# 第 5 章　自适应信任协商系统设计

基于第 4 章提出的自适应自动信任协商模型，本章主要讨论如何设计和实现一个自适应信任协商系统。

## 5.1　系统总体设计

自适应信任协商系统的主要内容包括证书管理器、策略管理器、协商决策模块、信任度评估模块和一致性校验器等，是对自动信任协商系统的改进。证书管理器主要负责管理加载证书和声明；策略管理器主要负责管理访问控制策略；信任度评估模块主要对用户的可信度做出评价，并提供可验证的功能；一致性校验器接收决策模块的请求，判断凭证集合是否满足访问控制策略；证书链处理模块验证凭证集合是否构成信任凭证链；协商决策模块调用其他各个模块以获得协商过程中需要的中间结果，负责整个信任协商过程的协调。该系统的模块图如图 5.1 所示。

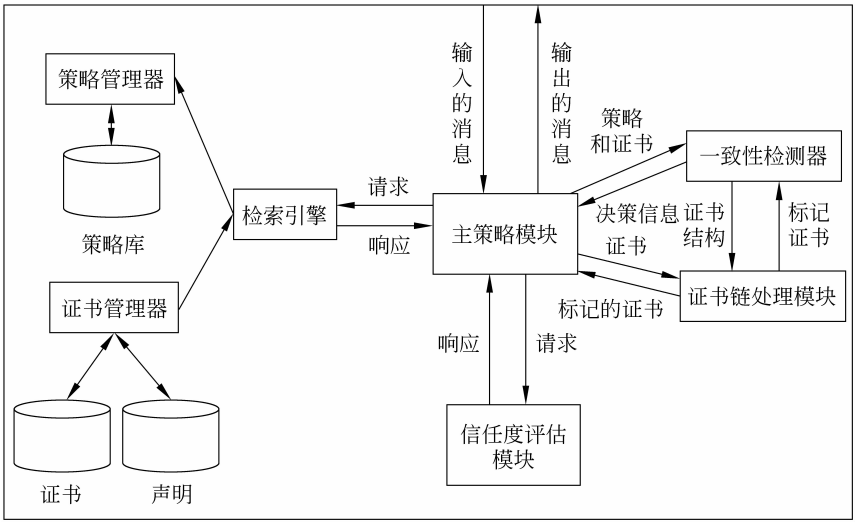


**图 5.1　信任协商系统模块图**

## 5.2　系统模块设计

### 5.2.1　主策略模块

当一个远程消息被可视化模块处理后，它将被传递到主策略模块。这个模块可以被称作一个协调者。对于一个信任协商会话，它将收到的消息交给合适的策略模块进行处理。当主策略模块收到一个远程消息，并且可以获取协商状态，主策略模块生成向远程协商方反

馈消息的内容。主协商模块可以无限次地调用一致性校验模块、证书链处理模块以及策略管理器、证书管理器。

主策略模块(见图 5.2)主要提供如下方法。

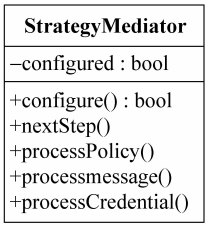


图 5.2　主策略模块

Nextstep()：该方法的参数为协商消息，返回类型也为协商消息类型。该方法的功能是处理接收到的消息，并返回反馈给对方的消息，同时需要获取会话 ID 以及状态信息。

processPolicy()：该方法的功能是查看访问控制策略是否能够被满足，如果存在这样的信任凭证集合，那么查看这些凭证是否可以被披露。如果有凭证受保护，那么查找对应的访问控制策略。

processCredential()：该方法的功能是查看凭证集合中的凭证是否被保护。如果不受保护，将其添加到反馈消息中；如果被保护，则查看访问控制策略是否已经被满足。如果满足凭证可以被披露，将其添加到反馈消息中。

