

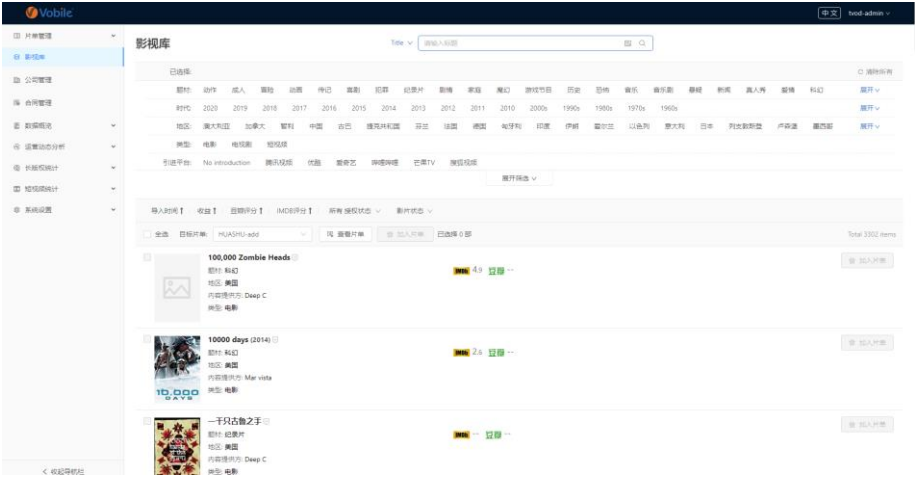
产品项目成果汇编

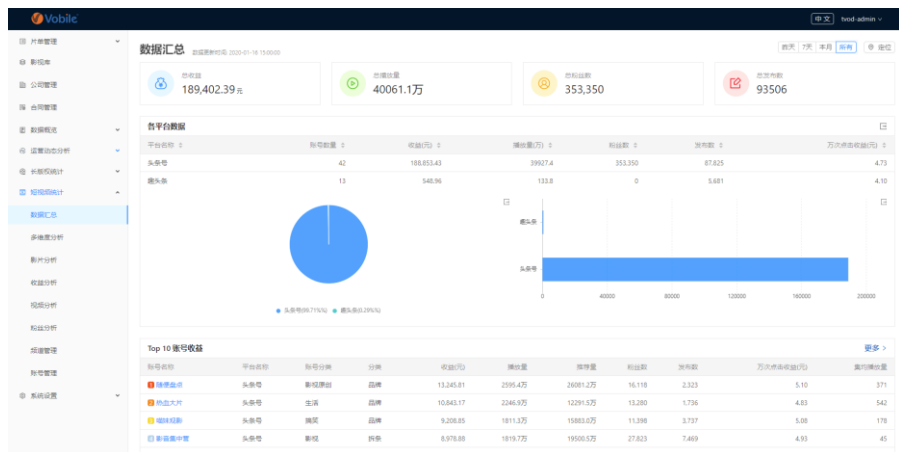
本人先后从事过银行 MIS、教育培训、GPS 车联网、医疗 ERP、小游戏、民航
管理系统、APP、电子教育、版权保护、爬虫、媒体等等若干行业，业余时间
还涉足游戏、编译器等领域。不论该领域有前人躬耕数年还是一片处女地，都
可以快速成长为专家，并在其中找到新的突破点，最终有所发明，用更新更好
的产品赢得市场认可。

以下是若干相关产品和项目的介绍。

版权分发系统暨短视频运营系统（2019）

本系统为实现 TVOD 业务开发，覆盖了意向片单、片单入库、片单出库、作品
库检索、收益分析等业务场景。后因业务转向短视频，又迅速开发出短视频
运营相关产品功能。在以产品为中心的思想推动下，研发和运营高度默契，短
视频业务很快实现盈利。





系统后端采用 d2js 技术，前端采用 react 技术。IMDB 豆瓣等全球影视库爬取采用 python3 实现。

由于大量运用了 Postgres JSONB 类型，系统数据库结构简单，表数目少表达能力强。本小组以一个产品经理、一个前端工程师、一个后端工程师、半个爬虫工程师的规模完成了产品研发并持续支持业务迭代。

本人从零开始在广州异地招聘组建小组，担任管理、架构、技术指导等各项工作。

广东广播电视台广播信号安全传输（2018-2019）

广播电台有线、无线传输环节较多，一旦播出异常容易发生政治事故，因此各广播电台都非常重视播出安全。

公司很早就介入了该市场，但一直没能给出让客户满意的产品，眼看客户就要流失。

2018 年底本人从零开始到广州进行需求调研，制定产品及架构方案，最终设计完成一款实时获取播出信号，实时比对，传输链路可定制的高质量播出安全产品。用户在浏览器可直接观看每个节点的实时声波波形图，随时可收听效果，播出异常在声波图实时渲染并产生报警。项目于 2019 年完工并实施后赢得客

