一个适应性的人工免疫系统的软件构架*)

严 悍 张 琨 李千目 刘凤玉
(南京理工大学计算机科学与技术系 南京 210094)

摘 要 适应性是人工免疫系统(AIS,Artificial Immune System)的重要特性之一。在AIS软件开发应用中,数据源的进化和学习算法的进化是两个有复杂关联的适应性问题。为此我们扩展并改进了已有的AIS构架,提出一个新的适应性软件构架。该构架以基因计算为中心,扩展了元基因来适应数据源的进化,并设计了可接入学习算法构件和算法验证机制来解决算法进化的适应性问题。在该构架支持下,数据源的进化独立于学习算法的设计,同时使学习算法能适用于多种数据源且能独立进化。该构架可简化AIS软件的复杂性,可提高AIS开发应用的效率,也有助于实现将来的自适应的免疫计算。

关键词 适应性,人工免疫系统,软件构架,元基因,进化计算

An Adaptable Software Architecture for Artificial Immune System

YAN Han ZHANG Kun LI Qian-Mu LIU Feng-Yu
(Department of Computer Sci. & Tech., Nanjing University of Science and Technology, Nanjing 210094)

Abstract Adaptability is one of the most significant properties of AIS (Artificial Immune System). However, AIS software development and application are facing two complex and tangled adaptability problems: the evolution of data sources and evolution of multiple learning algorisms. We expand and improve on the existing AIS architecture, and propose an adaptable software architecture for AIS. The architecture is gene computing-centered, with meta-gene expanded to adapt to the evolution of data sources, and with pluggble learning algorism components and algorisms validation mechanism to adapt to the multiple algorisms revoution. With this architecture, evolution of data sources will be independent with learning algorisms; on the other hand, learning algorisms reuse for multiple data sources and evolve independently. This architecturesimplifies complexity of AIS software, and hence improves the efficiency of AIS development and application. Further, it is helpful to realize future self-adaptive immune computing.

Keywords Adaptability, Artificial immune system, Software architecture, Meta-gene, Evolution computing

1 引言

人工免疫系统(AIS,Artificial Immune System)作为人工智能的一个分支日益受到关注。AIS利用计算机和网络来模拟生物免疫系统的工作原理,先将外界数据源转换为基因表示,再通过基因计算实现学习、记忆、识别等特性,并应用于信息安全性、异常检测、错误诊断等领域[1,2]。适应性是生物免疫的一个重要特性,但目前对于AIS仍是复杂的、亟待解决的问题[1]。已有研究主要从学习算法的角度探讨如何适应新的异物(nonself),但从AIS软件开发应用的角度来看,还有其它适应性问题,如数据源的进化、多学习算法的进化等。这些适应性问题难以采用某种算法来解决,本文从软件构架(software architcture)的角度来探讨AIS适应性设计的方案,这对于AIS开发和应用具有重要意义,也有利于将来实现自适的免疫计算。

适应性的本质在于适应环境或条件的变化。适应性对于AIS尤为重要,有两个原因:①适应性是生物免疫的本质特性,它保证了生物个体和种群在自然选择的环境中得以生存繁衍。AIS模拟生物免疫系统的特性,故此适应性对于AIS也同样重要。②适应性对于AIS来说,不仅保证其生存能力,而且使AIS能方便地应用于多个领域,以充分发挥AIS的基因计算的普适性作用。但AIS本质是基于计算机和网络的一种人工制品(artifact),通常利用二进制串的操作来实现基因计算[1,2]。生物免疫系统与AIS之间存在巨大差异,这也使AIS的适应性设计复杂化。

研究表明,AIS的适应性有多个方面:①异物适应性。在数据源结构一定的前提下,当出现新的异物模式时,通过适应性学习算法生成检测子(detector),以识别新异物的下一次入侵[1]。目前已有多种算法,如负选择[1,4]、克隆选择[4]、动态克隆选择[5]等。多数AIS研究集中在适应性学习算法方面,而且以学习算法为中心建立AIS构架,如ARTIS[1]。②数据源适应性。如何适应外界数据源的结构变化,如何处理多种不同的数据源,如何使数据源的进化或者改变能独立于学习算法设计。③多算法进化的适应性。如何能方便地实现多种学习算法而且能独立进化,如何选择特定算法以适应特定需求(我们探讨多个学习算法的进化问题,区别于具体算法的进化,故此我们称之为多算法进化)。④分布式适应性。当分布式免疫系统的拓扑结构或免疫策略发生变化时,各免疫单元如何适应(关于分布式免疫计算另文探讨)。