## 5.2.2　检索引擎

检索引擎主要负责检索证书和策略。此模块接受其他模块的请求，处理这些请求，反馈检索的结果。

由于检索引擎主要提供两种类型的检索：证书检索和策略检索，所以在模块中设计了父类 Query 以及它的两个子类 ProfileManager 和 PolicyManager，实现了父类的 processQuery 方法，如图 5.3 所示。这两个子类还实现了管理凭证和策略的功能，在后面的模块对此作介绍。

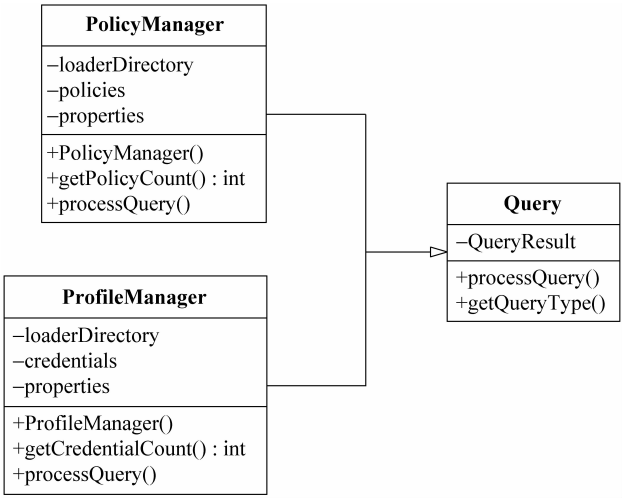


图 5.3　检索引擎

QueryResult 将检索到的证书集合或者策略集合以数组的方式返回。

检索引擎提供的方法如下。

processQuery()：该方法的主要功能是获取检索类型，根据检索类型调用相应的类处理检索请求。

getQueryType()：该方法的主要功能是返回检索的类型。

### 5.2.3 策略管理器

策略管理器负责加载用户硬盘上的策略文件，并且保证主策略模块可以通过检索引擎访问这些策略（见图 5.4）。此组件可以通过用户设置的策略文件目录来加载策略。

策略管理器提供如下方法。

getPolicyCount()：该方法返回加载策略的数目。

processQuery()：该方法的参数是检索请求，返回检索结果。根据请求类型，该方法可以实现检索所有的策略，也可以根据策略的 ID 检索策略。

### 5.2.4 证书管理器

证书管理器负责加载证书和声明，配置文件中同样需要指定证书存放的目录（见图 5.5）。

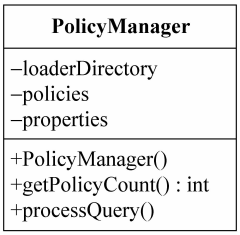


图 5.4 策略管理器

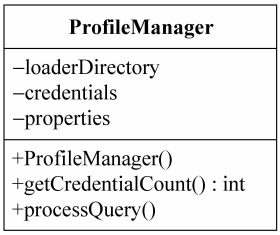


图 5.5 证书管理器

该管理器提供如下方法。

getCredentialCount()：该方法返回当前加载凭证的数目。

processQuery()：该方法的参数是检索请求，返回检索结果。根据请求类型，该方法可以实现检索所有的证书，也可以检索声明。

### 5.2.5 一致性校验器模块

一致性校验器是自动信任协商系统的一个重要组成部分，它的主要功能是判断给定的信任凭证集合是否能够满足针对特定资源定义的访问控制策略（见图 5.6）。一致性校验器首先验证信任凭证的有效性，进行匹配时过滤掉无效的证书。

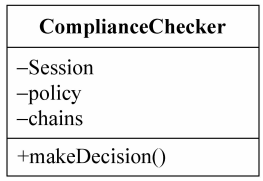


图 5.6 一致性校验器模块

一致性校验器提供了 makeDecision()方法，该方法的参数是信任凭证集合、访问控制策略和对资源的访问请求，它的返回结果是判定结果和引导信息。一致性校验器将判定的结果反馈到协商策略模块。