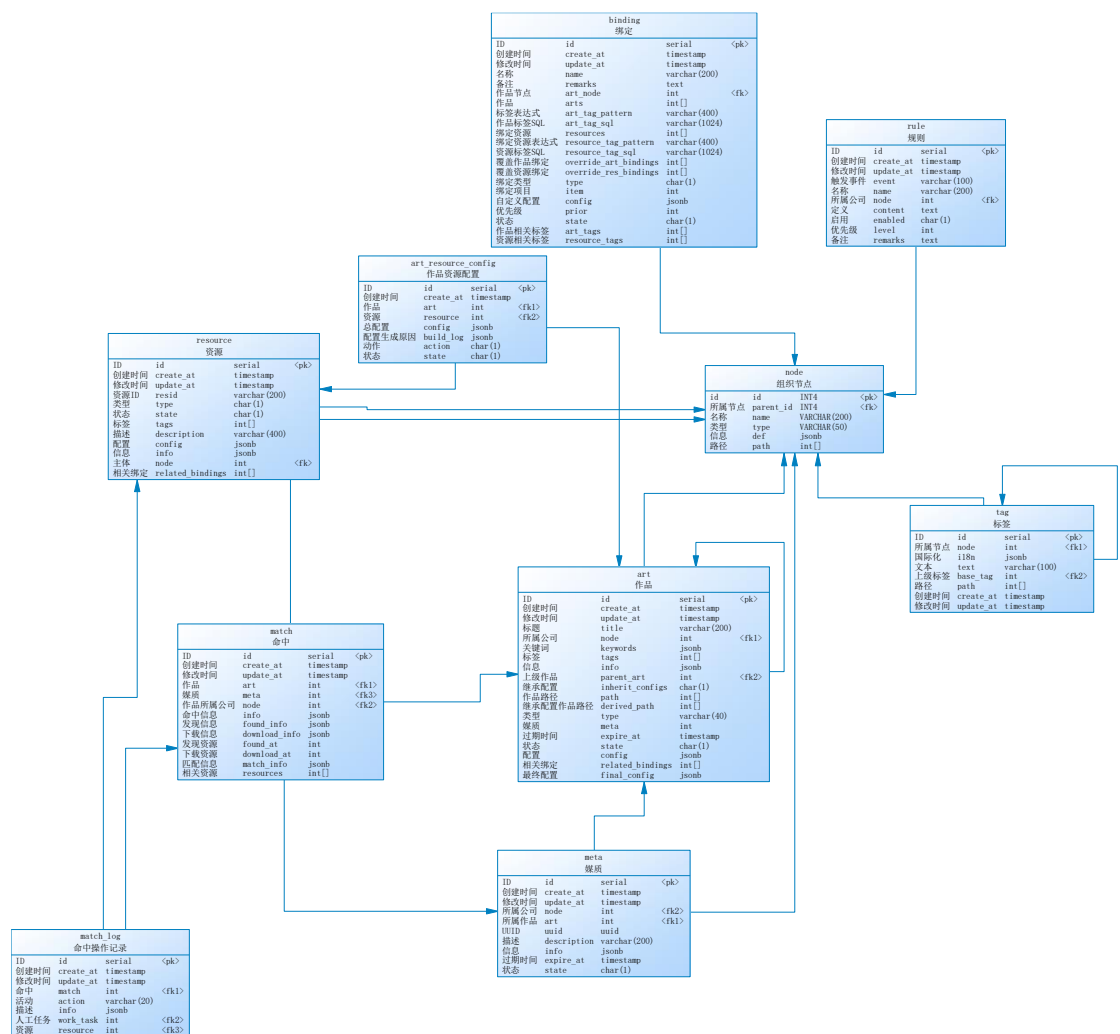
户好评，客户评价称为广电行业数一数二的好产品，公司用产品重新赢得了客户青睐。

版权监测方案（2017-2018）

版权保护为阜博通主营业务，但该产品一直缝缝补补了 10 余年，开始的设计不当造成后面运营、维护、迭代成本高企，且实际上影响了潜在市场增长。

本方案从 2017 年 8 月开始，到 2018 年初形成了初步可用的 demo，后因各种因素未能继续产品化并取代原系统，仅 YouTube 网站爬虫部分因性能突出一直在生产环境工作。

图为系统核心单元架构。



本方案系统导入了作品概念，矫正了以往业务围绕介质（meta）而展开的错误设计。作品既是一个 CMS 概念，在本业务场景中又是介质、关键词、邮件模板等业务概念的束。

系统设计了标签、规则、绑定等业务概念，极大的扩展了系统的灵活性，不但能顺应多变的需求变更，也让系统具备了超出版权保护行业推广到其它领域的的能力。

本系统具备完善的回溯功能，由于规则设计得当，每个环节均可追溯，由此大大提升了系统的可维护性。

因欲借助 asyncio 快速完成 demo，系统业务部分主要采用 python3 语言实现。管理部分采用 d2js 实现。

在本系统中本人担任发起、架构、管理、前后端编码等各项工作。

YouTube 爬虫（2018-2020）

```
vt@search-bankend:log$ grep -c 'fetch succ' vt-2017.log.2020-02-0*  
vt-2017.log.2020-02-01:2093347  
vt-2017.log.2020-02-02:2083647  
vt-2017.log.2020-02-03:2162832
```

YouTube 是全球最大的 UGC 视频网站，日产量很高，在本行业具有代表性。

为了验证上述方案的有效性，并验证关键词优化的价值，基于上述方案开发了 YouTube 爬虫系统及关键词/作品优化模块。

YouTube 爬虫将代理池理解为搜索机会，将竞争问题转化为分配问题，充分挖掘代理服务器资源，使单日爬取次数从 $20 \pm$ 万次提升到 $200 \pm$ 万次。

另外，根据实际运行数据的仔细观察和分析，本人编制了关键词/作品优化算法。该算法降低了无效关键词的频次，使每日新发现 URL 量从 100 万左右下

降到 30 万次左右，发现量下降随之降低了后续下载比对环节的成本。同时算法增加了高产作品的出场频次，使系统能更及时的发现新的侵权内容。

本爬虫采用 asyncio 技术，单核执行，采用 leveldb 去重，LRU 内存用量约 2G。由于 python 不适合 CPU 密集型场景，关键词优化模块采用 Java 重写，与爬虫通过 RabbitMQ 交互。

在爬虫开发中本人担任指导、编码、维护等各项工作。

hotview(2019)

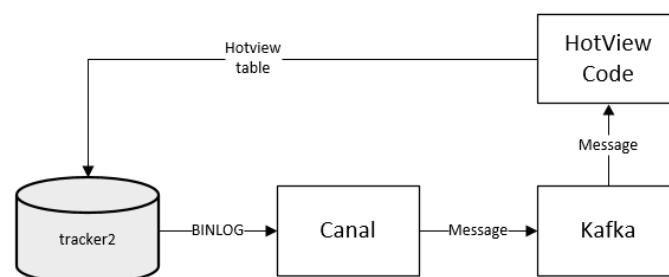
在 MySQL 优化过程中，本人参考 PostgreSQL 的 Continuous View 设计发明了 hotview。

一个典型的 hotview 采用 python 编写，声明式风格，可监听相关若干表的变动，同步 hotview 持有的数据表。

```
        AND b.notice_status = 'sent'
        AND b.company_id = a.company_id
        AND a.matchedVideo_id = mv.id
        AND a.company_id = mv.company_id
        AND mv.takeoff_time = 0
        AND mv.trackingMeta_id = me.meta_id
        AND mv.company_id = me.company_id
    }s
    ...
}

def __init__(self, view_db: Engine, logger: logging.Logger, cache_db: Engine = None):
    logger.info('start Hot_takedownNoticeItemForResend')
    meta = MetaData()
    mapping = [
        TableMapping('takedownNoticeItem', meta,
            ColumnMapping('id', cond_alias="a.id", view_alias="parentId", is_fixed=True, primary_key=True),
            ColumnMapping('notice_send_count', view_alias="noticeSendCount", is_fixed=False),
            ColumnMapping('updated_at', view_alias="updated_at", is_fixed=False),
            trace_insert=True,
            predicate=lambda row: int(row['parent_id']) == 0 and row['is_stop_resending'] in ['2', 'false', 2]),
        TableMapping('takedownNotice', meta,
            ColumnMapping('id', cond_alias="b.id", view_alias="takedown notice_id", is_fixed=True, primary_key=True),
            trace_insert=True, # 有时先插入 takedownNoticeItem 后插入 takedownNotice
            predicate=lambda row: row['notice_status'] in ['2', 'sent'] and int(row['id']) > 0
        ),
        TableMapping('matchedVideo', meta,
            ColumnMapping('id', view_alias="tracking_relation_id", cond_alias="mv.id", is_fixed=True, primary_key=True),
            ColumnMapping('last_refresh_at', view_alias="lastRefreshAt"),
            ColumnMapping('count_send_notice', view_alias="countSendNotice"),
            ColumnMapping('hide_flag', view_alias="hide_flag"),
            ColumnMapping('last_send_notice_date', view_alias="lastSendNoticeDate", is_fixed=False),
            ColumnMapping('is_poster_whitelist_filtered', view_alias="is_poster_whitelist_filtered"),
            predicate=lambda row: row['takeoff_time'] == '0000-00-00 00:00:00' or row['takeoff_time'] == 0
        ),
        TableMapping('metaExtraInfo', meta,
            ColumnMapping('meta_id', view_alias="metaId", cond_alias="me.meta_id", is_fixed=True, primary_key=True),
            ColumnMapping('trackingSetting_id', view_alias="trackingSetting_id")
        )
    ]
```

hotview 示例



原理图

系统采用阿里 canal 框架同步 mysql binlog, canal 可将 binlog 变动投递到 kafka 消息队列。hotview 从 kafka 收听变动, 根据 hotview 定义更新 hotview 数据表。

hotview 可适用于许多被认为需要 Lambda 大数据架构的场景, 非常适合 SQLer 提升统计报表、dashboard 等功能的性能。

python-executor(2019)