数据源适应性与多算法进化的适应性对于AIS软件设计是必需考虑的问题。如果缺乏这些适应性,当环境或条件发生某些改变时就难以适应。例如,当外界数据源发生结构变化时,比如增加了新属性,就可能导致有的学习结果不能再发挥作用,已实现的学习算法也必须修改。而数据源的进化往往是客观需要且难以预料。比如,一个基于AIS的乳腺癌诊断系统中的数据源来自各种仪器的检测结果,当增加新的仪器或改进检测手段时,就会改变数据源的结构。另一方面,学习算法是易变的且有多样性,而且一个AIS往往要实现多

*)基金项目:国家自然科学基金资助项目(60273035),南京理工大学青年学者基金资助。严 悍 副教授,博士,研究方向为软件工程与信息安全;张 琨 博士,研究方向为信息安全;李千目 博士生,研究方向为网络性能与信息安全;刘凤玉 教授,博导,研究方向为软件方法与信息安全。

个算法以满足不同需求。例如，一个网络入侵检测系统有时要求实时性更强，而有时又要求更高的准确性。单个学习算法难以满足这样的要求，就需要多个学习算法共存，每个算法都能独立进化，而且在多个算法之间可优胜劣汰、新陈代谢。如果缺乏这些适应性，算法就难以维护和进化，降低 AIS 的生存能力，也会间接导致大量的重复工作，降低了 AIS 开发应用的效率。

为了解决数据源适应性和多算法进化的适应性问题，我们探讨 AIS 软件构架的适应性设计，有以下原因：①这些问题难以通过某种算法或过程来解决；②已有的 AIS 构架并未考虑这些适应性问题；③适应性作为一种特性，应建立在软件构架所提供的基本功能基础之上。我们采用的主要技术途径是①扩展和改进已有的 AIS 构架；②采用适应性软件构架技术[6]，其中包含了分层结构、元数据技术、接口及构件技术、法验证技术等。本文在第 2 节提出一个适应性的 AIS 软件构架的设计方案，该架构提供了 AIS 的基本功能，且能满足这些适应性要求；第 3 节比较相关工作并讨论相关问题；最后是结论。

2 适应性构架

图 1(a)是一个适应性的 AIS 软件构架。该构架具有三层的简单结构。以基因计算为中心，元基因层(Meta-Gene)主要针对数据源的适应性问题，可接入层(Pluggable)则针对多算法进化的适应性。基因层(Gene)实现基因库及核心的基因操作，并隔离数据源和学习算法，使两者可独立进化。当数据源进化或改变时，不会影响可接入层中的学习算法的设计。另一方面，因算法独立于具体的数据源结构，就能方便地实现新的算法，或进化已有的算法，可独立进化且可复用于多种数据源。此构架具有学习、记忆、识别等功能，同时能适应数据源的进化和改变，也能适应多个学习算法的进化。图 1(b)中描述各层中的主要元素，以及各层的接口设计。下面分别阐述各层的设计原理。

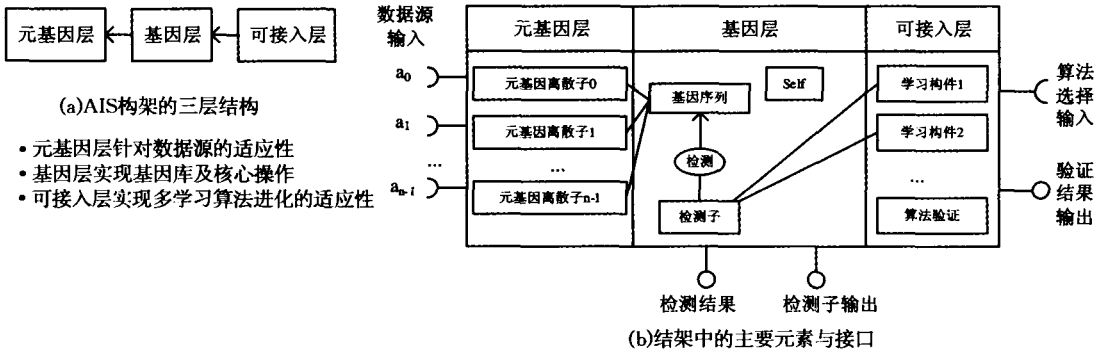

图 1 适应性 AIS 软件构架

2.1 元基因层

元基因层设计有三个功能：①根据数据源的各属性的特征，描述并存储各属性所对应的基因的数据结构规范及语义，我们称之为元基因(meta-gene)；②基因化(genelization)，即把数据源的属性值自动转换为基因表示的数据；③反基因化(degenelization)，当要把学习结果转换为可理解的形式时，就需要元基因作为规范来解释基因或基因序列的语义。