### 5.2.6 可视化模块

可视化模块的主要功能是将收到的远程消息以及发送的消息以固定的格式显示到窗口（见图 5.7）。它的输入是一条消息以及消息的方向。它分析消息的内容，并将具体内容以固定的格式显示到窗口。可以设置窗口为显示或者隐藏。

可视化模块提供了如下方法。

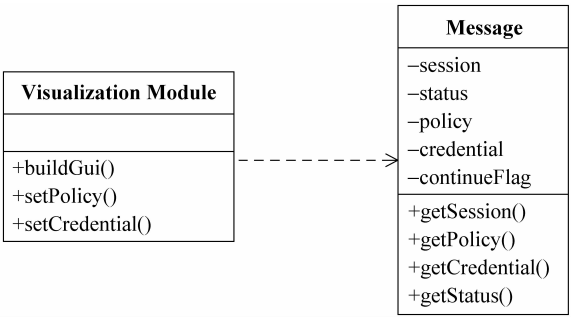


图 5.7　可视化模块

buildGui()：该方法设置输出窗口的样式以及窗口中显示的消息内容，包括消息方向、协商状态、加载消息中包括的凭证或者策略信息。

setPolicy()：该方法将消息中的策略添加到窗口。

setCredential()：该方法将消息中的证书添加到窗口。

Message 对象以固定的格式封装了消息的内容。双方之间传递的消息均被封装到 Message 中。主策略模块返回以及接收的消息都是 message 类型的。可视化模块的输入为 message 类型，通过解析 Message 中包含的信息进行显示。

### 5.2.7　信任度评估模块

信任度评估模块的主要功能是评估用户的信任度，并根据用户的信任度调整协商策略和访问控制策略(见图 5.8)。信任度评估的算法已经在前面作了介绍，这里主要介绍信任度评估模块如何获取用户的信任度，即信任度模块提供的调用接口。

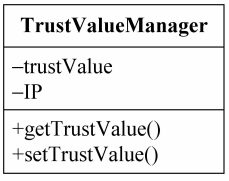


图 5.8　信任度评估模块

信任度评估模块通过用户的 IP 获取用户的信任度 trustValue。

该模块提供了 getTrustValue()方法：该方法的输入是一个 IP 地址，函数查找信任度数据库，返回此 IP 地址对应的信任度等级。

### 5.2.8　外部接口设计

信任协商系统为外部程序提供了一个外部调用接口(见图 5.9)。外部程序只需要调用 AATN 类，并使用 AATN 类提供的一些方法来完成信任协商的功能。外部调用程序还必须为本类提供一些配置信息，来引导信任协商会话过程。信任协商过程中双方传递的消息被封装在 Message 内，AATN 处理信任协商中对方发送过来的消息，将消息进行处理后，生成反馈的消息，封装为 Message 对象发出。

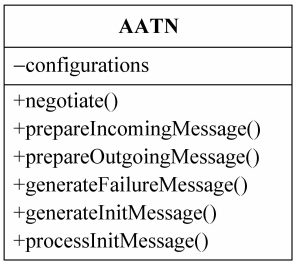


图 5.9　外部接口

外部接口提供了以下方法。

Negotiate()：该方法为本接口提供的主要信任协商方法。该方法接收远程方的消息，处理消息中包含的各种元素并产生一个响应。

prepareIncomingMessage()：该方法为 Negotiate 方法提供支持，Negotiate 通过调用该方法处理收到的协商远程方消息。在本方法的实现中需要调用信任协商系统的其他模块。