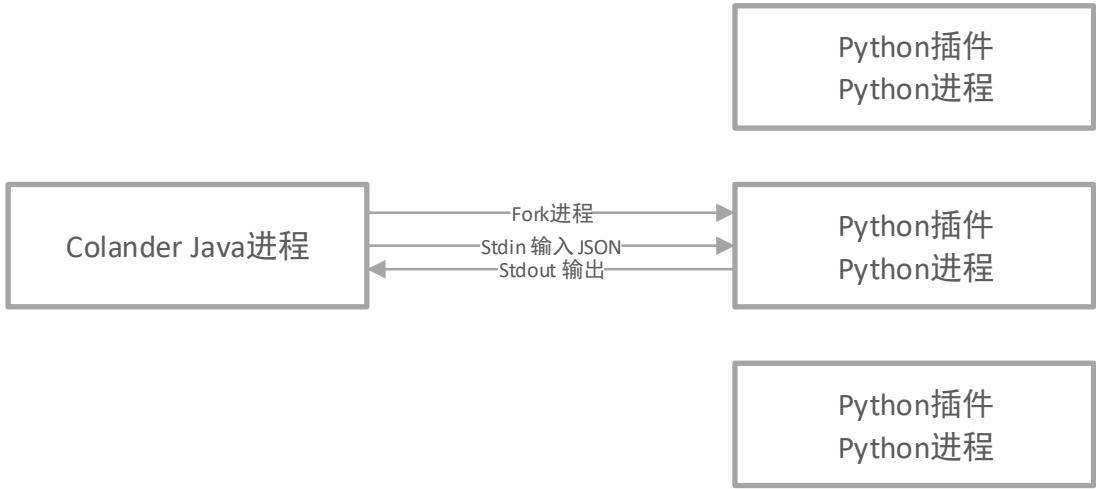
公司爬虫系统 Colander 为了实现系统开放灵活, 可支持 Python 脚本插件, Java 插件等各类插件, 其中大部分为 python 插件。colander 通过 fork 进程方式调用 python 插件, 每次调用都创建一个进程, 成本极高。

本服务将 fork 进程方式调用 python 脚本改造为以多线程方式调用 python 脚本。由于脚本都是 python2.7 编写的, 本系统也采用 python2.7 语言。

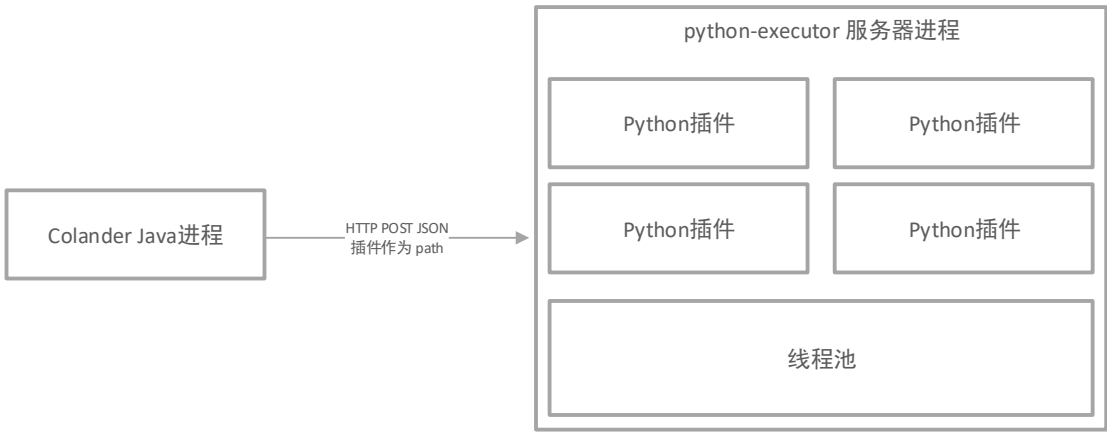
对调用者来说服务表现为一个 web 服务器, 通过 http post 脚本路径 可调用脚本, 通过 post body 注入 sys stdin, http 响应即 sys stdout。

