = new Point(20,40), WordWrap = true, Size
                    },
                    new Div(canvas){
                        Opacity = 0,
                        Id = "txHailPhone", Text = "-", Location = new Point(20,40), AutoSizeWithText = true
                    },
                    new Button(canvas, "拨打乘客电话", @"images\buttons\large_green.png", @"images\buttons\la
                OnClick = {(o, e) =>{
                    var btn = o as Button ;
                    if(! calling ){
                        AM.PhoneActor.Call(this, phoneNum, false);
                    }
                }}
            }
        }
    }
}

```

在设计中，该款产品围绕电招业务特点，出租车司机、乘客使用习惯，做出大量有趣有用的产品设计。例如，接受电招后，在出租车 LED 顶灯，会交替显示“电招”字样和乘客手机尾 4 位，这样一来，乘客很容易辨认出自己叫的车开过来了。

这款终端产品软件部分从构思和设计到技术实现，均为独立完成。并在 2011 年底，通过交通部 JT-808/T 认证。

Erlang 通讯服务器（2011）

2010 年，我领导将通讯服务器由 Java 迁移到 Erlang 语言，2011 年产品正式上线。测试中单台服务器可支持 10 万台车辆在线。运行数年，一直稳定工作，占用 CPU 不到 10%。

```

11%% ----- decode jt808 protocol packet -----
12
13decode(<<>>, _Handler, Session) -> <<>>;
14
15decode(Data, Handler, Tag) ->
16   Data1 = binary:replace(Data, <<16#7D, 16#02>>, <<16#7E>>, [global]),
17   Data2 = binary:replace(Data1, <<16#7D, 16#01>>, <<16#7D>>, [global]),
18   case Data2 of
19     <<16#7E:8, Command:2/binary-unit:8, _:2, 0:1, _:3, BodyLen:10, SimCard:6/binary-unit:8, DeviceSeq:16, Content:BodyLen/binary, _CheckCode:8, 16#7F>>
20       Handler(SimCard, Command, [DeviceSeq, {}], Content, Tag),
21       Remain;
22     <<16#7E:8, Command:2/binary-unit:8, _:2, 1:1, _:3, BodyLen:10, SimCard:6/binary-unit:8, DeviceSeq:16, PacketCount:16, PacketSeq:16, Content:Body>>
23       Handler(SimCard, Command, [DeviceSeq, {PacketCount, PacketSeq}], Content, Tag),
24%%     log4erl:info("remain:-p", [common:bin_to_hexstr(Remain)]),
25     Remain;
26   ->
27     Data
28   end.
29
30%% ----- encode jt808 protocol packet -----
31
32encode(Command, Seq, DeviceSim) ->
33   encode(Command, 0, 0, Seq, DeviceSim, <<>>, <<>>).
34
35encode(Command, Seq, DeviceSim, Content) ->
36   encode(Command, 0, 0, Seq, DeviceSim, <<>>, Content).
37
38encode(Command, Split, Seq, DeviceSim, SplitPkgInfo) ->
39   encode(Command, 1, 0, Seq, DeviceSim, SplitPkgInfo, <<>>).
40
41encode(Command, Split, EncryptType, Seq, DeviceSim, SplitPkgInfo, <<>>) ->
42   BodyProperty = <<0:2, Split:1, EncryptType:3, 0:10>>,
43   B = <<Command/binary, BodyProperty/binary, DeviceSim/binary, Seq:16, SplitPkgInfo/binary>>,
44   CheckCode = protocol:xorsum(B),
45   Body = <<B/binary, CheckCode:8>>,
46   Data1 = binary:replace(Body, <<16#7D>>, <<16#7D, 16#01>>, [global]),
47   Data2 = binary:replace(Data1, <<16#7E>>, <<16#7D, 16#02>>, [global]),
48   <<16#7E, Data2/binary, 16#7E>>;
49
50encode(Command, Split, EncryptType, Seq, DeviceSim, SplitPkgInfo, Content) ->
51   Len = size(Content),
52   BodyProperty = <<0:2, Split:1, EncryptType:3, Len:10>>,
53   B = <<Command/binary, BodyProperty/binary, DeviceSim/binary, Seq:16, SplitPkgInfo/binary, Content/binary>>,
54   CheckCode = protocol:xorsum(B),
55   Body = <<B/binary, CheckCode:8>>,

```


Jssp 网页技术（2013）

使用 Javascript 作为服务器端页面，基于 JDK 自带的 Javascript 引擎实现。



The image shows a JSP code editor window titled 'request_session_test.jsp' and a web browser window displaying the rendered page.

JSP Code (request_session_test.jsp):

