

3.6 模型与系统的关系

大多数的相关工作使用了超过一种的模型来建模和管理自适应系统,其中,常见的模型间的关系包括功能性需求模型、结构模型或行为模型对非功能性需求模型的影响关系,需求模型、结构模型、行为模型彼此之间的追踪关系,环境模型与软件模型之间的约束关系,软件模型与运行时系统之间的同步/镜像关系等。

功能性需求模型对非功能性需求模型的影响通常用贡献链接进行建模。一个贡献链接存在于一个目标(或特征)与一个非功能性需求之间,表示该目标(或特征)的实现/拒绝能给非功能性需求的实现/拒绝提供一定程度的证据。按照目标(或特征)对非功能性需求的影响是正还是负,可以将贡献链接的类型分为“帮助”和“伤害”,或者“++”“-”“+”和“-”。文献[21,23]即采用了这样的定性的贡献关系。为了便于基于非功能性需求的实现程度建立效用函数,文献[19,20]将定性的贡献关系进行量化,以支持定量的在线推理。文献[10,33]可以通过在线学习自动地学到特征对非功能性需求的贡献关系,学习到的结果是一个以特征值为自变量、功能性需求的满足程度为因变量的贡献函数。

结构模型对非功能性需求模型的影响通常会在各备选的构件、类或服务等上面进行标注。例如,文献[8,19,33,41]都在备选的构件上标注了其对于各非功能性需求的影响,代表了选择该构件后期望的吞吐量、延迟和交互协议等。行为模型会在状态或转移上标注其对于非功能性需求的影响。例如,文献[26]在代表服务的状态上标注了执行该服务消耗的时间、对可用性的影响等,在转移上标注的概率值可用来计算可达性需求的满足程度等。

高层的需求模型通常会与低层的结构或行为模型之间建立起映射关系,即,将一个需求模型中的元素映射到一个或一组结构/行为模型中的元素。两个模型之间的同步可以通过模型转换语言或者现有工具来实现。文献[10]在特征模型与体系结构模型之间建立了映射关系,使用模型转换语言实现两个模型的双向同步。文献[11]将一个特征映射为一个构件的集合,当该特征被绑定时,集合中的构件将被全部插入系统,当该特征被解绑定时,集合中的构件将被全部去除。文献[19]在目标模型与体系结构之间建立起了映射关系,将一个目标映射到一个设计问题,设计问题的每一个备选的设计方案被映射到了体系结构模型的一个片段。文献[23]将叶子级别的目标映射到体系结构的构件,可选目标的选择被映射到切换连接子,以实现构件的绑定和解绑定。

有的相关工作同时采用了与系统联系更紧密的设计模型以及更适合进行自适应验证或决策的分析模型。设计模型与分析模型可以通过模型转换语言进行同步,或者通过现有工具自动地根据设计模型生成分析模型。文献[39]使用双向模型转换语言,将建模了不同关注点的多个体系结构模型与镜像了系统当前结构的反映模型之间通过模型转换语言进行同步。文献[26]将一个包含可选片段的状态图转化为马尔可夫链进行分析。文献[31]采用中间模型作为从设计模型到分析模型转换的桥梁,设计模型可以通过两次模型转换生成分析模型。文献[16]采用工具将特征模型自动地转化为扩展的顺序图,再将顺序图转化为马尔可夫链模型以进行分析。

环境模型与软件模型之间的约束可以通过约束规则或自适应规则进行规约。文献[11,20,22,30,42]采用规则规约了环境模型与软件模型之间的约束关系,包括:当某个环境特征被选定时,特定的软件特征必须被选定(或不能被选定);或者当某个环境表达式被满足时,特定的软件特征必须被选定。与基于规则推理的方法不同,在线进行自适应规划的方法不会在离线阶段指定环境模型与软件模型之间的约束关系,而是由系统在在线过程中自动推理。

系统模型与系统之间需要进行即时的同步。系统的更新应当及时地在反映模型上进行体现,而反映模型被更新后,系统也应当被相应地更新。模型与系统间的同步可以通过模型-代码转换语言或者借助中间件来实现。模型与系统间的同步包括完全的和增量式的同步,其中,完全的同步是指从模型出发生成全部的应用代码,或从应用代码出发构建出全部的模型元素,这种方法在更新系统时需要停止运行和重新部署,不能支持高效的自适应;增量式的模型转换方法则会增量式地对模型或者系统进行更新,可以进行高效的更新并且不会中断系统的运行。自适应相关的工作通常选择的是增量的、在线的更新方式。文献[11]在进行重配置时,会计算出当前体系结构模型与目标体系结构模型的差别,再利用 OSGI 框架^[62]执行重配置的步骤,OSGI 框架可以在不重启系统的情况下安装、启动、重启和卸载构件。文献[19]在将模型的更新同步到系统时,首先由自适应机制自动生成从当前

的体系结构模型向目标体系结构模型作增量式转换的脚本,再由能够进行体系结构管理的中间件执行系统的重配置.文献[10,33]在进行自适应时,由中间件 XTEAM 对体系结构模型进行修改,同时将变化自动地反映到运行在中间件 Prism-MW 的系统上去.文献[30,39]采用了三元图语法规则(TGG rule)来实现模型与系统的同步.文献[18]在不同的模型层次之间建立起追踪关系,当抽象层次高的模型被选定之后,抽象层次低的模型和相应的代码就会由模型-模型和模型-代码的转换自动生成.文献[43]从结构模型出发,通过两次模型转换生成中间件可以解析执行的文档,再由中间件对运行于其上的系统进行重配置操作.文献[26]将每一个备选服务追踪到代码片段,并在代码片段上标注其对应的抽象功能.文献[42]在模型和系统之间增加了翻译层,该层中的翻译资源库保存了模型元素到系统元素之间的映射关系.