通过改造 colander 单台服务器的效率提升超过一倍, 由此缩减超过一半相关设

备成本。

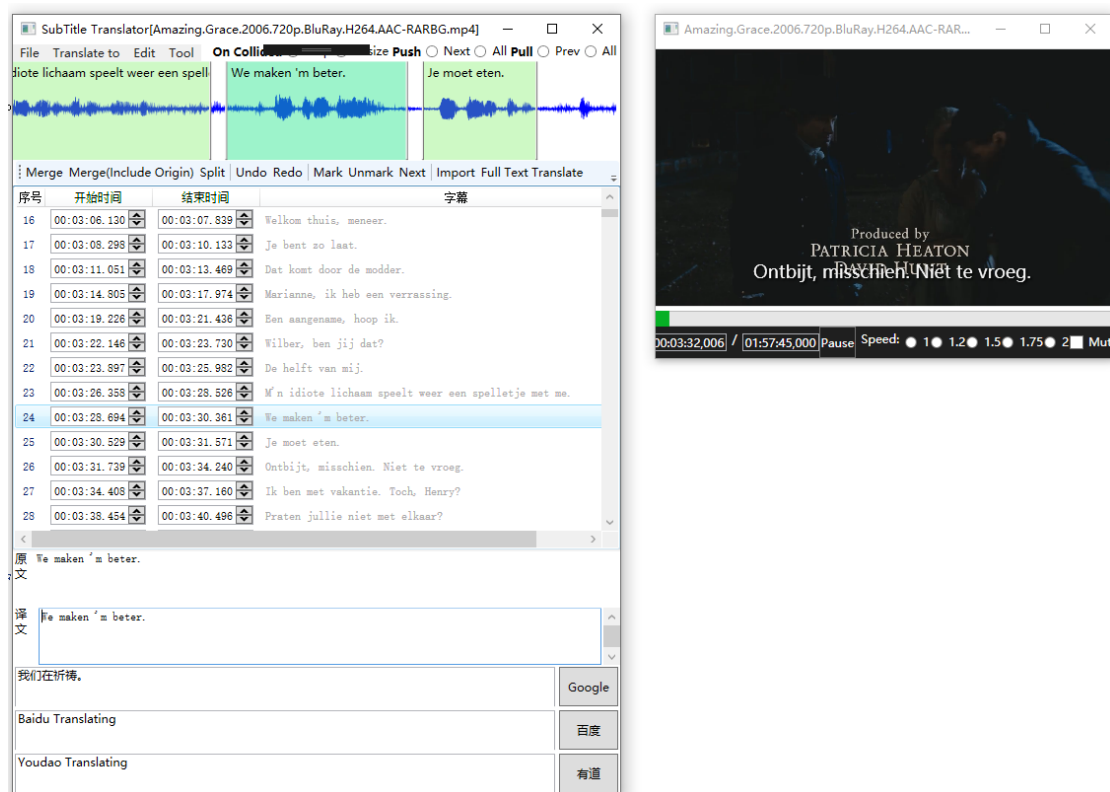


原模式



新模式

字幕翻译工具（2019）



Windows 客户端工具。采用 c# wpf 技术，在 visual studio 2017 开发。

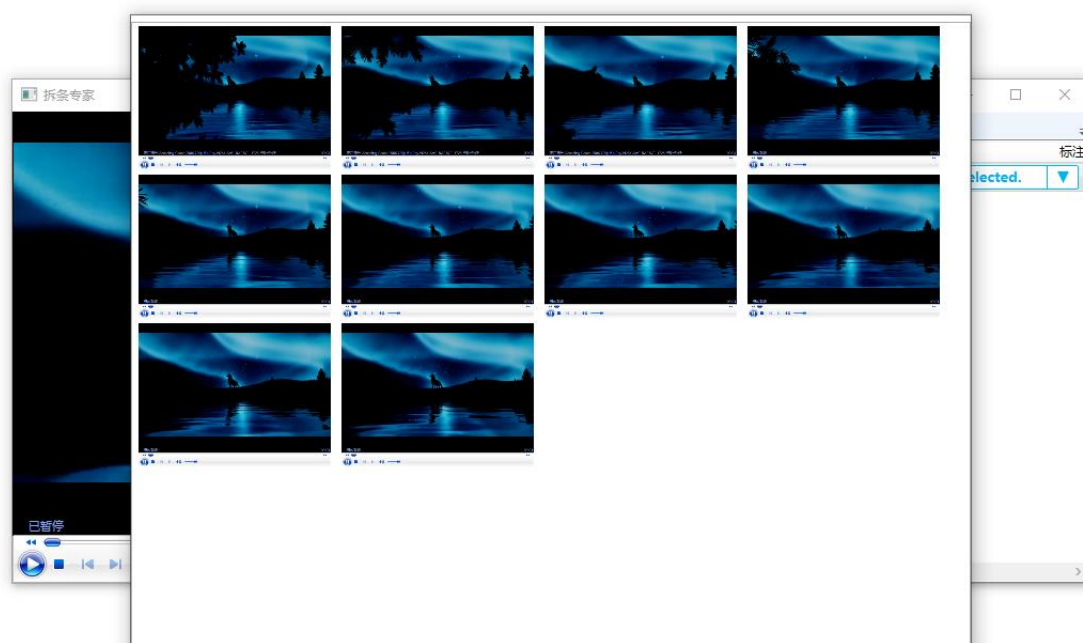
本工具用于字幕翻译，提升翻译效率。

本工具主要功能有：

1. 支持多种翻译源语言和目标语言
2. 支持 Google、百度、有道等线上翻译接口辅助翻译
3. 本工具将一个视频的字幕翻译视为一个工程，工程可支持按不同选型输出 SRT 字幕
 1. 支持去除辅助内容
 2. 支持去除说话人
 3. 支持合并折行
 4. 支持双语字幕输出

4. 针对特定语言合并拆分字幕而原始字幕不变
5. 在可视化声波图上叠加可视化时间轴，拖曳时间轴缩放调整字幕时间
6. 支持字幕整体移动
7. 实时字幕预览
8. 采用 OpenNLP 对英语自动断句

拆条工具（2019）



使用 C# WPF 编写的人工拆条效率提升工具。按下空格回溯数帧，选择帧作为拆条点，操作简单迅捷，大大提升了拆条效率。

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。

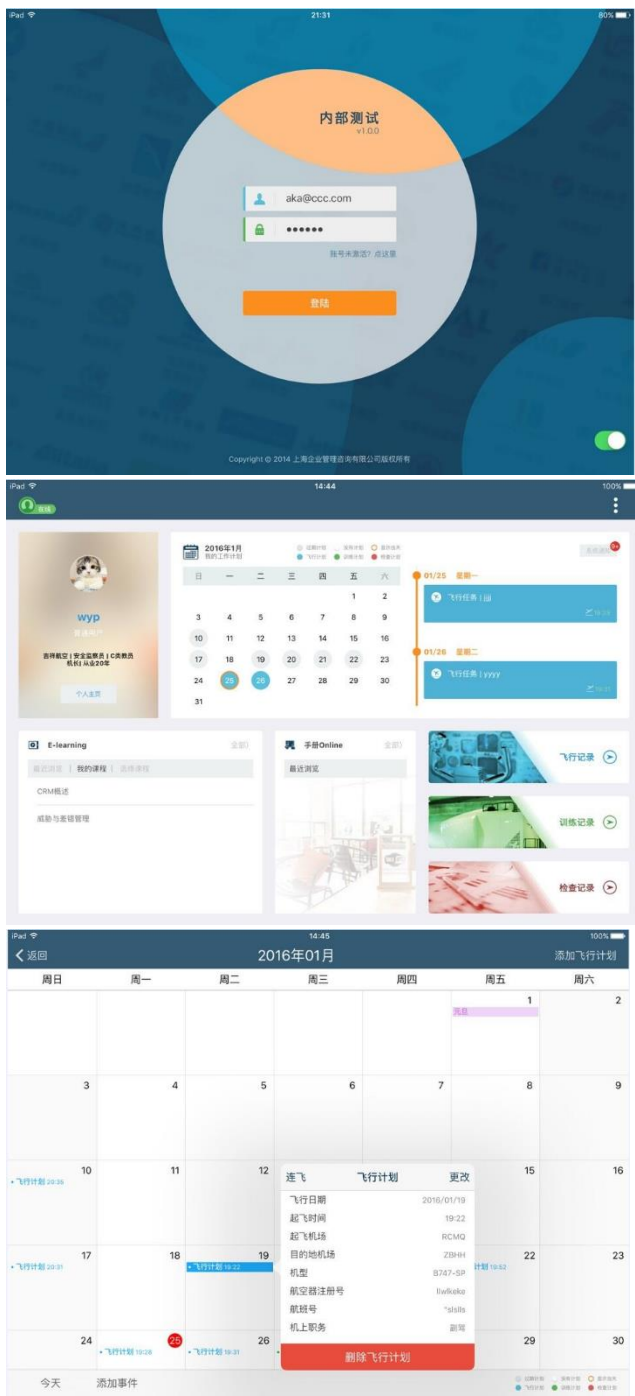
新闻参考：功夫财经获合一集团 1500 万人民币投资，视频网站入局财经内容市场_36 氪
<https://36kr.com/p/5047038.html>

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



成都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）

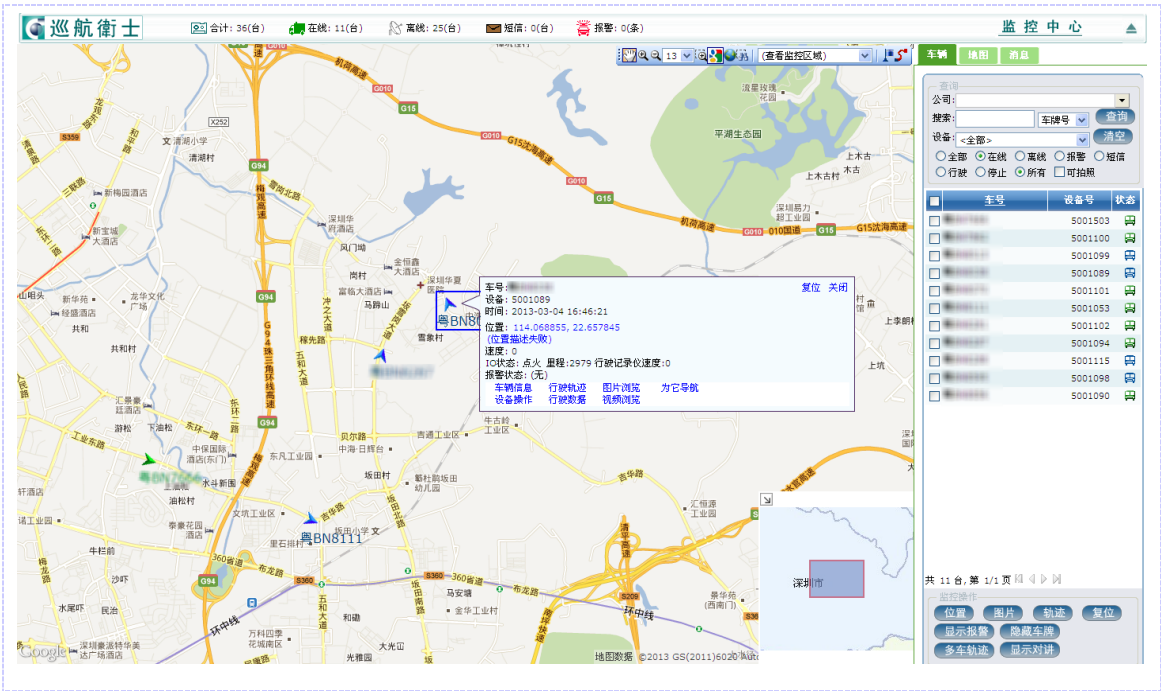


以上为认知科学健脑游戏。使用 HTML5 cocos2d-js 开发，适合平板电脑、大触摸屏。这些游戏看起来简单，每个游戏设计思路都不同，在我加入公司之前，一款游戏需要制作一周。有 Android 合作公司想改为 native 方式，花了一个月也未成功。

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 ActionScript** 生成 **Cocos2D JS** 动画代码的工具。这两项工具大大加速了从设计到程序的转换，使工期大幅缩短，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（2005-2012）