```
1 <html>
2 <head><title>测试 REQUEST SESSION</title></head>
3
4
5 <body>
6 [%
7     if(request.name){
8         session.name = request.name;
9         session.email = request.email;
10        response.sendRedirect('request_session_test.jsp');
11    }
12 %]
13
14 <p>
15 [% if(session.name){ %]
16     你好, [%= session.name %], 我将写信给你 [%= session.email %]
17 [% } else { %]
18     请输入您的个人信息
19 [% } %]
20 [% ] %]
21
22 </p>
23
24 <form action="request_session_test.jsp" method="post">
25 姓名: <input name="name">
26 Email: <input name="email">
27
28 <button type="submit">提交</button>
29 </form>
30
31 </body>
32
33
34 </html>
```

Browser Output:

The browser window shows the title '测试 REQUEST SESSION' and the URL 'localhost:8080/testweb/jssp_test/request_session_test.jsp'. The page content displays the prompt '请输入您的个人信息' (Please enter your personal information). Below this, there are input fields for '姓名:' (Name) and 'Email:', followed by a '提交' (Submit) button.

D2js 及 molecule（2015）

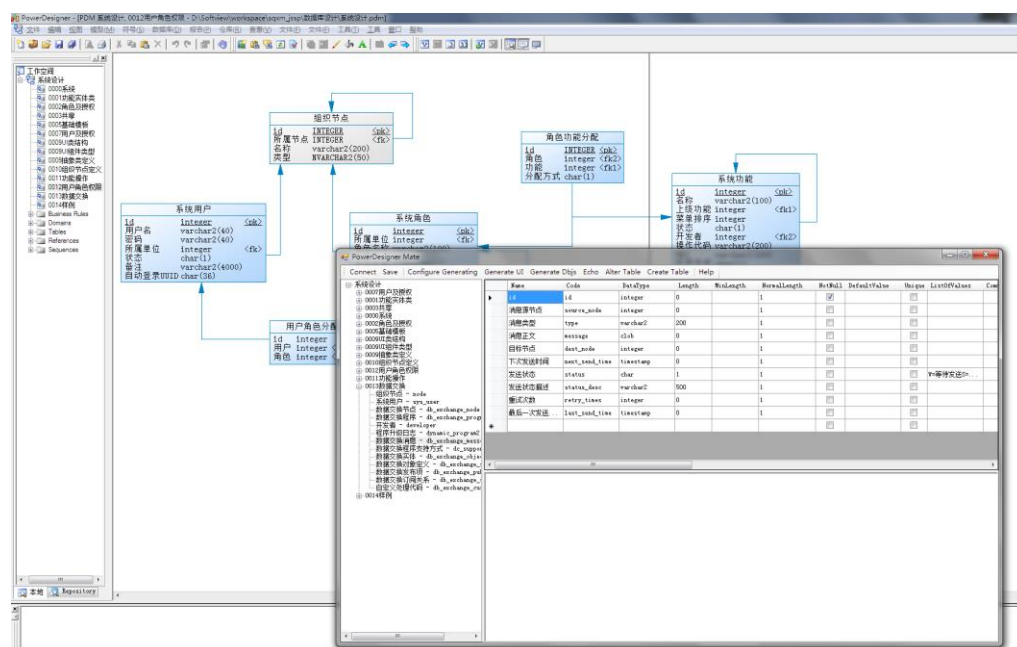
Book

Name	<input type="text"/>	Kind	武侠	Author	康德	<input type="button" value="Q Search"/>	<input type="button" value="+ Add"/>
<input type="checkbox"/>	ID	Title	Kind	Publish Date		动作	
<input type="checkbox"/>	1	纯粹理性批判	哲学	2015-07-28	EDIT	点击我	
<input type="checkbox"/>	7	红楼梦	小说	1977-02-03	EDIT	点击我	
<input type="checkbox"/>	2	判断力批判	哲学	1993-09-08	EDIT	点击我	
<input type="checkbox"/>	3	飘	小说	1982-05-06	EDIT	点击我	
<input type="checkbox"/>	10	射雕英雄传	武侠	1975-05-05	EDIT	点击我	
<div><input type="button" value="x Remove"/></div> <div><div><</div><div>1</div><div>2</div><div>></div></div>							

请详见 <https://github.com/inshua/d2js>。

值得一提的是，molecule 前端框架远远优于 angular、react 等市面上常见的前端框架，只要看第一个 molecule 的例子就可以发现，使用 angular 或 react 几乎无法实现，即使实现也很蹩脚。

基于 PowerDesigner 和 JSSP 的代码生成器（2013）



通过网格可以快速编辑字段、表结构、主键、外键、约束等。

连接 JSP 可以生成代码。一个 JSP 文件相当于一个模板文件：

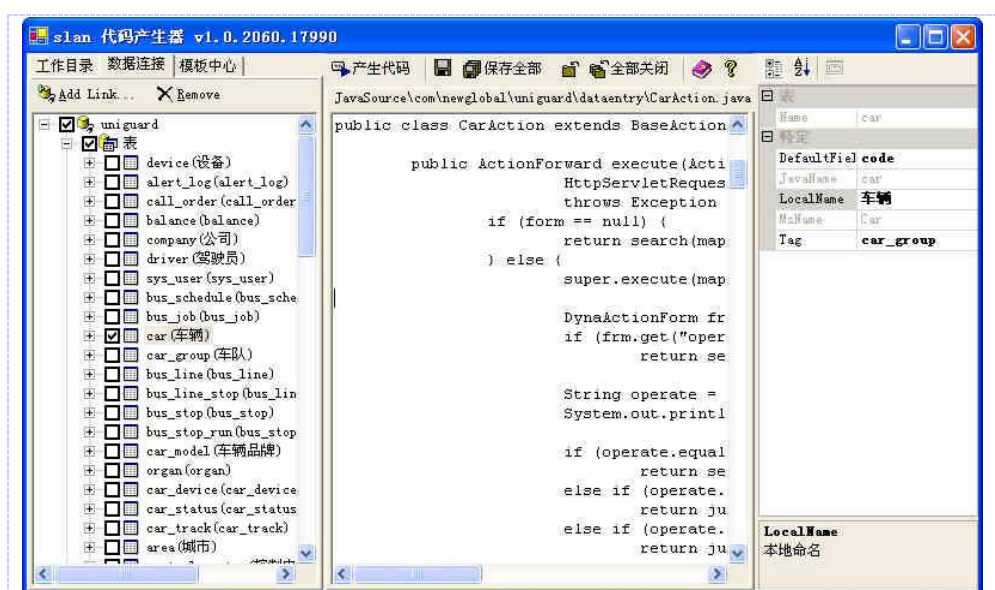
```