- 小结

本节介绍了模型与模型、模型与系统间的关系,对于基于模型的自适应来说,如何将模型层次上的自适应同步到运行时的系统中,是非常重要的研究课题.在进行模型到系统的同步时,可以通过模型-代码转换或者通过中间件直接把变化同步到系统中,也可以借助中间模型进行同步.例如,将功能性需求模型发生的变化同步到系统中时,通常会首先将变化同步到层次较低的结构模型或行为模型上,再由结构或行为模型将变化同步到系统.不同于功能性需求建模了系统能够提供的服务,非功能性需求建模了系统提供服务的质量.因此,非功能性需求不能直接与结构模型、系统等建立映射关系,一般地,功能性需求模型、结构模型、行为模型中的部分元素与非功能性需求之间存在正向或负向的贡献关系.

3.7 基于模型的自适应规划方法

在基于模型的自适应方法中,规划单位通常是模型中的元素,例如特征、目标、构件、服务等,或者是模型中的参数和在模型基础上定义的操作.这里的单位指的是进行自适应规划时的单位,不考虑之后带来的相应的改变.例如,文献[10]以特征为单位做出规划后,会通过模型间的同步对体系结构元素做出改变,这里的规划单位将被认为是特征.也有相关工作采用了多种自适应规划单位,例如,文献[19]可以以特征为基本单位在需求层次上进行规划,也可以以构件为单位在体系结构层次上进行规划;文献[35]可以以构件或者参数为单位进行规划.

如表 3 所示,常用的自适应规划方法包括基于规则的自适应、模型检查、控制论、基于效用函数的最优化、基于目标的推理、强化学习等.也有工作将多于一种的方法结合起来进行自适应规划.

在基于规则的方法中,自适应策略由一组规则进行预先定义,每条规则指定在一个特定的环境中应该执行某种特定的操作.自适应规则的形式通常包括“事件:条件 \rightarrow 操作”(ECA 规则)或“条件 \rightarrow 操作”(CA 规则)两种.相关工作^[11,14,15,17,18,20,22,30,42,47]对规则的“条件”和“操作”有不同的定义,例如,文献[14]的规则模型中,“条件”为环境特征,“操作”为软件特征的绑定;文献[15]的“条件”为一个基于环境信息的布尔表达式,“操作”为软件构件的插入或拔出;文献[47]的规则形式包括“前件 imply(推出)后件”和“前件 exclude(排除)后件”两种,其中,前件为环境表达式,后件为软件特征.采用基于规则的方法进行决策时,系统逐条判断各条自适应规则的触发条件是否为真:若为真,则执行这条规则指定的操作.若不同规则指定的操作存在冲突,则由系统根据指定的规则间的优先顺序做出选择,或者由利益相关者手动地在冲突的规则之间进行选择.自适应规则规约了自适应的逻辑,指定了在特定的环境变化发生时应当执行的自适应操作.该类方法的优点是简单和规划过程的高效,但缺点是因为策略被预先定义所以不够灵活,并且很可能有着过度庞大的状态空间或者规则集.

基于目标推理的方法通常会指定理想的状态,或者指定一个或多个刻画理想状态集合的标准,所有处于这个集合中的状态都是可以接受的,系统需要通过计算找到可从当前状态转移到理想状态的操作.基于目标推理的方法将自适应规划转化成了一个满足性问题或有约束的满足性问题.文献[21]采用了基于目标推理的方法,其理想状态是根目标被满足,所有使得根目标被满足的备选方案都是可行的.当在候选的方案中进行选择时,以该候选集对非功能性需求的贡献以及非功能性需求的优先级为依据进行选择.文献[23]采用了一个 PID 控制器来判断何时需要进行基于目标的推理,理想状态为关键目标被满足并且使得控制变量的“业务价值”在合法范围内的状态.文献[35]定义的理想状态为所有的非功能性需求被满足,即,所有的关键性能指标均处于合法的范围内.当有非功能性需求被违反,即,对应的关键性能指标偏离了合法范围时,系统会根据各自适应操作对关键

性能指标的影响,选择出能使关键性能指标恢复到合法范围的自适应操作集合.文献[20]结合了基于目标的推理方法和基于案例的推理方法来进行自适应规划:对于没有出现过的自适应场景,该工作采用基于目标推理的方法找到最优的自适应方案,同时会把场景与对应的方案作为一个案例保存下来;对于出现过的自适应场景,该工作将采用基于案例的推理方法,即,采用保存在案例中的该场景对应的自适应方案.

基于效用的方法将每个变体的理想程度映射为一个真值标量,在规划时,从当前可行的变体中选择出使效用函数值最大的变体.基于效用的方法将自适应规划转化成了一个最优化问题.部分相关工作将效用函数定义为系统非功能性需求(例如快速响应、高可用性、高安全性等)满足程度的加权和.在非功能性需求满意程度的衡量上,文献[10]为每个非功能性需求引入了可观测的属性作为其度量,能够基于度量的观测值,按预先定义的计算标准得到非功能性目标的满意度;文献[8]给每个候选的变体标注了在特定环境下的非功能性属性(例如,内存、响应时间)的预测函数,根据该函数和监测到的环境状态,可以计算出选择该变体时非功能性属性的预测值;文献[33]定义了根据系统配置估计非功能性需求满足度的分析模型,该分析模型是基于模糊数学和概率理论建立的,可以度量软件对非功能性需求影响的不确定性.文献[24,44]将目标模型映射到了动态决策网络上,非功能性需求的满意度被映射为效用节点,备选的自适应方案被映射为决策节点,备选方案对非功能性需求的贡献也被映射到网络中作为期望效用,从而将自适应规划转化成了基于效用函数和该复杂因果网络的最优化问题.