prepareOutgoingMessage()：该方法为 Negotiate 方法提供支持，Negotiate 通过调用该方法处理发送到协商远程方的消息。

generateFailureMessage()：当协商过程中出现错误时，该方法生成消息，告知协商远程方终止会话。

## 5.3 AATN-Jess 策略语言

### 5.3.1 策略语言设计需求

策略语言用来描述策略，因此策略语言的语义是否清晰，描述是否详尽，决定了策略对于信任协商系统的可读性。信任协商系统对策略语言的要求主要体现在以下诸方面[1]。

(1) 定义良好的语义。一个定义良好的策略语言应该具有简单、紧凑和定义规范的语义，即使用该策略语言编写的策略，其含义与该语言的特殊应用无关。

(2) 单调性。信任协商对策略语言的单调性需求表现在：证书与策略的披露应对用户的授权产生影响；额外证书/策略的披露只能让用户获得额外资源/服务操作的权限。

(3) 证书结合。不同证书描述了特定主体不同的特征。信任协商要求策略语言具有很强的表达能力，能够使用“交”、“并”等操作将不同的证书结合起来，以满足需要提交多个证书的策略。

(4) 认证。信任协商的参与方均有多个证书，以便通过证书交换来建立信任关系。在系统运行过程中，证书提交者需证明其拥有与证书签名使用的公钥相对应的私钥信息，以确保证书的有效性。

(5) 属性值约束。通常一个证书就是一个结构化的对象，它包含关于主体属性的信息，name-value 就是属性信息的典型代表。证书可关联到某种指定的证书类型，用来简化证书规范和管理。

(6) 内部证书约束。为了更好地评估远程参与方的属性，即使参与方使用了不同的密钥，策略也应该可以表达一些约束，用来比较属于同一主体的不同证书的值。

(7) 证书链。当某一证书中的主体是证书链中下一证书的发布者时，策略语言应提供足够的描述能力来表达和约束证书链。

(8) 传递闭包。在特定的环境中，信任关系具有传递性。这要求策略语言允许策略编写者来描述信任链中的数量和类型约束。

(9) 外部函数。在协商过程中，需要一个标准的函数库来规范对诸如日期、时间和货币等的操作和比较。

(10) 本地证书变量。当处理资源的标准离线策略时，本地证书变量可使这些策略自动地与其证书关联起来，提高策略与证书的匹配效率。

(11) 检测提交者。策略编写者可以指定策略中哪些原子策略应该由访问者提交的哪些证书来满足。

(12) 敏感策略保护。敏感策略里可能包含一些个人隐私或商业机密。策略语言具有

敏感信息保护机制，以避免/防止重要信息外泄。

(13) 具有互操作语言的统一形式和使用：这种需求强调了协商方法的应用能力，即在设计策略语言时，须充分考虑其是否可以在真实的环境中使用，以及是否可以集成到已有的上下文中。

### 5.3.2 AATN-Jess 语言特点

Jess(Java Expert System Shell)语言[2]是 1995 年由美国 Sandia 国家实验室分布式系统计算组成员 Ernest J. Friedman-Hill 用 Java 实现的一个经过扩充的 CLIPS 版本。它除了继承了 CLIPS 的特点外，还具有支持正向和逆向推理，可以在系统运行环境下直接调用 Java 的类库等特点。这些特点将专家系统的开发过程同功能强大的 Java 语言结合起来，使采用 Jess 语言开发的专家系统具有良好的移植性和嵌入性，可以方便地应用到网络上的不同机器中。另外，Java 多线程机制使 Jess 可以与其他应用程序并发执行，同步机制保证了对共享数据的正确操作，通过使用不同的线程完成特定的行为，就可以很容易地实现网络上的实时交互行为。这些特点都符合 P2P 网络环境下信任协商系统的要求，比较适合作为 P2P 网络信任协商系统中的安全策略语言。但由于 Jess 语言是为专家系统设计开发的，其某些地方不符合信任协商安全策略语言的要求。因此本节在 Jess 基础上设计了 AATN-Jess 语言，对原有 Jess 语言进行了简化与提升，使其更加符合自适应信任协商系统的要求，增强了协商的效率。