40
41   if(params.q){
42     if(!tb.nodeColumn){ sql += ' where 1=1 '; }
43     tb.columns.filter(function(itm){return itm.searchIndex && itm != tb.nodeColumn;})
44       .sort(function(itm){return itm.searchIndex;}).forEach(function(col){
45       cond(dbTypeToExtFieldType(col.type) == 'string', function(){
46         if(params.q.{$= col.code $} != null) sql += ' and instr({$= col.code $}, :{$= col.code $}) > 0';
47       }, function(){
48         if(params.q.{$= col.code $} != null) sql += ' and {$= col.code $} = :{$= col.code $}';
49       });
50     });
51   }
52
53   this.appendSort(sql, params.sort);
54
55   return {success : true, data : this.query(sql, params.q, pageDef)};
56 }
57 function validations(){
58   $V(rcd, ([ $= tb.columns.forEach(function(col){
59     $={$= col.code $} : [{$= col.notNull ? 'V.notNull,' : '' $={$= col.length ? 'V.longest(' + col.length + ',' : '' $={$= col.listOfValues ? 'V.inside
60     ($ )}; $}
61   })]);
62   $= $}
63 }
64 dbjs.create = function(params, rcd){
65   [% validations(); $]
66
67   rcd.{$=tb.pkColumn.code$} = this.nextId([%- tb.pkColumn.sequence || 'SEQ_ENTITY' $]);
68   actas('{$= tb.code $}.add', '[$= tb.code $]', rcd.{$=tb.pkColumn.code$}, rcd);
69   this.insertRow('{$= tb.code $}', rcd, [%- tb.columns.map(function(col){return col.code;}) $]);
70   return {type : 'exec', data : rcd};
71 }
72
73 dbjs.modify = function(params, rcd){
74   [% validations(); $]
75
76   actas('{$= tb.code $}.edit', '[$= tb.code $]', rcd.{$=tb.pkColumn.code$}, rcd);
77   this.updateRow('{$= tb.code $}', rcd, [%- tb.columns.map(function(col){return col.code;}) $]);
78   return {type : 'exec', data : rcd};
79 }
80
81 dbjs.destroy = function(params, rcd){
82   actas('{$= tb.code $}.delete', '[$= tb.code $]', rcd.{$=tb.pkColumn.code$}, rcd);
83   this.deleteRow('{$= tb.code $}', rcd);
84   return {type : 'exec', data : rcd};
85 }
86

```

其它个人作品

SLAN 代码生成器 (2005)



这个工具可以根据选中的表、字段，按照模板生成代码。模板可以使用 VB.NET

或 C# 语言编写，模板的观感类似 ASP/JSP，在目标代码中混杂模板语言代码。

例如：