基于概率模型检查的自适应规划方法通常将软件的行为或环境建模为马尔可夫模型,例如离散时间马尔可夫链、连续时间马尔可夫链、马尔可夫决策过程等,同时,用概率计算树逻辑的形式表述目标.PRISM^[61]是一种为概率模型检查提供支持的工具,它支持各类马尔可夫链的建模,并能够支持概率计算树逻辑.大量基于马尔可夫链的自适应工作都采用了 PRISM 来进行概率模型检查.文献[25,28,31]采用离散时间马尔可夫链建模软件的行为,在运行过程中,自适应机制会通过概率模型检查来验证软件行为是否能够满足需求.在当前的软件行为不能满足需求的情况下,系统会重新选择其他备选方案以保证需求的持续满足.文献[26,32]用马尔可夫决策过程建模了软件的行为,在进行自适应规划时,自适应机制会在各备选的不确定转移之间进行选择,采用概率模型检查的方法选出使得非功能性需求被满足的概率最大的那个备选转移.文献[32]介绍了主动自适应的工作,它能根据对环境情况的预测,在变化发生前主动地做出自适应规划.文献[28]会根据运行时的监测结果对马尔可夫链上的概率进行自动的更新,然后在更新后的马尔可夫链上,用概率模型检查的方法得到使得到达目标状态概率最大的自适应方案.

(非概率的)模型检查的方法通常将软件的行为或环境建模为形式化的状态表达式,将目标用形式化的逻辑语言进行描述,在此基础上,通过模型检查验证需求的满足情况或者选择能最优化需求的备选方案.文献[27]将系统的行为建模为有限状态表达式,将目标用线性时序逻辑进行描述,通过模型检查的方法自动合成在不同领域假设情况下的自适应策略.文献[29]用时序逻辑语言描述了安全性需求,采用符号转移系统建模了环境和系统的演化模型,该演化模型对未来所有可能出现的状态进行了预测.基于演化模型和形式化的需求,该工作采用模型检查的方法判断每个状态是否有可能违反需求,并将有可能违反需求的状态标记为不合法的,接着找出可以避免进入或者离开不合法状态的自适应策略.

基于控制论^[63]的自适应规划方法根据控制模型做出自适应规划,该控制模型描述了如何根据观察值与目标值的偏差对控制参数进行调整.设计一个精确地用于自适应软件的控制模型非常困难,因为实时系统通常有着复杂的非线性的动态性.文献[34,36]通过在线学习自动地构建控制模型,可以避免设计该模型的复杂性问题.文献[34]通过在线学习得到分析模型,分析模型建模了控制参数与非功能性属性的关系,基于该模型得到控制策略,控制策略描述了如何根据属性值的偏差对控制参数的值进行调整.文献[36]采用的分析模型建模了控制信号与属性值的关系以及控制信号与软件配置的关系,基于第 1 个模型可以得到控制策略,即:如何根据当前属性值与目标属性之间的差异,选择控制信号;基于第 2 个模型得到软件配置,找到最佳软件配置被转化为了一个有约束的最优化问题.

基于强化学习的自适应规划方法在自适应场景与强化学习问题之间建立起映射关系,通过强化学习的算

法选择最优的自适应操作.将环境和软件状态映射为强化学习中的状态,将自适应操作映射为强化学习中的操作,将非功能性需求的满足度映射为强化学习中的奖励函数,通过强化学习可以得到一组使得奖励函数最大化的策略,其中,一条策略定义了一个状态与该状态下的最优操作的映射关系.文献[37,45,46]均采用了强化学习算法来进行自适应规划,其中,文献[46]采用了基于世界模型的强化学习,即:首先学习到环境模型,之后,基于该模型进行规划来获得最优策略.

- 小结

本节介绍了自适应规划方法.在选择适合的规划方法时,需要根据自适应软件的特点以及希望通过自适应达到的目的等进行考虑.若在设计阶段,开发者具备对于运行时的系统和环境的充足的知识,则可由利益相关者预先将自适应逻辑规约为自适应规则,采用基于规则的推理方式进行自适应规划.若自适应系统的动态性和不确定性较强,则不适合进行基于规则的规划,而更适合主动推理的规划方式.若自适应的目的是为了保证系统始终处于一个理想状态的集合中,则适合使用基于目标的推理方法;若自适应的目的是为了使某一表示系统状态理想程度的效用函数达到最优,则适合使用基于效用函数的最优化;若自适应的目的是为了将某一度量指标控制在理想值附近,则适合采用控制论的方法;若自适应的目的是为了保证某一属性的满足,且该属性可以被形式化并可被验证,则可采用模型检查或概率模型检查的方式进行规划;若自适应的目的是为了最优化即时奖励信号,可采用强化学习的方式进行规划.

4 总结与展望

本节对研究现状进行了分析,并在此基础上给出了未来的研究方向,主要关注 3 个方面的内容:建模方法研究、模型间的关系和自适应规划算法.

4.1 建模方法研究

对于基于模型的自适应方法来说,选择最恰当的建模方法对软件和环境进行建模是一个重要的研究课题.