元基因层使 AIS 能适应外界数据源的变化。这些变化包括增加新属性、淘汰旧属性、改变属性，也能处理多个不同的数据源，各自具有不同的属性结构。例如，一个网络入侵检测系统可能同时要检测主机入侵，网络入侵和主机入侵需要处理不同的数据源。假如没有元基因层的设计，就难以适应这些变化。

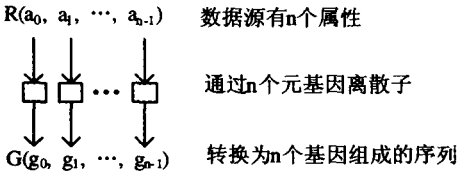

图 2 数据源的属性转换为基因并组成基因序列

不同数据源可能有不种形式的来源，如一个实时采集过程，或一个数据库，或一个数据文件。一个数据源包含多个属性，其模式可抽象为：$R(a_0, a_1, \cdots, a_{n-1})$，其中 R 表示据源的标识，括号中是一组属性。数据源 R 的一个实例就是 R 的各属性取一个值而组成的序列$(v_0, v_1, \cdots, v_{n-1})$，一个值对应一个属性。这样的一个序列将转换为由 n 个基因组成的一个基因序列(见图 2)。

建立元基因层需要两个步骤。第一步是建立元基因，明确数据源的属性规范并加以描述；该过程可采用一个“元数据挖掘”的方法来实现，即从属性的值来判断其属性的数据类型。元数据挖掘可以是自动进行的，大多数需要人机交互，这是因为确定属性的名字和语义往往需要人工参与。第二步是确定离散子(discretir，采用离散化技术把数据源的实例转换为对应的二进制表示的基因序列。按照属性的类型和值的分布，确定性值的离散区间，把属性值转换为二进制串，即基因表示。这个过程我们称为“基因化”。一个属性对应一基因；一个属性的值将转换为等长的基因。这样一个数据源的实例将转换为等长的基因序列。基因序列将在基因层中处理。对连续数值的离散化比较复杂，可参见文[8]。

数据源的适应性问题依赖于元基因结构和行为来解决。例如，当要添加一个属性时，就增加一个元基因，并确定其类型、名称、位长、离散区间等性质。无论是添加、修改还是删除属性，都不影响已有的学习算法。当需要处理多个数据源时，就建立多组元基因，各自独立存储，用标识符加以区分。事实上，一个网络入侵检测系统与一个乳腺癌诊断系统之间的区别仅在于元基因层具有不同的元基因，故此在元基因支持下，容易实现多领域的 AIS 应用。

总之，元基因层综合运用了元数据技术和离散化技术，把外界数据源转换为基因层中的基因序列，而且能适应数据源的进化或改变。

2.2 基因层

基因层的主要功能是：①按照元基因规范，提供基因、基因序列的存储和检索，一般称之为基因库；②提供关于基因的

基本的、公共的操作，如随机生成、匹配、负选择、检测、遗传、变异等。

基因层需要元基因作为其结构规范，同时为学习算法提供操作接口。基因层可设计自己的内部结构，如基因序列、检测子、Self 等；而且基因层的具体存储可有多种选择，例如，基于数据库、基于 XML、基于数据文件等。关于基因层的设计可参见文[1～4]。

基因层中最重要的一种基因序列集合是检测子。它表示异物的基因模式，是学习或训练的结果，用于对未知的基因序列进行识别或检测。一种数据源通常对应一个检测子集合（也可有多个检测子集合，以实现多重分类）。如果一个未知基因序列与该集合中任一个检测子发生匹配，就认为该序列是一个异物，并通过一个接口输出检测结果（见图 1(b)）。一个异物检测结果可认为是一个危险信号，将传播给其它系统，如排异反应系统。还有一个输出接口把检测子传播给相邻接的其它检测系统，这对分布式免疫系统非常有用。