```
<SLAN language="VB.NET">
For i As Integer = 0 To 10
    Print("a[%0] = a[%1]", i, i * 2)
Next
</SLAN>
```

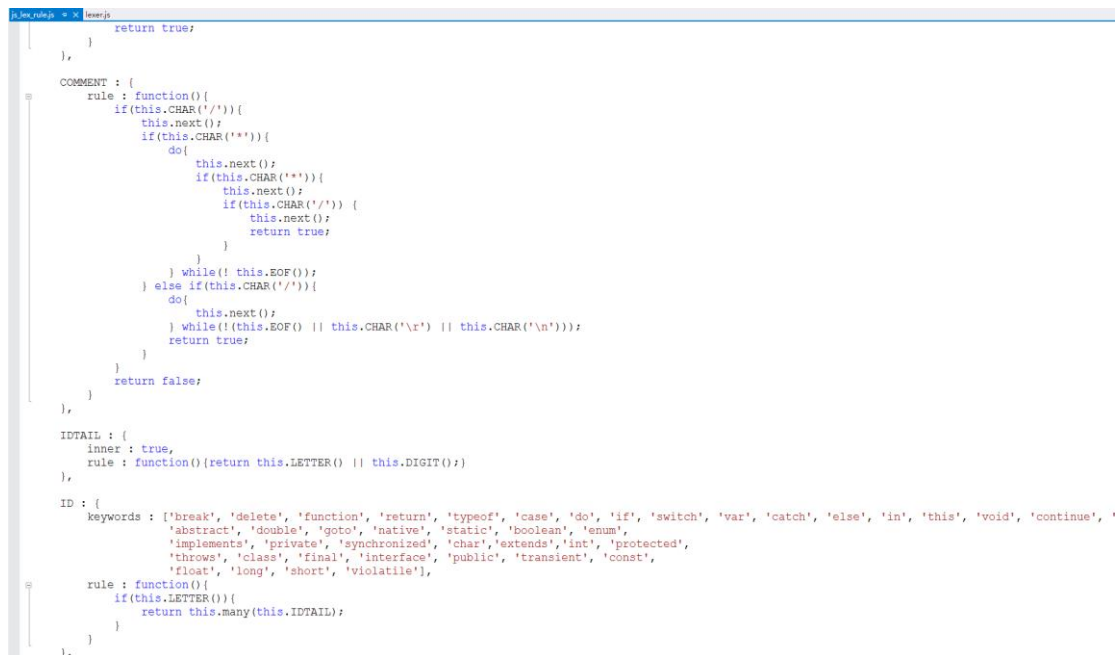
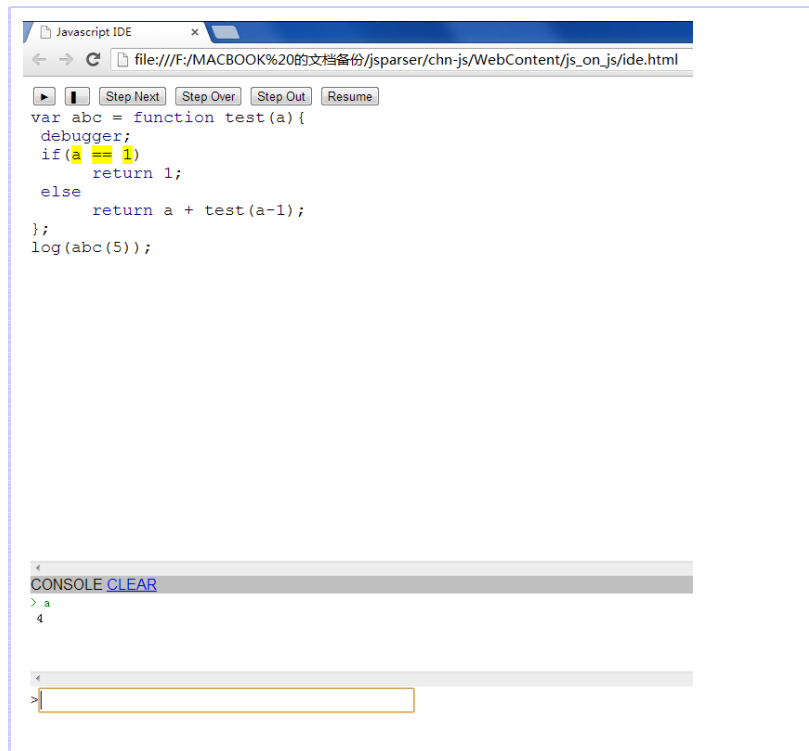
工具使用 CodeDom 将模板翻译为 VB.NET/C# 代码，编译为程序集，执行后生成目标语言代码。目标语言可以是任何编程语言和非编程语言。

后来我放弃了代码生成思路，就停止了这个工具的研发——事实上，JSP/ASP 就是唾手可得的代码生成器，node.js 还可以更灵活。

此外，在实际工作中，我也制作了不少特殊用途的小工具，此处不罗列。

js-on-js (2012)

<http://code.google.com/p/chn-js/>



使用 js 语言编写 js 语言的编译器（包括手工打造递归向下的语法分析器、词法分析器、编译器、自定义字节码的解释器、以及一个比较原始的 IDE 等），并设

计了一个中文化的 js 语言。约 3 周时间完成。值得一提的是，该中文化的 js 语言发明的一个语法设计不知因为何种机缘出现在三年后 Apple 的 Swift 的语言中。