BS Webapp（2005）

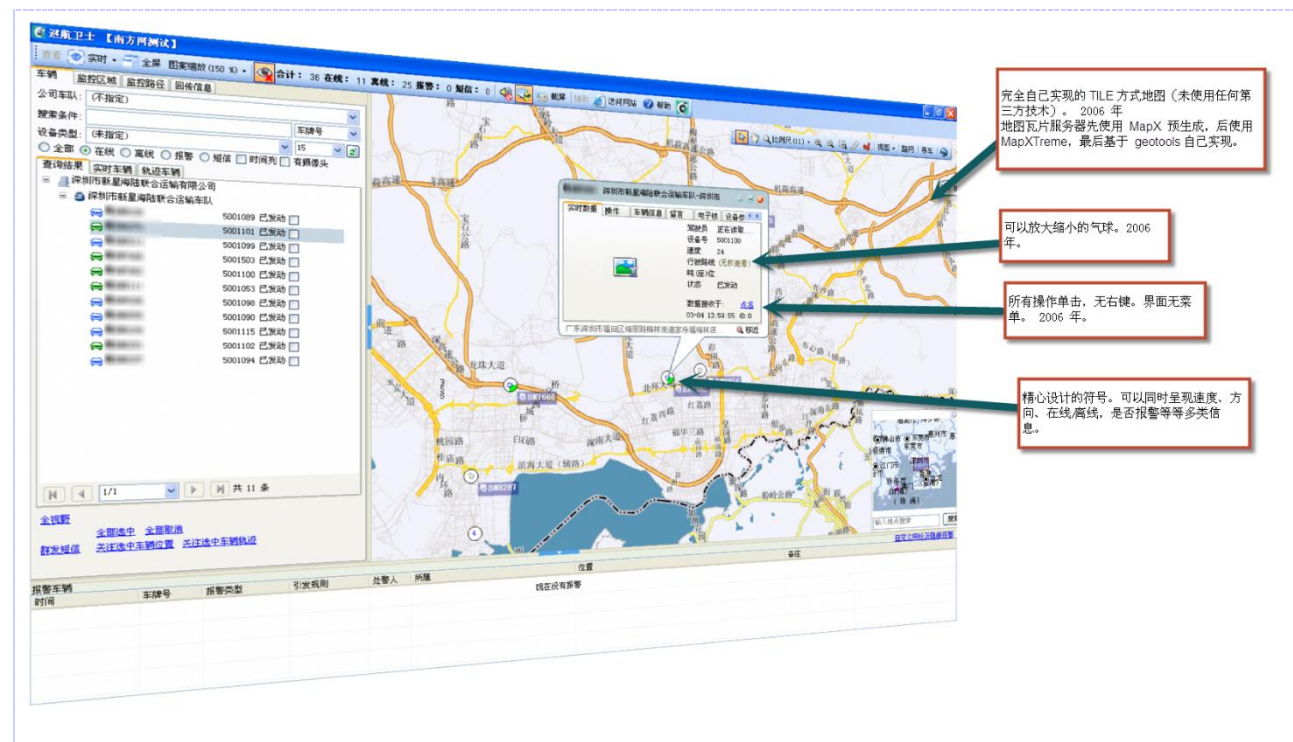


BS 所用地图为本人独立开发的 Web 地图引擎，在业界还在为 MapX 好还是 MapXtreme 好而纠结的时候（2006），本人即已独立研发出类似 Google Map 可以实用的 Web 地图技术。后来又通过包装加入了 Google 地图、图吧地图、四维地图等其它第三方 WEB 地图。

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。



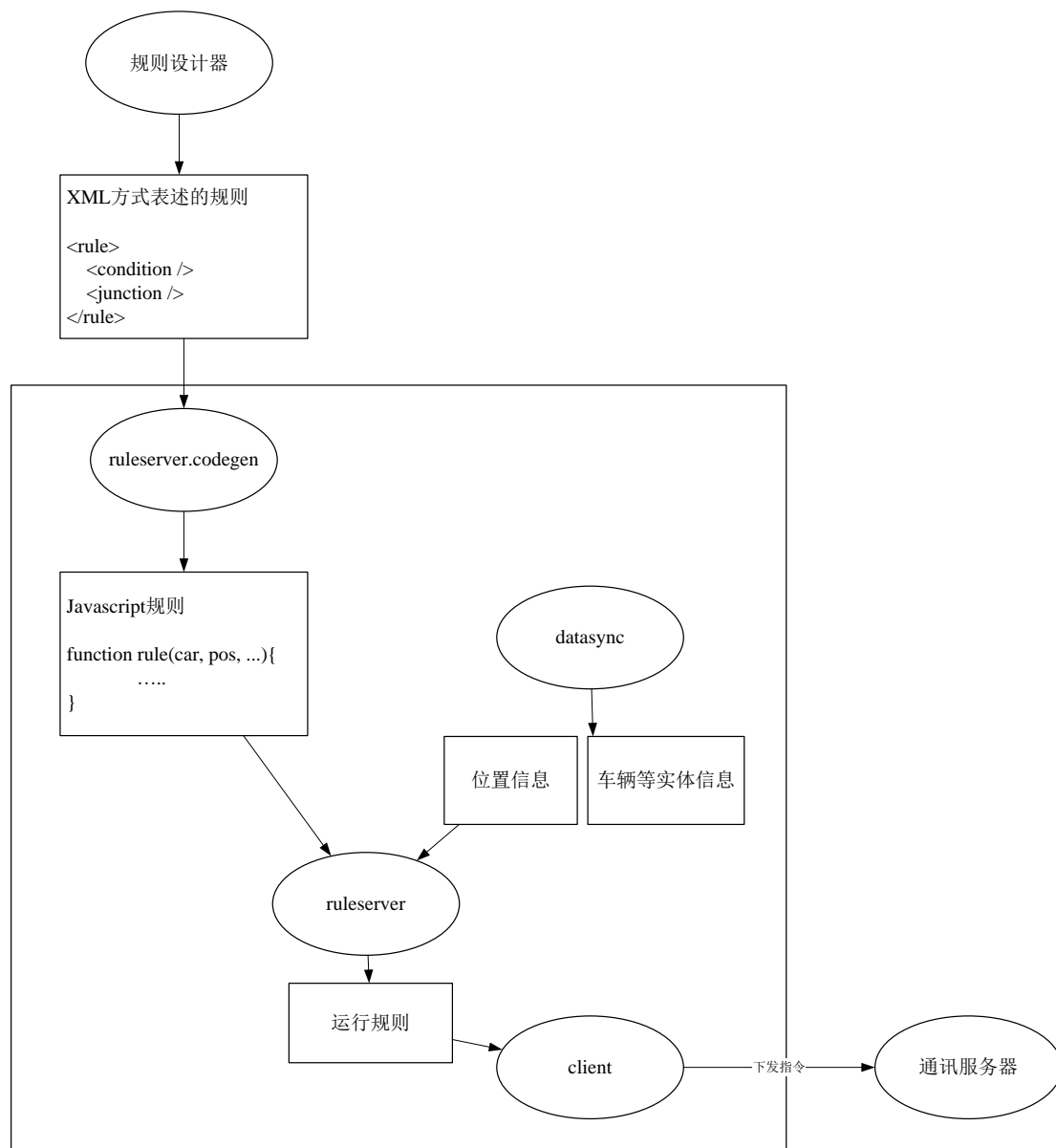
地图控件、树控件、气球控件都是自己编写。其中地图需要快速移动，需要达到较高的 FPS 指标，对图像动画编程有较高的要求；监控系统使用 MVC 思想设计，车辆数据每 5 秒轮询一次以更新数据。在更新画面时用户往往还在操作地图，需要对 .NET 并发编程有较好理解，不然极易崩溃；整个界面设计不采用当时流行的菜单+MDI+对话框，全系统几乎没有子窗口，完全没有模式对话框，界面和操作都集中在一个监控窗口完成，简洁干净，操作集中方便，尤其单车操作分解清晰，使用气球的表达方式，生动活泼，本套界面设计除了对业务需要有较深的理解，更重要的是对 UI 表达能力的考验。