基因层中另一种重要的基因序列集合是 Self（自体）样例。原因是多数学习算法要模拟生物免疫系统中的“负选择 negative selection”[1,3,4]。当一个检测子与 Self 中任一个基因序列匹配时，就应淘汰该检测子，只有不匹配的检测子才可能生存。我们在基因层中设计 Self 并实现负选择，目的是简化学习算法的设计。

2.3 可接入层

可接入层的设计功能如下：①把一种学习算法实现为一个可接入的构件，以方便接入新的学习构件，或进已有算法；②每种学习算法都可利用基因层提供的功能，生成新的检测子或进化已有的检测子；③提供算法验证机制，根据具体要求来选用合适的算法。

把学习算法作为可接入构件，有以下理由：①算法是易变且多样的，新的算法不断出现，已有算法也会不断改进，所以算法应该是动态进化的。②在实际检测或识别过程中，仅使用基因层中存储的检测子，通常不需要学习算法，除非是在线学习。③研究表明，目前尚不存在一种学习算法同时具有所有优势，比如，最快学习（即最短学习时间）、最高正检率（正检率 TP，True Positive）、最高准确性（TP-FP，其中 FP 为正误率 False Positive）、最高检测性能（即最少检测子），而且将来也不大可能出现这样的算法。故此一个 AIS 中应有多种算法并存。解决多算法的生存与进化问题的一种可行途径，就是把各算法实现为可接入的构件，再根据具体要求选择合适的算法（关于正检率 True Positive 和正误率 False Positive，请参见文[1,4]）。④可复用性。每一种学习算法对于特定领域的 AIS 来说都应该是适用的，例如，克隆选择算法应能同时适用于网络入侵检测与乳腺癌诊断。

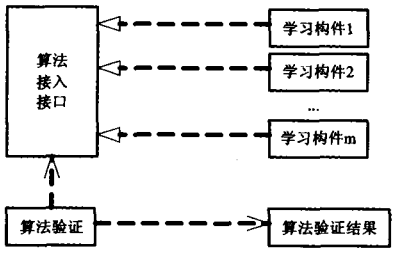

图 3 可接入层的设计

可接入层的适应性设计主要是[illegible]构件设计。在基因层与可接入层之间的接口设计有两个作用：隔离与连接。隔离的目的是隐藏设计的实现细节；而连接则为学习构件提供所需的样例，学习构件所生成的检测子也能转储到基因层。所有的学习构件都按统一的接入接口加以实现，这样多个学习算法就能方便地实现并接入（见图 3）。尽管学习算法之间可能存在某些相互作用，但各算法构件相互之间是独立的，这样使各构件可独立进化。在可接入层还设计了一个接口来选择要启用的学习构件（见图 1(b)），选用一个学习构件通常还需提供一组阈值。

除了接口设计之外，还需要一个算法验证机制，以获取算法的基准测试结果作为生存或淘汰的标准。在提供相同样例与相同基准的前提下，对多个学习算法进行验证。通过接口输出算法的验证结果，包括检测子数量、学习时间、正检率 TP、正误率 FP 等。这些结果及相应的阈值可作为适应性选择的依据。算法的选择可通过人机交互进行，也可自动进行。算法的验证往往采用标准的、通用的方法，如 10 轮交叉验证法（tenfold cross-validation）[4]。

总之，可接入层综合采用了接口及构件设计技术和算法验证技术，使学习算法可方便地实现并接入，并能对多个算法进行验证和比较，按不同的设计要求选择合适的算法。另一个好处是算法设计得以简化，这是因为算法都具有统一接口规范，而且能得到基因层的功能支持。

此外，一些基因分析算法也可作为可接入层的构件，例如分析关键基因、或基因之间之间的相关性等。这些构件往往采用数据挖掘的相关算法，分析检测子的基因特征，输出分析结果，对基因层一般没有副作用。把基因分析结果表示为可理解的形式，就需要元基因层的“反基因化”功能。