在自适应软件系统的建模方面,相关工作采用了不同的模型,从某一方面上对软件的需求、软件的结构和行为、环境的状态和行为等进行了建模.但是,采用不同的模型对软件规约、环境和需求进行建模,不利于自适应系统进行基于模型的规划和验证.自适应软件需要通过自适应以保证 S (系统规约)、 E (环境)、 R (需求)被持续地满足,若能提出一个集成的建模框架,采用同一种模型从 3 个维度上对系统规约、环境、需求分别进行建模,则可为模型间关系的建立、基于模型的自适应系统的规划和验证过程提供更好的支持.

此外,现有的自适应系统工作对环境的建模较为简单.在环境的建模方面,相关工作大多数对环境的变化点和变体进行了简单的建模,对环境行为进行建模的工作较少,也缺乏对环境不确定性的显式表示.有少数的相关工作建模了环境的行为,例如文献[29]使用了符号转移系统,文献[32]使用了概率过程.但是文献[32]中的概率过程只考虑了一个单个的环境变量,文献[29]考虑的环境变化是确定的几个动作.

针对现有工作的问题,这里给出下面我们所建议的研究方向.

- 自适应系统的集成式建模框架.采用同一种模型(例如特征模型),从不同的维度上对系统规约、环境和需求进行建模.该集成式建模框架能够为基于模型的推理提供更好的支持;
- 环境行为和不确定性的建模.对环境的不确定性进行建模,有利于做出考虑了不确定性的更合理的自适应规划.对环境可能的变化进行预测,可以在环境变化之前进行主动的自适应规划.

4.2 模型间的关系研究

这里主要考虑自适应方案与非功能性需求之间的关系、软件模型与环境模型之间的关系以及软件模型与系统间的同步.

4.2.1 备选自适应方案与非功能性需求之间关系的研究

在基于模型的自适应方法中,非功能性需求或软目标的实现程度通常作为评价备选自适应方案的标准.因此,更好地规约或者学习备选方案对软件目标实现程度的贡献是一个重要的研究课题.

文献[21]等采用了定性的目标模型建模备选的自适应方案(叶子级别的功能性目标)对非功能性需求的贡献.定性的目标模型是一种广泛使用的目标模型,但当被用于自适应系统的在线推理中时,往往会存在粒度太粗、无法很好地区分备选方案的问题.

文献[19,23]用实数代替了“+”“-”这样的定性关系,但并未给出系统的量化方法,而是假设这样的数字可以在设计阶段根据经验直接给出.文献[8,19,35,41]在变体上标注了备选方案对非功能目标满足度的影响,但同样是假设领域专家会在设计阶段给出影响值,并且假设这样的影响值在运行阶段不会发生变化.

文献[10,33]等可以通过在线学习学到特征对非功能性需求的影响,能够基于收集到的运行时的特征值和非功能性需求度量的观测值,通过线性回归、MD5 模型树等方法学习到特征对非功能性需求满足程度的影响.学习到的影响值会在运行时根据环境的具体情况进行即时的更新.文献[33]的学习结果是控制参数对度量值的影响函数,文献[10]的学习结果是特征对度量值的影响函数,这个函数可以是以环境变量的取值作为分段参数的分段函数.这两个工作都没有学习环境对于非功能性需求的度量值的影响.

针对现有工作的问题,这里给出下面我们所建议的研究方向.

- 将经验与学习算法相结合,得到备选方案与非功能性需求的关系.在设计阶段,可以采用基于 AHP 的量化方法等,由软件专家对备选方案和非功能性需求之间的关系进行量化;在运行阶段,可以采用在线学习算法,根据运行环境对量化值进行更新;
- 在规约或者量化备选方案对非功能性需求的影响时,将环境属性考虑进来.非功能性需求的满足程度是系统与环境相互作用的结果,软件配置对非功能性需求满足程度的影响,在不同环境情况下会有较大的差别.

4.2.2 软件模型与环境模型之间关系的研究

现有的研究工作一般通过规则来规约环境模型与软件模型之间的约束,通常使用的规则形式为“事件:条件→操作”(ECA 规则)或“条件→操作”(CA 规则).这样的规则存在很多缺点,包括:

- 1) 规则不能随需求的变化而动态演化.这是因为 ECA 形式的规则缺乏与需求之间的关系,因此很难以需求为依据对其进行动态的定制;
- 2) 很难检测冲突的发生.这是因为 ECA 形式的规则缺乏良好的结构化,即,每一条规则都是全局的,从而导致无法高效地在整个规则库上进行冲突消解.

同时,对于高可配置系统或者环境频繁变化的系统来说,由人工定义自适应规则非常困难.这是因为:(1) 在环境状态和软件配置的数量都非常巨大的情况下,无法用一组较少的人工定义的规则集来覆盖它们之间复杂的对应关系;(2) 由于规则与需求之间缺乏联系,很难对定义出的规则集的质量进行验证;(3) 当规则中蕴含的预设知识与环境动态性所反映的真实知识不相符时,这些在设计时预先定义的规则就无法在运行时带来较好的自适应结果;(4) 当需求发生变化或者环境发生非稳态的变化时,旧的规则集就会失效.

针对现有工作的问题,这里给出下面我们所建议的研究方向.

- 提出一种新的规则,同时,对需求、软件规约和环境的关系进行规约.每条规则的触发条件不仅包括环境情况,还包括其能够维护的需求.当采用这样的规则集时,处于激活状态的规则子集会因需求设定的不同而不同;
- 提出能够从目标和环境出发,自主地生成自适应规则(即环境模型与软件模型间的约束关系)的算法.