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。

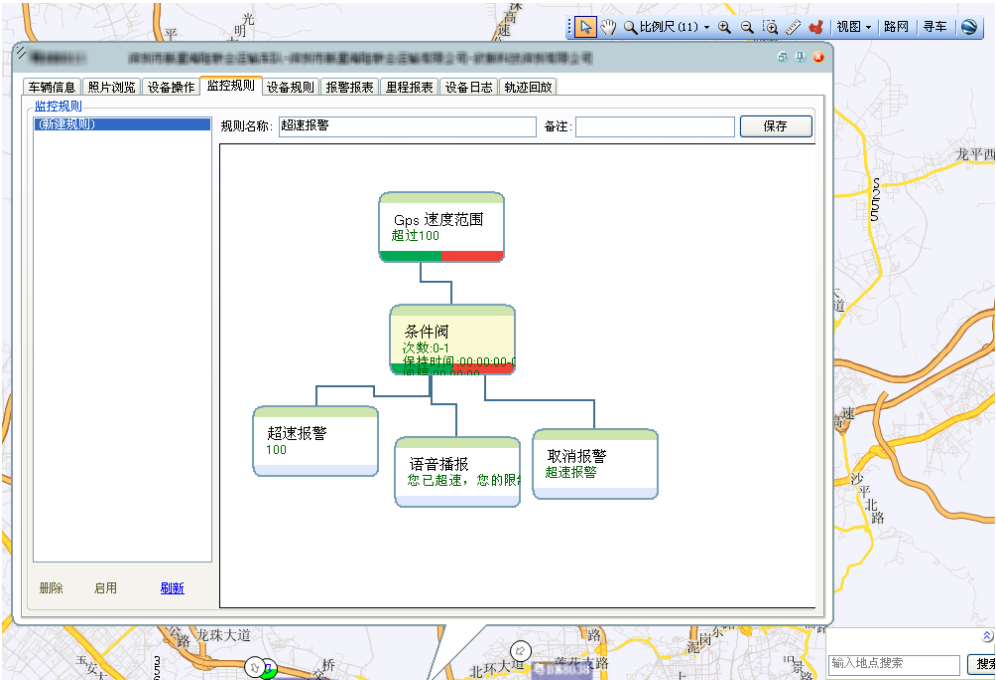


规则服务器主要逻辑过程

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。采用栈隔离实现并发。每条规则生成一个 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。

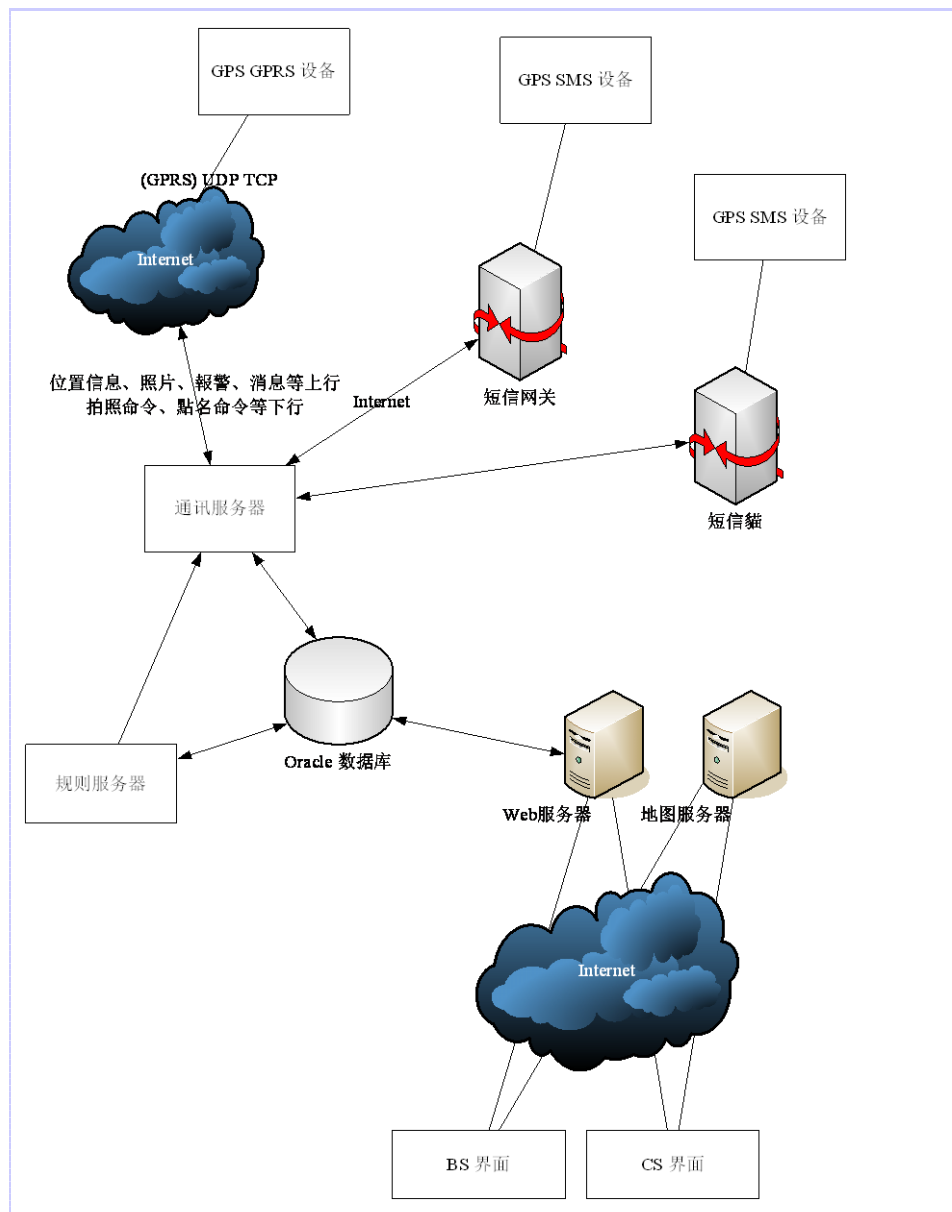
本人独立设计了如下**规则设计器**：



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 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。

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。

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(canvas1){
                AutoHeight = true,
                Solid = true,
                //PaddingX = 22,
                BackgroundColor = ColorUtil.FromArgb(0xffdcdf5),
                Rectangle = new Rectangle(20,45,300, 73),
                Elements = new Div[]{
                    new Div(canvas1){
                        Opacity = 0,
                        Text = "乘客位置:", Location = new Point(20,20), AutoSizeWithText = true
                    },
                    new Div(canvas1){
                        Opacity = 0,
                        Id = "txHailLocation", Text = "-", Location = new Point(20,40), WordWrap = true, Size
                    },
                    new Div(canvas1){
                        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);
                    }
                })
            }
        }
    }
}

```

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#7
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     log4rl: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>>,