AATN-Jess 在保留了 Jess 语言的环境友好、方便用户编写策略等特点的同时，对 Jess 的语法结构进行了修改，取消了 Jess 语法结构中需要将证书中涉及的内容封装为若干对象的要求，使得 AATN-Jess 语法更容易理解，同时也提高了信任协商系统对策略语言的解析效率。AATN-Jess 相对于其他安全策略语言具有如下特点：

(1) 具有良好的系统兼容性。AATN-Jess 是在 Jess 的基础上进行简化和提高的，其也是用 Java 语言进行开发的。Java 语言具有跨平台特性，这样有利于 AATN-Jess 应用于不同的系统，这一特性也符合 P2P 网络的要求。同时在设计过程中将代码封装为不同的类，便于开发过程中的调用，也有助于日后根据自身需要进行修改。

(2) 简洁的语法结构及较高的协商效率。AATN-Jess 支持面向过程的编程方式，它提供了一些语句来控制规则后件的操作流程，如使用 if…then…else 和 while…do 语句，这样它就能很有效地利用面向过程编程的优势。AATN-Jess 的这些特性使系统拥有很强的知识表示能力。同时 AATN-Jess 去掉了 Jess 在编写过程中封装对象的要求，简化了语法结构，也有助于提高协商效率。

(3) 完善、友好的开发环境。AATN-Jess 提供了两个交互式的、命令行的开发环境，但也可以使用文本编辑器编辑代码，然后再通过系统命令以批处理的方式载入到系统中。这样使用户更容易上手，不需要了解 Java 知识就可以编写自身所拥有的敏感资源和敏感证书的保护策略。

### 5.3.3 AATN-Jess 语法结构

AATN-Jess 语言包含 9 个语言要素，即 Template(模板)、Rule(规则)、LHand-rule(规则左键)、RHand-rule(规则右键)、Pattern(模式)、Match(配比)、Function(函数)、Solt(槽

值)和 MultiSolt(多槽值),所有访问控制策略文件都是由这 9 个语言要素嵌套组合而成的,其嵌套语法如图 5.10 所示。

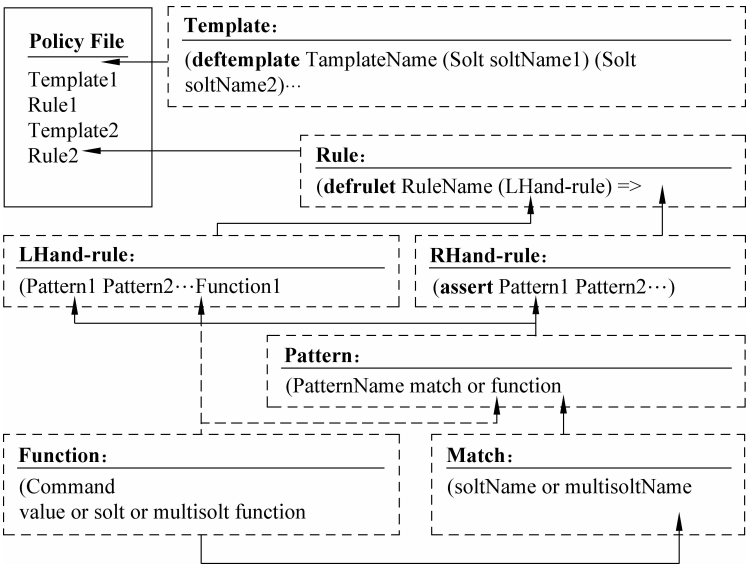


**图 5.10　AATN-Jess 语法逻辑示意图**

其中:

(1) 每一个访问控制策略文件(Policy File)都包含一个或者多个模板(Template)和规则(Rule)。

(2) 每一个模板是若干 Solt(槽值)和 MultiSolt(多槽值)的组合,置于一个小括号"()"内,用于存放访问控制策略要求的证书属性。

(3) 每一个规则由一个规则左键(LHand-rule)和一个规则右键(RHand-rule)组成,左键和右键之间以"=>"链接并置于一个小括号"()"内,用于定义访问控制策略对证书的约束,在证书集合能够满足规则左键的约束时,产生一个断言,即规则右键。