4.2.3 软件模型与系统模型之间同步关系的研究

基于模型的自适应系统是目前自适应系统主流的研究方向.为了弥合模型与实现间的鸿沟,相关工作提出了一些支持的框架与中间件,它们可以根据系统的设计模型自动地生成代码.但是现有中间件支持的模型类型和同步类型仍然很有限,尤其是缺乏对以下两个问题的支持:(1) 如何根据设计模型,将现有的非自适应的传统系统再造为自适应系统;(2) 如何根据设计模型,将自适应系统与非自适应系统结合起来.因此,研究如何弥合自适应系统的模型与实现间的鸿沟、支持更安全的模型与系统间的同步方式、处理复杂场景下的模型与系统间的同步,是一个可能的未来研究方向.

4.3 自适应规划算法

自适应系统需要具备自主进行自适应规划的能力,即:当变化发生后,能够自主地选择一组自适应操作,以使得 S(软件配置)、E(环境状态)、R(用户需求)被持续地满足.自适应规划应当能够响应以下两种变化:(1) 环境的变化;(2) 需求的变化.其中,环境的变化可能是稳态的或非稳态的,稳态的变化是指环境状态的变化,非稳态的变化是指环境状态迁移概率的变化或者环境状态对需求影响的变化;需求的变化是指需求的增加或者减少以及各需求优先级的变化.

表 3 展示了现有的规划算法.这些规划算法可以划分为反射式的算法和主动推理的算法.反射式的规划算法通常依赖于预先规约好的自适应规则,具有规划过程高效、规则易读性强且易修改的优点.但是这类方法采用的自适应规则通常是在离线阶段预先定义的,并且缺乏与高层需求的联系.因此,反射式的规划算法只能响应稳态的环境变化,而无法响应需求的变化和非稳态的环境变化.相反地,采用主动推理算法的自适应系统会根据运行时的环境情况和决策标准推理出最适合的操作.因此,主动推理算法更有可能做出有效的自适应规划.但是在有多个需求、多个控制参数或多个备选方案的情况下,主动推理算法具有较高的复杂性.在备选方案空间庞大的情况下,基于效用函数的最优化、基于目标的推理、概率模型检查、控制模型的合成等规划算法都会面临低效性的问题,有可能无法及时响应运行时的变化.

综上,如何提高现有的动态规划算法、设计出结合反射式与主动推理自适应算法优点的新算法,是一个重要的研究点.理想的规划算法应当具备以下能力:(1) 有效,即,能够自主地选择自适应操作去响应环境和需求的变化,并且操作的执行能够保证需求被持续满足;(2) 高效,即,规划过程应当高效,能够及时地对运行时的变化做出响应.

5 结 论

本文对现有的基于模型的自适应方法的相关研究工作进行了综述.首先,提出了 6 个研究问题,关注相关工作如何进行建模、分析和规划等;然后,对现有的基于模型的自适应方法的相关研究工作进行了分析,对提出的 6 个研究问题进行了回答;最后,基于对现有方法的分析,指出现有研究中存在的问题和不足,提出了解决这些问题的设想,还给出了与基于模型的自适应方法相关的未来的研究方向.

致谢 在此,我们向对本文的工作给予支持和建议的老师和同学表示感谢.

References:

- [1] Cheng BH, Lemos R, Giese H, Inverardi P, Magee J, Andersson J, Becker B, Bencomo N, Brun Y, Cukic B, Serugendo GM, Dustdar S, Finkelstein A, Gacek C, Geijs K, Grassi V, Karsai G, Kienle HM, Kramer J, Litoiu M, Malek S, Mirandola R, Müller HA, Park S, Shaw M, Tichy M, Tivoli M, Weyns D, Whittle J. Software engineering for self-adaptive systems: A research roadmap. In: Proc. of the Software Engineering for Self-Adaptive Systems. Springer-Verlag, 2009. 1–26. [doi: 10.1007/978-3-642-02161-9_1]
- [2] de Lemos R, Giese H, Müller HA, Shaw M, Andersson J, Litoiu M, Schmerl B, Tamura G, Villegas NM, Vogel T, Weyns D, Baresi L, Becker B, Bencomo N, Brun Y. Software engineering for selfadaptive systems: A second research roadmap. In: Proc. of the Software Engineering for Self-Adaptive Systems II. Dagstuhl Castle: Int'l Seminar, 2010. [doi: 10.1007/978-3-642-35813-5_1]
- [3] Wang QX, Shen JR, Mei H An introduction to self-adaptive software. Computer Science, 2004,31(10):168–171 (in Chinese with English abstract). [doi: 10.3969/j.issn.1002-137X.2004.10.045]
- [4] Brun Y, Serugendo GDM, Gacek C, Giese H, Kienle H, Litoiu M, Müller H, Pezzè M, Shaw M. Engineering self-adaptive systems through feedback loops. In: Proc. of the Software Engineering for Self-Adaptive Systems. Berlin, Heidelberg: Springer-Verlag, 2009. 48–70. [doi: 10.1007/978-3-642-02161-9_3]
- [5] Feng YD, Huang G, Mei H. A method for modeling and realizing self-adaptive software architecture. Acta Scientiarum Naturalium Universitatis Pekinensis, 2008,44(1):67–76 (in Chinese with English abstract). [doi: 10.3321/j.issn:0479-8023.2008.01.014]
- [6] Zhao XP, Li MS, Wang Q, Chen ZC, Liang JN. An agent-based self-adaptive software process model. Ruan Jian Xue Bao/Journal of Software, 2004,15(3):348–359 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/20040304.htm>