3 相关研究及讨论

Steven A. Hofmey 等提出一个 AIS 构架 ARTIS[1]，并指出适应性是目前 AIS 最欠缺的特性之一（另外两个是健壮性和自主性）。该构架主要探讨异物的适应性学习和识别问题。ARTIS 中的检测子生命周期、Self 基因序列和负选择等机制对我们的软件构架具有指导意义。不同之处在于我们所针对的适应性问题更为宽泛。除了异物适应性之外，数据源适应性和多算法进化的适应性也很重要。我们的构架相对于 ARTIS 是一种适应性的扩展与改进，扩展了元基因层以适应数据源的变化，改进了可接入层中学习算法的进化机制，这对于 AIS 的软件实现和跨领域应用具有重要价值。

Steve Cayzer 等指出种群基因库会随时间而进化[3]，导致基因库与预期环境之间的偏差，建议一种“元学习 meta-learning”的途径来适应这种进化。这种思路已体现在我们构架中的元基因层和部分基层的设计上。进一步，我们认为各领域 AIS 应用之间的区别主要表现在元基因所处理的不同数据源，故此我们把元基因与基因库分隔开，使元基因可进化以适应不同的数据源，而且使数据源进化不影响基因库的操作，也不影响已实现的学习算法。

Jungwon Kim 等研究 AIS 并用于网络入侵检测[4,5]，它所采用的离散子、离散化方法、克隆选择和 10 轮交叉验证法对于我们的构架设计具有指导作用。不同之处在于我们把离散子与元基因相结合，形成一种紧凑结构，既能表示属性结构，也能完成基因化。另一个区别是检测过程不同，我们采用与学习算法相一致的检测，即在离散化之后对基因序列进行检测，而不是用检测子对数据源进行检测，这样更为简单。最主要区别在于多算法进化机制，我们的构架中把学习算法设计为独立的可接入构件，如克隆选择、动态克隆选择（算法验证结果另文再叙），使系统能适应多方面的需求。

适应性软件构架采用可复用构件与其它技术，在分布式、异构环境中简化软件进化的复杂性[6]。我们采用了构件技术来设计可接入层的各种学习算法，以简化算法设计和算法进化的复杂性，而且使算法构件对于不同领域的 AIS 应用具有可复用性，以提高 AIS 开发应用的效率和质量。此外，我们的构架也潜在支持分布式免疫计算，这是进一步的工作。

自适应(self-adaptive)系统能根据环境或条件的变化自行改变其结构或行为，无需人工参与[7]。AIS 应该是一种自适应的系统，但目前还难以全面实现，根本原因在于其复杂性。本文从软件构架设计的角度探讨 AIS 的适应性解决方案，虽然其中大多适应性工作需要人工参与，但这种基于构架的途径对于将来的自适应免疫计算具有指导作用。

结论 适应性是 AIS 的一个重要特性，除了学习算法所针对的异物适应性之外，在 AIS 的软件开发应用方面还存在数据源进化和多算法进化等适应性问题。为此我们扩展并改进了已有的 AIS 构架，提出一个新的 AIS 软件构架，以满足这些适应性要求。该构架以基因计算为中心，设计了元基因来适应数据源的变化，并设计了可接入构件和算法验证机制来解决多算法进化的适应性问题。在该构架支持下，数据源的进化或改变不影响学习算法的设计，同时使学习算法能适用于多种数据源。该构架简化了 AIS 软件开发的复杂性，可提高开发应用的效率，对于实现下一步的分布式免疫计算和将来的自适应免疫计算具有指导意义。

参 考 文 献

1 Hofmeyr S, Forrest S. Architecture for an Artificial Immune System. Evolutionary Computation. Morgan-Kaufmann, San Francisco, CA, 2000, 7(1): 1289～1296

2 Ji Z, Dasgupta D. Artificial Immune System (AIS) Research in the Last Five Years. Published in the proceedings of the Congress on Evolutionary Computation Conference (CEC) Canberra, Australia, 2003

3 Cayzer S, Smith J, Marshall A R J, Kovacs T. What have Gene Libraries done for AIS? Digital Media Systems Laboratory, HP Laboratories Bristol, HPL-2005-116, 2005. www.hpl.hp.com/techreports/2005/HPL-2005-116.pdf

4 Kim J, Bentley P J. Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Clonal Selection with a Negative Selection Operator, the Congress on Evolutionary Computation (CEC-2001), Seoul, Korea, 2001. 1244～1252

5 Kim J, Bentley P J. A Model of Gene Library Evolution in the Dynamic Clonal Selection Algorithm. In: Proceedings of the First International Conference on Artificial Immune Systems (ICARIS2002) Canterbury, 2002. 175～182