```
判斷
    如果 a == 1 [[           // 後面必須接語句塊，即：不允許
if(cond) aa; 必須是 if(cond) {aa;}
        dadada
    ]] 否則 [[
        dididi
    ]]

循環
    如果 a == 1 反復[[
        dadada
    ]]

    反復[[
        dadada
    ]]如果 a == 1;
```

发信站：水木社区 (Tue Oct 23 14:37:38 2012), 站内

趁有假期，用js實現了一套js的詞法分析器、語法分析器、編譯器、解譯器，以及一個很簡單的ide，自己實現了函數和閉包。另外寫了中文化的詞法分析器語法分析器，實現了一個我比較滿意的中文編程語言。

<http://code.google.com/p/chn-js/>

VBA 解释器 (2016)

VBA 代码:

```
1Attribute VB_Name = "Module1"
2Sub Main()
3    OptionalTest a:=2, c:=3
4    OptionalTest c:=3, a:=2
5'    OptionalTest 2
6'
7    ParamArrayTest 1, 2, 3, 4, 5
8
9    Dim a
10   Set a = New Class1
11
12   a.OptionalTest a:=2, c:=3
13   a.OptionalTest c:=3, a:=2
14End Sub
15
16Sub ByRefTest(ByRef a As Integer)
17   a = a + 1
18End Sub
19
20Sub OptionalTest(a As Integer, Optional b As Integer = 3, Optional c As Integer = 4)
21   Debug.Print a + b + c
22End Sub
23
24Sub ParamArrayTest(a As Integer, ParamArray b())
25   For Each i In b
26       a = a + i
27   Next
28   Debug.Print a
29End Sub
```

解释器运行:

```
new Interpreter().executeVbProject("vba\CallTest\CallTest.vbp");
```

解释器转换及运行结果:

```
LIB CallTest
MODULE Module1
PUBLIC ByRefTest (a Integer) As null
0:    Assign (EA a Integer) = (EA a Integer).Apply + (VbValue Integer 1)
End ByRefTest