- [7] Morin B, Barais O, Jezequel JM, Fleurey F, Solberg A. Models@ run. Time to support dynamic adaptation. *Computer*, 2009, 42(10).
- [8] Floch J, Hallsteinsen S, Stav E, Eliassen F, Lund K, Gjorven E. Using architecture models for runtime adaptability. *IEEE Software*, 2006,23(2):62–70. [doi: 10.1109/MS.2006.61]
- [9] Blair G, Bencomo N, France RB. Models@ run. Time. *Computer*, 2009,42(10). [doi: 10.1109/MC.2009.326]
- [10] Elkhodary A, Esfahani N, Malek S. FUSION: A framework for engineering self-tuning self-adaptive software systems. In: *Proc. of the 18th ACM SIGSOFT Int'l Symp. on Foundations of Software Engineering*. ACM Press, 2010. 7–16. [doi: 10.1145/1882291.1882296]
- [11] Cetina C, Giner P, Fons J, Pelechano V. Autonomic computing through reuse of variability models at runtime: The case of smart homes. *Computer*, 2009,42(10):37–43. [doi: 10.1109/MC.2009.309]
- [12] Swanson J, Cohen MB, Dwyer MB, Garvin B, Firestone J. Beyond the rainbow: Self-Adaptive failure avoidance in configurable systems. In: *Proc. of the 22nd ACM SIGSOFT Int'l Symp. on Foundations of Software Engineering*. ACM Press, 2014. 377–388. [doi: 10.1145/2635868.2635915]
- [13] Alferes GH, Pelechano V. Context-Aware autonomous Web services in software product lines. In: *Proc. of the 15th Int'l Software Product Line Conf. (SPLC)*. IEEE, 2011. 100–109. [doi: 10.1109/SPLC.2011.21]
- [14] Acher M, Collet P, Fleurey F, Lahire P, Moisan S, Rigault J. Modeling context and dynamic adaptations with feature models. In: *Proc. of the 4th Int'l Workshop Models@ run. Time at Models 2009 (MRT 2009)*. 2009. 10. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.204.5778>
- [15] Fleurey F, Delhen V, Bencomo N, Morin B, Jézéquel JM. Modeling and validating dynamic adaptation. In: *Proc. of the 3rd Int'l Workshop on Models@Runtime, at MoDELS 2008*. Toulouse, 2008. [doi: 10.1007/978-3-642-01648-6_11]
- [16] Ghezzi C, Sharifloo AM. Dealing with non-functional requirements for adaptive systems via dynamic software product-lines. In: *Proc. of the Software Engineering for Self-Adaptive Systems II*. Berlin, Heidelberg: Springer-Verlag, 2013. 191–213. [doi: 10.1007/978-3-642-35813-5_8]
- [17] Zhao TQ, Zan T, Zhao HY, Hu ZJ, Jin Z. Integrating Goal model into rule-based adaptation. In: *Proc. of the 23rd Asia-Pacific Software Engineering Conf.* 2016. 289–296. [doi: 10.1109/APSEC.2016.048]
- [18] Bencomo N, Sawyer P, Blair GS, Grace P. Dynamically adaptive systems are product lines too: Using model-driven techniques to capture dynamic variability of adaptive systems. In: *Proc. of the Int'l Conf. Software Product Lines, Vol.2*. Limerick, 2008. 23–32. https://link.springer.com/chapter/10.1007/978-3-642-35813-5_8
- [19] Chen B, Peng X, Yu Y, Nuseibeh B, Zhao W. Self-Adaptation through incremental generative model transformations at runtime. In: *Proc. of the 36th Int'l Conf. on Software Engineering*. ACM Press, 2014. 676–687. [doi: 10.1145/2568225.2568310]
- [20] Qian W, Peng X, Chen B, Mylopoulos J, Wang H, Zhao W. Rationalism with a dose of empiricism: Case-Based reasoning for requirements-driven self-adaptation. In: *Proc. of the 22nd IEEE Int'l Requirements Engineering Conf. (RE)*. IEEE, 2014. 113–122. [doi: 10.1109/RE.2014.6912253]
- [21] Lapouchnian A, Yu Y, Liaskos S, Mylopoulos J. Requirements-Driven design of autonomic application software. In: *Proc. of the 2006 Conf. of the Center for Advanced Studies on Collaborative Research*. IBM Corp., 2006. 7. [doi: 10.1145/1188966.1188976]
- [22] Cheng BHC, Sawyer P, Bencomo N, Whittle G. A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. In: *Proc. of the Int'l Conf. on Model Driven Engineering Languages and Systems*. Berlin, Heidelberg: Springer-Verlag, 2009. 468–483. [doi: 10.1007/978-3-642-04425-0_36]
- [23] Chen B, Peng X, Yu Y, Zhao W. Are your sites down? Requirements-Driven self-tuning for the survivability of Web systems. In: *Proc. of the 19th IEEE Int'l Requirements Engineering Conf. (RE)*. IEEE, 2011. 219–228. [doi: 10.1109/RE.2011.6051650]
- [24] Bencomo N, Belaggoun A. Supporting decision-making for self-adaptive systems: From goal models to dynamic decision networks. In: *Proc. of the REFSQ 2013*. 2013. 221–236. [doi: 10.1007/978-3-642-37422-7_16]
- [25] Calinescu R, Ghezzi C, Kwiatkowska M, Mirandola R. Self-Adaptive software needs quantitative verification at runtime. *Communications of the ACM*, 2012,55(9):69–77. [doi: 10.1145/2330667.2330686]
- [26] Ghezzi C, Pinto LS, Spoletini P, Tamburrelli G. Managing non-functional uncertainty via model-driven adaptivity. In: *Proc. of the 2013 Int'l Conf. on Software Engineering*. IEEE Press, 2013. 33–42. [doi: 10.1109/ICSE.2013.6606549]
- [27] D'Ippolito N, Braberman V, Kramer J, Magee J, Sykes D, Uchitel S. Hope for the best, prepare for the worst: Multi-Tier control for adaptive systems. In: *Proc. of the 36th Int'l Conf. on Software Engineering*. ACM Press, 2014. 688–699. [doi: 10.1145/2568225.2568264]