6 Aniort P. A Distributed Adaptable Software Architecture Derived From a Component Model. ACM, Computer Standards & Interfaces, Special issue: Adaptable Software Architectures. ISSN: 0920-5489. 2003, 25(3): 275～282

7 Oreizy P, Gorlick M, Taylor R N, et al. An Architecture-Based Approach to Self-Adaptive Software, IEEE Intelligent Systems, 1999, 14(3): 54～62

8 Fayyad U M, Irani K B. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, Proceeding of The Thirteenth International Joint Conference on Atificial Inelligence, 1993. 1022～1027

(上接第 259 页)

能的最优解为表示“建造行为”的谓词，即“修建”，并建议用户选择合适的汉语词，供用户确认。

4.2 实例检索

一旦用户同意优选结果，系统将自动选用该结果，表示建造行为的谓词类 build，并在分析出该句的句法语义表示后，在实例库中以该词对应的中间语言概念词语及其语义句法关系为索引，查询相应的实例。设在实例库中，以 build 为索引的实例包括有“创立理论”、“增进情谊”、“建造房屋”等中间结果(虽然在中间语言词汇集中，中间语言词语与汉语的义项对应，但为增强系统鲁棒性，我们有时以光杆词语来索引，以适应不同用户选择不同中间语言词语造成的理解偏差，毕竟对义项的把握并非每个人都感觉一致)。在这些实例中，谓词和体词之间都存在 gol 关系(从无到有地产生某种结果这样一种行为，而体词表示的结果本身)。

(1) gol(build(gol ＞abstract thing), theory(icl＞abstract thing))

(2) gol(build(gol ＞relation), will(icl＞relation))

(3) gol(build(gol ＞building), house(icl＞building))

build 的语义类暂时未列出来。在以上检索到的实例中，theory，will 的语义类别是“抽象物”和“关系”，而(3)中的 house 语义类属于建筑物。与分析所得的“塔”的语义属性同类，因此优先选择(3)确定用户输入词“修建”的中间语言词汇是 build(gol＞building)。而在实例库中，(3)体现的汉语对应表示是“建造房屋”。

4.3 知识积累

根据第 2 步的实例匹配处理，系统将生成“建造塔”，并以汉语形式反馈给用户。如果用户对此予以确认，则系统将“建造塔”存入用户实例库 LocalBASE 中，并将此成果提交发送到给多语平台的后端知识服务器，由有权限的系统管理员确定是否追加到系统平台的知识库 GlobalBASE，作为全局共享的知识。

如果用户不满足系统反馈的结果，可以通过多次交互直至产生所需语句。最终的中间语言处理表示结果如图 2 所示。配合实例参考和规则生成，则形成系统反馈的结果“人们开始建造一座高塔”。而最终形成的各类关系和属性同时将作为翻译成果存入实例知识库中，供后期翻译服务时继续采用。

参 考 文 献

1 Boitet C. Advantages of the UNL language and format for web-oriented cross-lingual applications. Seminar on linguistic meaning representation and their applications over the World Wide Web, 2000

2 常宝宝，詹卫东，柏晓静，等. 服务于汉英机器翻译的双语对齐语料库和短语库建设. 见：第二届中日自然语言处理技术国际研讨会论文. 北京大学，2002

3 Déjean H. Learning Syntactic Structures with XML. Proceedings of CoLL-2000 and LLL-2000, Lisbon, Portugal, 2000

4 Schütz J. One web, One Language: The universal networking language. 2003

5 Mitamura T, Nyberg E. the KANTOO Machine Translation Environment. In: Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future, Cuernavaca, Mexico, Oct. 2000

6 Sabarís M F, Alonso J L R, Dafonte C, et al. Multilingual Authoring through an Artificial Language. EAMT Summit VIII, Santiago, Spain. Sep. 2001

7 Senellart J, Boitet C, Romary L. SYSTRAN New Generation: The XML Translation Workflow. In: Proceedings of MT Summit IX, New Orleans, USA. Sep. 2003

8 Uchida H, Zhu M. The Universal Networking Language beyond Machine Translation. International Symposium on Language in Cyberspace, Seoul, Korea. Sept. 2001

9 熊文新. 基于中间[illegible]语生成规则处理. 见：1998 中文信息处理国际会议论文[illegible]学出版社，1998. 515～523

10 熊文新. 中间语言[illegible]程中的增强处理. 计算机工程与应用，2005(9)：171～173