```

其它个人作品

d2js 及 molecule (2013-2015)

```
d2js.fetch = function(params){
  this.checkPrivilege("uac.sys_role.query");

  params.node = params.node || this.session.user.node.id;

  sql{.
    select t.id,t.node,t.name,t.code,t.node_scope,t.role_level,t.state,t.remarks,nd.name node_name from
    uac.node nd
    where t.node = nd.id and (nd.id = :node or nd.path @> array[:node])

    sql{.?(params.name)
      and strpos(t.name, :name) > 0
    .}
    sql{.?(params.state)
      and t.state = :state
    .}
    and t.role_level >= :role_level
  .}

  params.role_level = this.session.user.role.role_level;

  sql = this.orderBy(sql, params._sorts, {name: 'asc'});

  return this.query(sql, params, params._page);
};

d2js.create = function(rcd){
  this.checkPrivilege("uac.sys_role.add");
  $V(this,rcd, {id : [],
    node : [V.notNull],
    name : [V.longest(100),V.uniqueInNode('uac.sys_role')],
    code : [V.longest(100),V.uniqueInNode('uac.sys_role')],
    role_level : [V.longest(5)],
    state : [V.longest(1),V.inside(["N","P"])]

  });

  this.doTransaction(function(){
    rcd.id = this.nextId("uac.SEQ_ENTITY");
    this.actas('uac.sys_role.add', 'uac.sys_role', rcd.id, rcd);
    this.insertRow('uac.sys_role', rcd, ["id","node","name","code","node_scope","role_level","state","re
    this.updateDispatchedFunctions(rcd.dispatched_functions, rcd.id);
  });
};
```

Book

Name

Kind

武侠

Author

康德

Q Search

+ 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

✕ Remove

<

1

2

>

详见 <https://github.com/inshua/d2js>。

d2js 将 SQL 与 JS 语言穿插融合在一起，为构建复杂的面向数据库的应用带来了极大的便利，在各类企业应用中都具有很好的通用性。该框架学习轻松，通常一天即可上手。

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。

```
request_session_test.jssp
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.jssp');
11    }
12 %]
13
14 <p>
15
16 [% if(session.name){ %]
17     你好, [%= session.name %], 我将写信给你 [%= session.email %]
18 [% } else { %]
19     请输入您的个人信息
20 [% } %]
21
22 </p>
23
24 <form action="request_session_test.jssp" method="post">
25
26 姓名: <input name="name">
27 Email: <input name="email">
28
29 <button type="submit">提交</button>
30 </form>
31
32 </body>
33
34 </html>
```




测试 REQUEST SESSION

localhost:8080/testweb/jssp_test/request_session_test.jssp

请输入您的个人信息

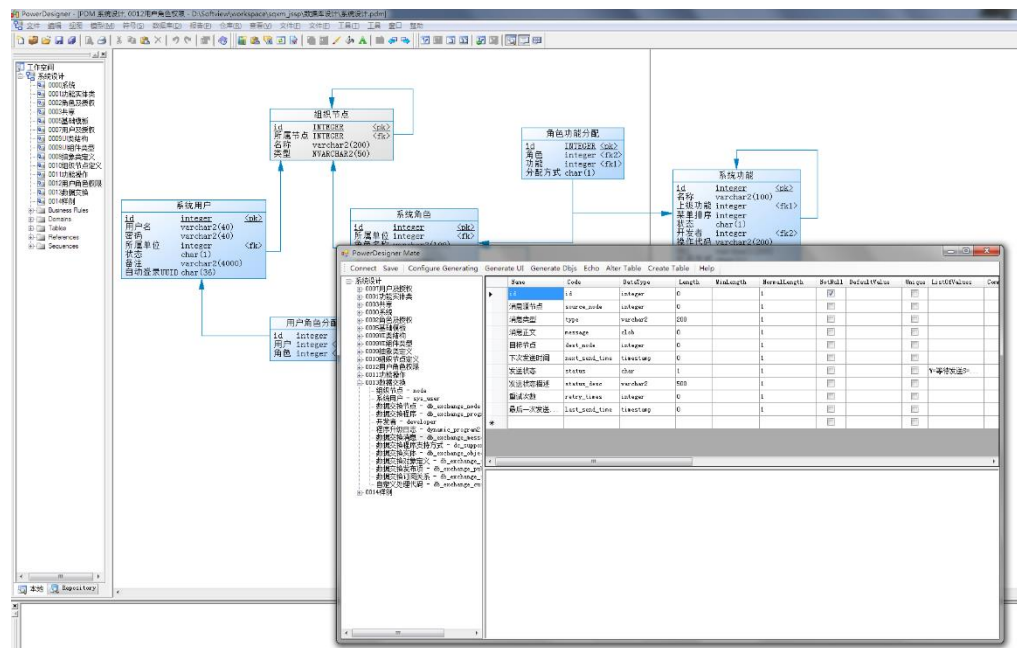
姓名: Email:

用户权限框架（2014）

<https://github.com/inshua/uac>

基于长期项目实践，积累了一套灵活的用户权限框架。该框架采用 d2js 实现，前端呈现目前采用 molecule + bootstrap，也可以采用其它前端框架。uac 模块将权限正交分解为功能集合、实体视野两个域，在岗位进行结合，对各类企事业单位均有良好的适配性。该框架可便捷的与 Activiti 流程引擎、进销存、商城等其它子系统融合，形成适合行业的应用。

基于 PowerDesigner 和 JSSP 的代码生成器（2013）

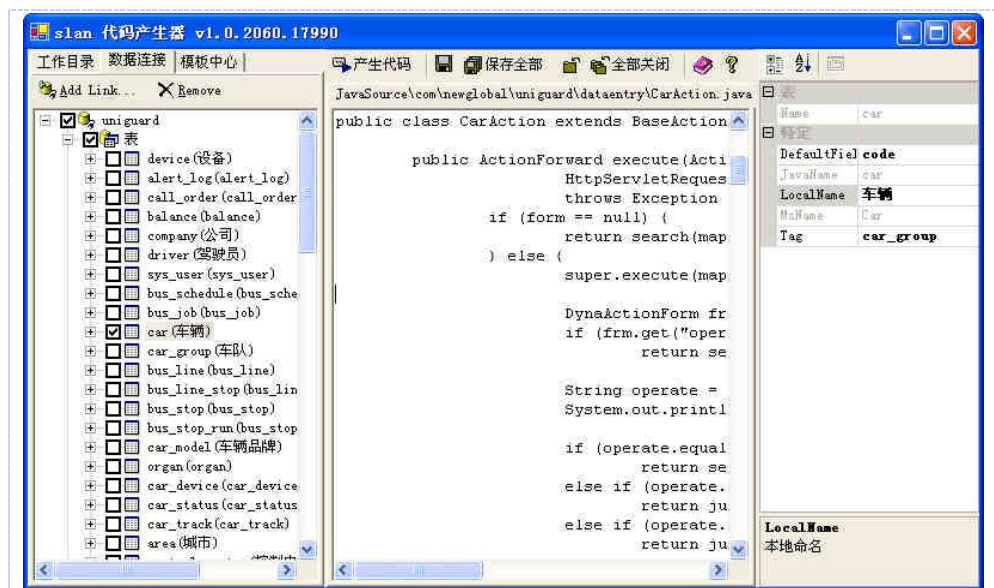


通过网格可以快速编辑字段、表结构、主键、外键、约束等。

连接 JSSP 可以生成代码。一个 JSSP 文件相当于一个模板文件：

```
dbjs_template.jsp 11
<!-- 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.searchUIIndex && itm != tb.nodeColumn;})
44     .sort(function(itm){return itm.searchUIIndex;}).forEach(function(col){
45     cond(dbTypeToExtFiledType(col.type) == 'string', function(){%}
46     if(params.q.[col.code] != null) sql += ' and instr([col.code], :[% col.code %]) > 0';
47     [% %]}];
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 SV(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.[col.pkColumn.code] = this.nextId([col.pkColumn.sequence || 'SEQ_ENTITY' %]);
68 actas('[% tb.code %].add', '[col.code %]', rcd.[col.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', '[col.code %]', rcd.[col.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', '[col.code %]', rcd.[col.pkColumn.code], rcd);
83 this.deleteRow('[% tb.code %]', rcd);
84 return {type: 'exec', data: rcd};
85 };
86 }
```