(4) 每一个规则左键包含若干模式(Pattern)和函数(Function),用于定义访问控制策略对证书集合的约束。

(5) 每一个规则右键包含若干模式(Pattern),用于包装规则产生的断言。

(6) 每一个模式由若干配比(Match)和函数(Function)组成。

(7) 每一个配比都包含两部分,前半部分是变量名(soltName 或者 multisoltName),后半部分是与之配比的值(value)或者约束这个值的函数(Function),意为选取该变量的值等于当前配比提供的值或者函数值的证书。

(8) 每一个函数包含一个操作符(Command)和若干操作对象(包括值、变量、值或者变量的函数)。

(9) 此外,AATN-Jess 语言涉及的数据类型和变量等遵循 Jess 语言的语法规则。

为了直观地说明 AATN-Jess 的语法,本节给出以下示例:给出一个简单的策略,以观察儿童具有的技能。有如图 5.11 所示的 Credential 1、2、3 三个证书,图 5.12 所示的策略 Policy_example.clp 可以获得所有儿童(age 属性值小于或者等于 6)的所有技能(skill)。

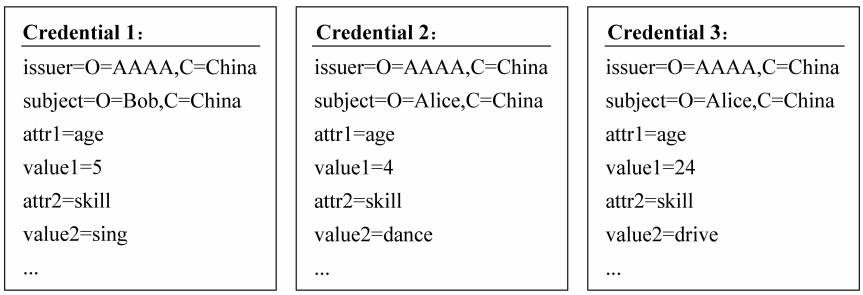


图 5.11　标识个人信息的证书

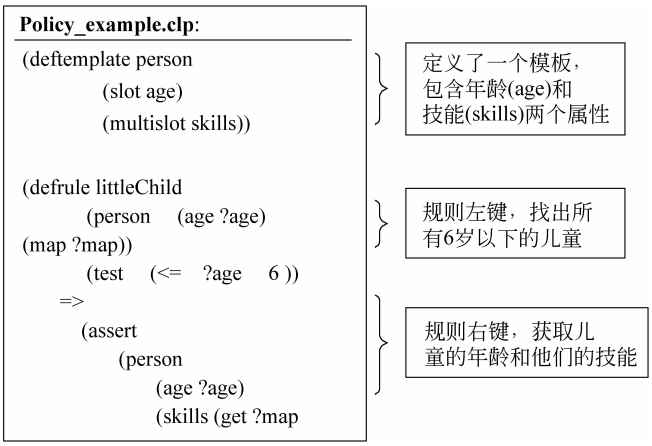


图 5.12　观察 6 岁以下儿童技能的策略

本例中，策略 Policy_example.clp 最终只命中 Credential1 和 Credential2 并获得相应的技能 sing(唱歌)和 dance(跳舞)，因为“(test (<=?age 6))”排除了所有 age 属性值大于 6 的证书。

### 5.3.4　AATN-Jess 策略语言编辑器

为了方便用户定义自己的访问控制策略，我们向用户提供了策略编辑器 PEditor 1.0。根据以上 AATN-Jess 语法规则，我们将 AATN-Jess 语法逻辑组织成 DOM 树(我们称之为策略编辑树)，树中每一个节点都是语法中的一个元素，用户只需在树上添加节点就可以定义访问控制策略，而不必担心语法格式错误造成的策略不一致问题。