- [28] Sykes D, Corapi D, Magee J, Inoue K. Learning revised models for planning in adaptive systems. In: Proc. of the 35th Int'l Conf. on Software Engineering (ICSE). IEEE, 2013. 63–71. [doi: 10.1109/ICSE.2013.6606552]
- [29] Tsigkanos C, Pasquale L, Ghezzi C, Nuseibeh B. On the interplay between cyber and physical spaces for adaptive security. IEEE Trans. on Dependable and Secure Computing, 2016. [doi: 10.1109/TDSC.2016.2599880]
- [30] Vogel T, Giese H. A language for feedback loops in self-adaptive systems: Executable runtime mega models. In: Proc. of the 2012 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). IEEE, 2012. 129–138. [doi: 10.1109/SEAMS.2012.6224399]
- [31] Grassi V, Mirandola R, Randazzo E. Model-Driven assessment of QoS-aware self-adaptation. In: Proc. of the Software Engineering for Self-Adaptive Systems. Berlin, Heidelberg: Springer-Verlag, 2009. 201–222. [doi: 10.1007/978-3-642-02161-9_11]
- [32] Moreno GA, Cámara J, Garlan D, Schmerl B. Proactive self-adaptation under uncertainty: A probabilistic model checking approach. In: Proc. of the 10th Joint Meeting on Foundations of Software Engineering. ACM Press, 2015. 1–12. [doi: 10.1145/2786805.2786853]
- [33] Esfahani N, Kouroshfar E, Malek S. Taming uncertainty in self-adaptive software. In: Proc. of the 19th ACM SIGSOFT Symp. and the 13th European Conf. on Foundations of Software Engineering. ACM Press, 2011. 234–244. [doi: 10.1145/2025113.2025147]
- [34] Filieri A, Hoffmann H, Maggio M. Automated design of self-adaptive software with control-theoretical formal guarantees. In: Proc. of the 36th Int'l Conf. on Software Engineering. ACM Press, 2014. 299–310. [doi: 10.1145/2568225.2568272]
- [35] Rosa L, Rodrigues LET, Lopes A. Self-Management of distributed systems using high-level Goal policies. In: Proc. of the Software Engineering for Self-Adaptive Systems. 2010. 162–190. [doi: 10.1007/978-3-642-35813-5_7]
- [36] Filieri A, Hoffmann H, Maggio M. Automated multi-objective control for self-adaptive software design. In: Proc. of the 10th Joint Meeting on Foundations of Software Engineering. ACM Press, 2015. 13–24. [doi: 10.1145/2786805.2786833]
- [37] Kim DS, Park SY. Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software. In: Proc. of the SEAMS 2009. 2009. 76–85. [doi: 10.1109/SEAMS.2009.5069076]
- [38] Tajalli H, Garcia J, Edwards G, Medvidovic N. PLASMA: A plan-based layered architecture for software model-driven adaptation. In: Proc. of the IEEE/ACM Int'l Conf. on Automated Software Engineering. ACM Press, 2010. 467–476. [doi: 10.1145/1858996.1859092]
- [39] Pietschmann S. A model-driven development process and runtime platform for adaptive composite Web applications. Int'l Journal on Advances in Internet Technology, 2010,2:277–299.
- [40] Morin B, Fleurey F, Bencomo N, Jezequel J, Solberg A, Dehlen V, Blair G. An aspect-oriented and model-driven approach for managing dynamic variability. In: Proc. of the Int'l Conf. on Model Driven Engineering Languages and Systems. Berlin, Heidelberg: Springer-Verlag, 2008. 782–796. [doi: 10.1007/978-3-540-87875-9_54]
- [41] Rouvoy R, Barone P, Ding Y, Eliassen F, Hallsteinsen S, Lorenzo J, Mamelli A, Scholz U. Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments. In: Proc. of the Software Engineering for Self-Adaptive Systems. Berlin, Heidelberg: Springer-Verlag, 2009. 164–182. [doi: 10.1007/978-3-642-02161-9_9]
- [42] Garlan D, Cheng SW, Huang AC, Schmerl B, Steenkiste P. Rainbow: Architecture-Based self-adaptation with reusable infrastructure. Computer, 2004,37(10):46–54. [doi: 10.1109/MC.2004.175]
- [43] Geihs K, Reichle R, Wagner M, Khan MU. Modeling of context-aware self-adaptive applications in ubiquitous and service-oriented environments. In: Proc. of the Software Engineering for Self-Adaptive Systems. Berlin, Heidelberg: Springer-Verlag, 2009. 146–163. [doi: 10.1007/978-3-642-02161-9_8]
- [44] Salehie M, Pasquale L, Omoronyia I, Ali R, Nuseibeh B. Requirements-Driven adaptive security: Protecting variable assets at runtime. In: Proc. of the 20th IEEE Int'l Requirements Engineering Conf. (RE). IEEE, 2012. 111–120. [doi: 10.1109/RE.2012.6345794]
- [45] Amoui M, Salehie M, Mirarab S, Tahvildari L. Adaptive action selection in autonomic software using reinforcement learning. In: Proc. of the 4th Int'l Conf. on Autonomic and Autonomous Systems. 2008. [doi: 10.1109/ICAS.2008.35]
- [46] Ho HN, Lee E. Model-Based reinforcement learning approach for planning in self-adaptive software system. In: Proc. of the 9th Int'l Conf. on Ubiquitous Information Management and Communication (IMCOM 2015). 2015. [doi: 10.1145/2701126.2701191]
- [47] Fernandes P, Werner C, Murta LGP. Feature modeling for context-aware software product lines. In: Proc. of the 20th Int'l Conf. on Software Engineering & Knowledge Engineering (SEKE 2008). World Scientific, 2008. 758–763. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.525.1191>
- [48] Mylopoulos J, Chung L, Nixon B. Representing and using non-functional requirements: A process-oriented approach. IEEE Trans. on Software Engineering, 1992,18(6). [doi: 10.1109/32.142871]