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>
```

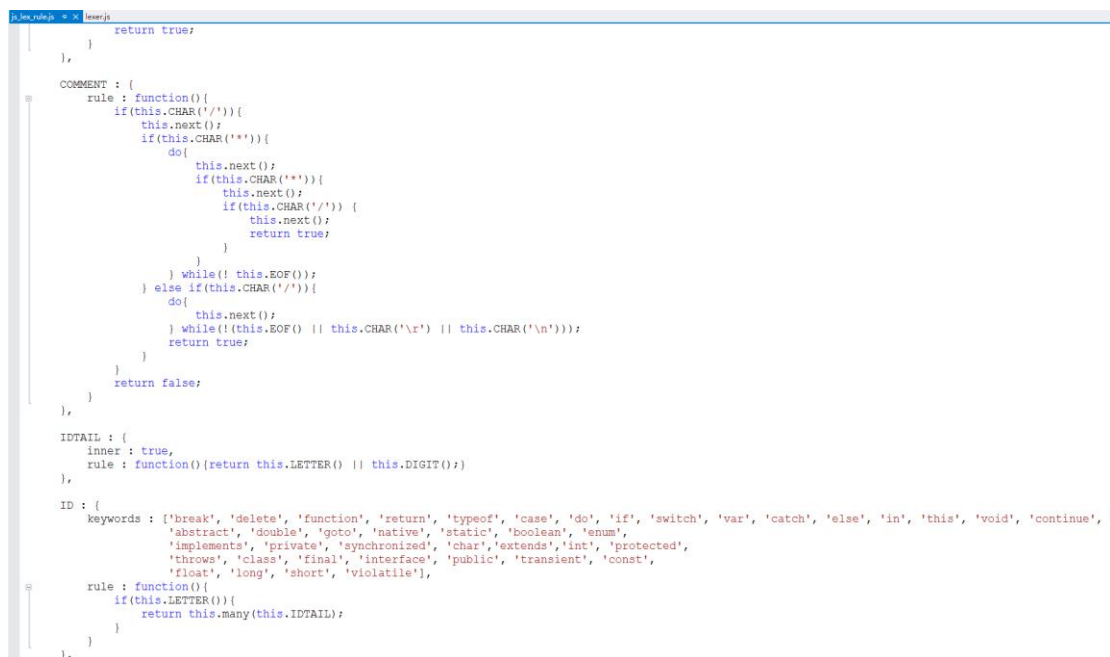
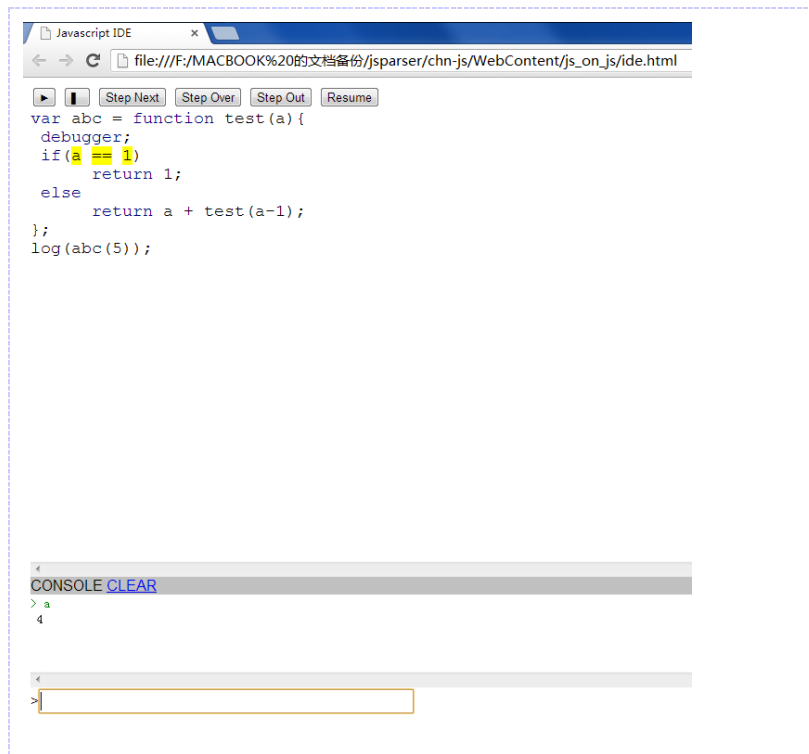
工具使用 CodeDom 将模板翻译为 VB.NET/C# 代码，编译为程序集，执行后生成目标语言代码。目标语言可以是任何编程语言和非编程语言。

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)

<https://github.com/inshua/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也出现在三年后 Apple 的 Swift 的语言中。

```

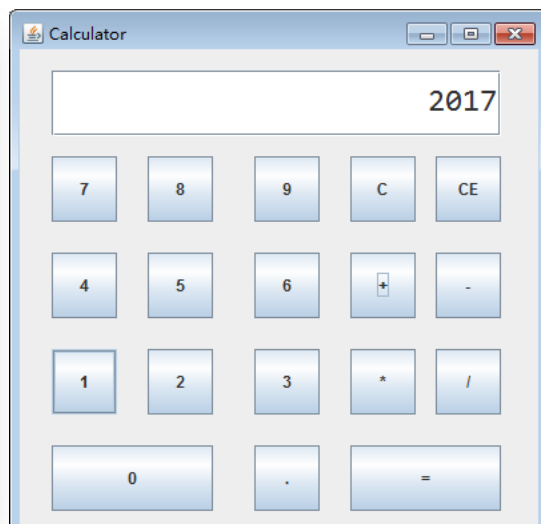
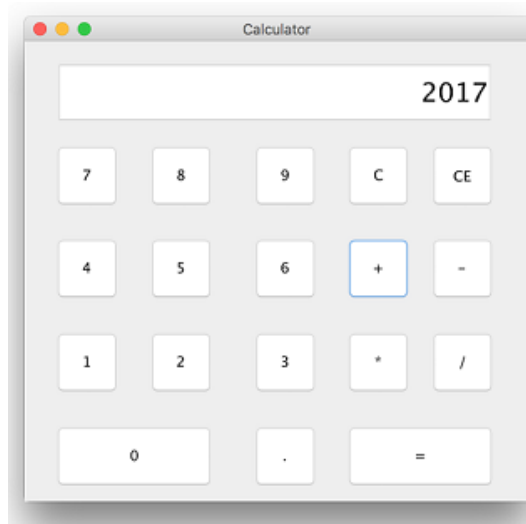
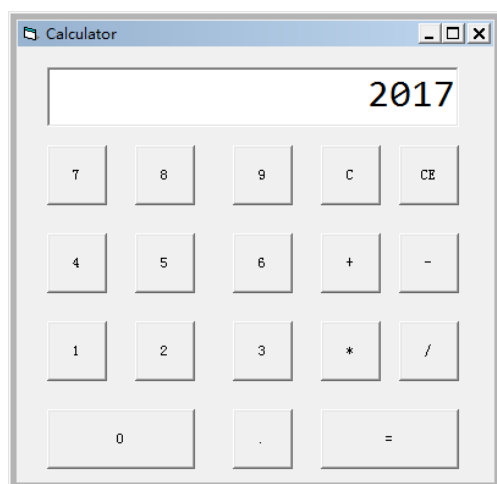
判斷
  如果 a == 1 [[          // 後面必須接語句塊，即：不允許
if(cond) aa; 必須是 if(cond) {aa;}
    dadada
  ]] 否則 [[
    dididi
  ]]

循環
  如果 a == 1 反復[[
    dadada
  ]]

  反復[[
    dadada
  ]]如果 a == 1;

```

VBA 解釋器 (2016)



VBA 代碼：

```

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

```

本项目基本覆盖了 VBA 常见语言特性，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，并支持与 Java 的互操作。唯 VBA 的函数库未完全实现。

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。本项目扩充了 VBA 语言，增加了 Rule 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
24

```

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。