为了演示 PEditor 1.0 的使用，我们编辑了前面示例中的 Policy_example.clp 文件，使用 PEditor 1.0 编辑的 Policy_example.clp 文件的策略编辑树如图 5.13(左边部分)所示。图 5.13 标出了策略编辑树上的节点和目标策略文件代码之间的联系，这种直观的联系便于用户理解，从而使得 PEditor 1.0 变得更加实用。策略编辑树上各节点的前置图标与各节点的节点性质名是一一对应的关系，关联如下：T—Template、S—Solt、Ms—Multisolt、R—Rule、Rl—LHand-rule、Rr—RHand-rule、P—Pattern、M—Match、F—Function，V—Value。通过不同的前置图标，用户可以了解每一个节点的意义，方便编辑和修改。

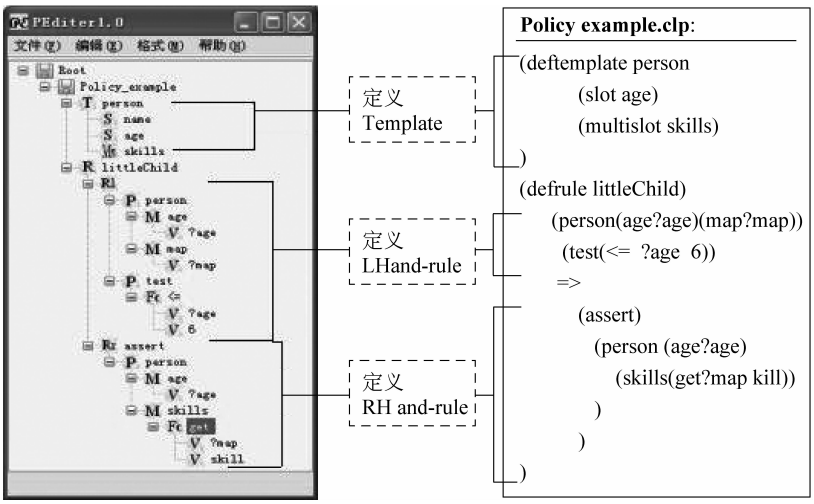


**图 5.13　Policy_example. clp 策略编辑树与原策略文件的对比**

## 5.4　本章小结

本章在第 4 章提出的自适应自动信任协商模型的基础上，主要讨论了自适应信任协商系统的设计和实现问题。首先，阐述了系统总体设计和模块设计，然后着重介绍了在系统设计和实现中使用的 AATN-Jess 策略语言，分析了 AATN-Jess 的语言特点和语法结构。

## 参 考 文 献

[1] 廖振松，金海，李赤松，等. 自动信任协商及其发展趋势[J]. 软件学报，2006，17(9)：1933-1948.

[2] 张国煊，张翔. 如何用开发专家系统[J]. 计算机与现代化，2003，1：29-31.

[3] Smith B，Seamons KE，Jones MD. Responding to policies at runtime in TrustBuilder. In：Proc. of the 5th Int'l Workshop on Policies for Distributed Systems and Networks. Washington：IEEE Computer Society Press，2004. 149-158.

[4] Shaoxu Guo，Wenbao Jiang. An Adaptive Automated Trust Negotiation Model and Algorithm [J]. In：International Conference on Communications and Intelligence Information Security. Nanning，Guangxi Province，China：IEEE Press，2010：130-134.

[5] Wenliang Chen，Wenbao Jiang，Analysis and Design of an Adaptive Automated Trust Negotiation System，2011 IEEE International Conference on Mechatropic Science，Electric Engineering and Computer (MEC2011)，August 2011 (EI：20114114423112).

[6] 郭少旭. 自适应信任协商技术研究[D]. 北京信息科技大学硕士学位论文，2010.