- [49] Chung L, Nixon B, Yu E, Mylopoulos J. Non-Functional Requirements in Software Engineering. Kluwer, 2000. <http://www.springer.com/us/book/9780792386667>
- [50] Mylopoulos J, Chung L, Nixon B. On non-functional requirements in software engineering. In: Proc. of the Conceptual Modeling: Foundations and Applications. Berlin, Heidelberg: Springer-Verlag, 2009. 363–379. [doi: 10.1007/978-3-642-02463-4_19]
- [51] Darimont R, Delor E, Massonet P, Lamsweerde AV. GRAIL/KAOS: An environment for Goal-driven requirements engineering. In: Proc. of the 19th Int'l Conf. on Software Engineering. ACM Press, 1997. 612–613. [doi: 10.1145/253228.253499]
- [52] Whittle J, Sawyer P, Bencomo N, Cheng BHC, Bruel JM. Relax: Incorporating uncertainty into the specification of self-adaptive systems. In: Proc. of the 17th IEEE Int'l Requirements Engineering Conf. 2009. (RE 2009). IEEE, 2009. 79–88. [doi: 10.1109/RE.2009.36]
- [53] Vardi MY. An automata-theoretic approach to linear temporal logic. In: Proc. of the Logics for Concurrency. Berlin, Heidelberg: Springer-Verlag, 1996. 238–266. [doi: 10.1007/3-540-60915-6_6]
- [54] Van Lamsweerde A. Goal-Oriented requirements engineering: A guided tour. In: Proc. of the 5th IEEE Int'l Symp. on Requirements Engineering. IEEE, 2001. 249–262. [doi: 10.1109/ISRE.2001.948567]
- [55] Batory D. Feature models, grammars, and propositional formulas. In: Proc. of the Int'l Conf. on Software Product Lines. Berlin, Heidelberg: Springer-Verlag, 2005. 7–20. [doi: 10.1007/11554844_3]
- [56] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. Feature oriented design analysis (FODA) feasibility study. Technical Report, CMU/SEI-90-TR-21-ESD-90/TR-222, Carnegie Mellon University, 1990.
- [57] Roos-Frantz F, Benavides D, Ruiz-Cortés A, Heuer A, Lauenroth K. Quality-Aware analysis in product line engineering with the orthogonal variability model. Software Quality Journal, 2012,20(3-4):519–565. [doi: 10.1007/s11219-011-9156-5]
- [58] Clements P, Bachmann F, Bass L, Garlan D, Ivers J, Little R, Merson P, Nord R, Stafford J. Documenting Software Architectures: Views and Beyond. Addison-Wesley, 2003. <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=30386>
- [59] Rumbaugh J, Jacobson I, Booch G. Unified Modeling Language Reference Manual. Pearson Higher Education, 2004. <https://dl.acm.org/citation.cfm?id=993859>
- [60] Kemeny JG, Snell JL. Finite Markov Chains. Princeton: van Nostrand, 1960.
- [61] Kwiatkowska M, Norman G, Parker D. Prism: Probabilistic model checking for performance and reliability analysis. ACM Performance Evaluation Review, 2009,36(4):40–45. [doi: 10.1145/1530873.1530882]
- [62] Gu T, Pung HK, Zhang DQ. Toward an OSGi-based infrastructure for context-aware applications. IEEE Pervasive Computing, 2004,3(4):66–74. [doi: 10.1109/MPRV.2004.19]
- [63] Lee EB, Markus L. Foundations of optimal control theory. Journal of the Royal Statistical Society, 1967, 589. [doi: 10.2307/2343766]

附中文参考文献:

- [3] 王千祥,申峻嵘,梅宏.自适应软件初探.计算机科学,2004,31(10):168–171. [doi: 10.3969/j.issn.1002-137X.2004.10.045]
- [5] 冯耀东,黄昱,梅宏.一种自适应软件体系结构建模及其实施方法.北京大学学报(自然科学版),2008,44(1):67–76. [doi: 10.3321/j.issn:0479-8023.2008.01.014]
- [6] 赵欣培,李明树,王青,陈振冲,梁金能.一种基于 Agent 的自适应软件过程模型.软件学报,2004,15(3):348–359. <http://www.jos.org.cn/1000-9825/20040304.htm>



赵天琪(1990—),女,山西运城人,博士,主要研究领域为需求工程,自适应系统,强化学习.



张伟(1978—),男,博士,副教授,主要研究领域为群体化软件开发方法,软件需求工程.



赵海燕(1966—),女,博士,副教授,CCF 高级会员,主要研究领域为需求工程,软件复用,程序语言.



金芝(1962—),女,博士,教授,博士生导师,CCF 会士,主要研究领域为需求工程,知识工程.