PUBLIC ParamArrayTest (a Integer, b Array[array of Variant]) As null
0:    For Each Iterator At i Variant
1:        ForEach.Next
2:        Assign (EA a Integer) = (EA a Integer).Apply + (EA i Variant).Apply
3:        IF NOT ForEach.hasNext GOTO 5
4:        GOTO 1
5:        (EA PUBLIC Const DEBUG Object Debug = (VbValue Object Debug org.siphon.visualbasic.runtime.JavaModuleInstance@425
End ParamArrayTest

PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null
0:    (EA PUBLIC Const DEBUG Object Debug = (VbValue Object Debug org.siphon.visualbasic.runtime.JavaModuleInstance@425
End OptionalTest

PUBLIC Main () As null
0:    (EA PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null [org.siphon.visualbasic.runtime.statements.Name:
1:    (EA PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null [org.siphon.visualbasic.runtime.statements.Name:
2:    (EA PUBLIC ParamArrayTest (a Integer, b Array[array of Variant]) As null [(VbValue Integer 1), (VbValue Integer 2
3:    Dim a Variant
4:    Assign (EA a Variant) = New CallTest.Class1
5:    (EA a Variant .OPTIONALTEST [org.siphon.visualbasic.runtime.statements.NamedArgumentStatement@11ce2e22, org.sipho:
6:    (EA a Variant .OPTIONALTEST [org.siphon.visualbasic.runtime.statements.NamedArgumentStatement@13cda7c9, org.sipho:
End Main

End Module Module1

MODULE Class1
PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null
```

```

PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null
0:      (EA PUBLIC Const DEBUG Object Debug = (VbValue Object Debug org.siphon.visualbasic.runtime.JavaModuleInstance@425
End OptionalTest

PUBLIC Main () As null
0:      (EA PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null [org.siphon.visualbasic.runtime.statements.Name
1:      (EA PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null [org.siphon.visualbasic.runtime.statements.Name
2:      (EA PUBLIC ParamArrayTest (a Integer, b Array[array of Variant]) As null [(VbValue Integer 1), (VbValue Integer 2
3:      Dim a Variant
4:      Assign (EA a Variant) = New CallTest.Class1
5:      (EA a Variant .OPTIONALTEST [org.siphon.visualbasic.runtime.statements.NamedArgumentStatement@11ce2e22, org.siphc
6:      (EA a Variant .OPTIONALTEST [org.siphon.visualbasic.runtime.statements.NamedArgumentStatement@13cda7c9, org.siphc
End Main

End Module Module1

MODULE Class1
PUBLIC OptionalTest (a Integer, b Integer, c Integer) As null
0:      (EA PUBLIC Const DEBUG Object Debug = (VbValue Object Debug org.siphon.visualbasic.runtime.JavaModuleInstance@425
End OptionalTest

Class WithEvents New Object CallTest.null
End Module Class1
|
8
8
15
8
8
8

```

我已经实现了包括数据类型、流程控制、函数声明调用、传值传址、错误处理、堆栈信息、默认属性、默认方法、集合、数组、常量、枚举、模块、类模块、Implement、调试器等等在内 VBA 语言所有语法特征, 并支持与 Java 的互操作。唯 VBA 的函数库仍未完全实现。

设计该解释器目的是为了实现在一种更有实用价值的规则引擎语言。我扩充了 VBA 语言, 增加了 WHEN 语法设计, 示例如下:


```

2 Sub Main()
3     Dim r
4     Stop
5     r = SqrRoot(1.0, 2)
6     Debug.Print r
7 End Sub
8
9 Function IsGoodEnough(guess As Double, x As Double) As Boolean
10     IsGoodEnough = Abs(x - guess * guess) <= 0.00001
11 End Function
12
13 Function NewGuess(guess As Double, x As Double) As Double
14     NewGuess = (guess + x / guess) / 2.0
15 End Function
16
17 Rule SqrRoot(guess As Double, x As Double) As Double
18     SqrRoot = SqrRoot(NewGuess(guess, x), x)
19 End Rule
20
21 Rule SqrRoot When IsGoodEnough(guess, x)
22     SqrRoot = guess
23 End Rule
~

```

使用 WHEN 子句的规则可以轻松表达公积金、年假等复杂的业务逻辑，较 DROOLS 等规则引擎更为亲民。

该语法借鉴自 Erlang 的 Guard 语法。

VBA 语言的库函数较为繁琐，业余时间有限，该项目现处于搁置状态。