



AI+ 研发数字峰会
AI+ Development Digital summit



需求分析流程自动化： 从NLP到软件规格说明

李智 | 教育区块链与智能技术教育部重点实验室
广西师范大学

科技生态圈峰会 + 深度研习



—1000+ 技术团队的共同选择



 **K+峰会**  **敦煌站**

K+ 思考周®研习社

时间: 2025.08.29-30

 **K+峰会**  **上海站**

K+ 金融专场

时间: 2025.10.17-18

 **K+峰会**  **香港站**

K+ 思考周®研习社

时间: 2025.11.25-26



K+峰会详情



 **AiDD峰会**  **上海站**

AI+研发数字峰会

时间: 2025.05.17-18

 **AiDD峰会**  **北京站**

AI+研发数字峰会

时间: 2025.08.08-09

 **AiDD峰会**  **深圳站**

AI+研发数字峰会

时间: 2025.11.28-29



AiDD峰会详情



李智

博士/教授/博导,广西师范大学独秀学者

中国计算机学会(CCF)杰出会员、软件工程专委会首届常务委员、系统软件、形式化方法及服务计算专委会委员,IEEE/ACM高级会员,担任SCI期刊《专家系统:知识工程》副主编,主要研究方向包括人机物融合系统建模、非功能需求抽取、生成、验证及工具支持、智能化需求工程及人机交互等,获得国家自然科学基金项目3项、省部级项目6项,发表学术论文70余篇(含CCF推荐A类期刊论文5篇,3次获得国际学术会议最佳论文奖),做学术会议邀请报告20余次(含大会特邀主旨报告2次、中国计算机大会CNCC技术论坛主席及讲者2次)。

目录

CONTENTS

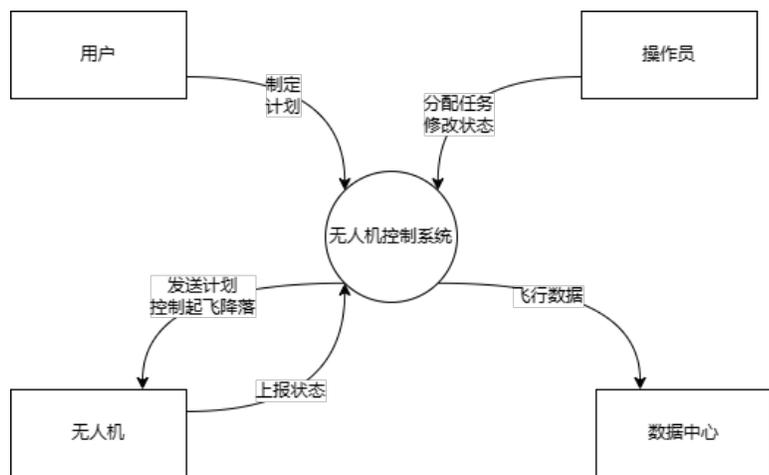
1. 背景
2. 需求文档预处理
3. 信息抽取及需求分类
4. 需求文档自动生成问题图
5. 需求模型至软件规格的转换

PART 01

背景

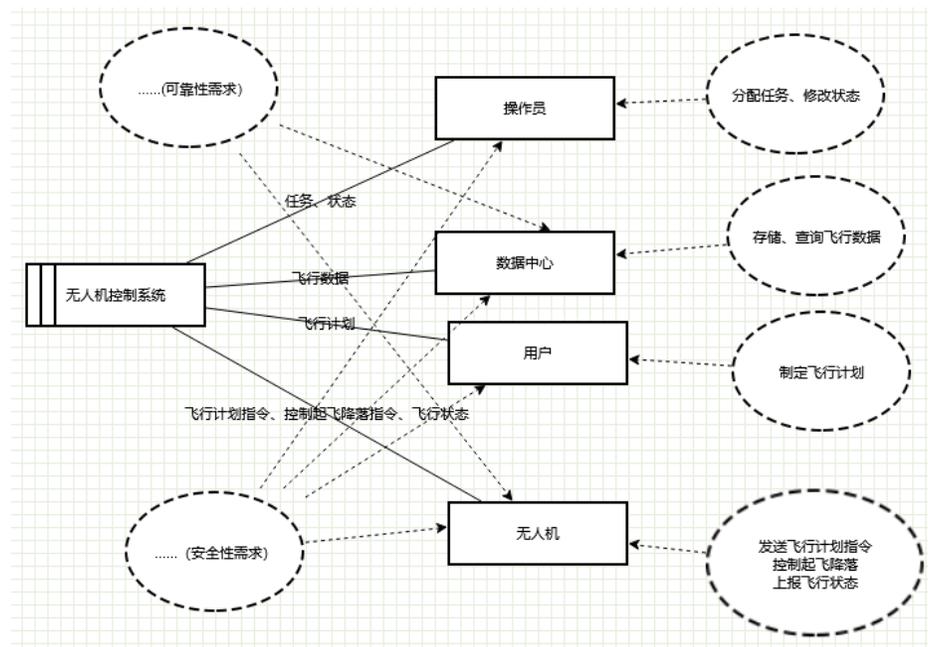
▶ 研究背景-问题框架

问题框架(PF, Problem Frames)方法是由软件工程著名学者Michael Jackson提出的一种结构化需求分析方法。PF强调需求最初来源是**问题所有者**(problem owner), **是否真正满足**(例如项目验收)必须在**现实世界场景中进行观察及测试**, 而不能仅观察和测试软件与外部环境的接口进行项目验收。因此, 需求工程核心任务是对软件上下文(现实世界)进行刻画**描述和建模**。该方法是在**上下文图**(context diagram)的基础上增加和定位**用户需求**的描述, 成为**问题图**。



无人机控制系统上下文图

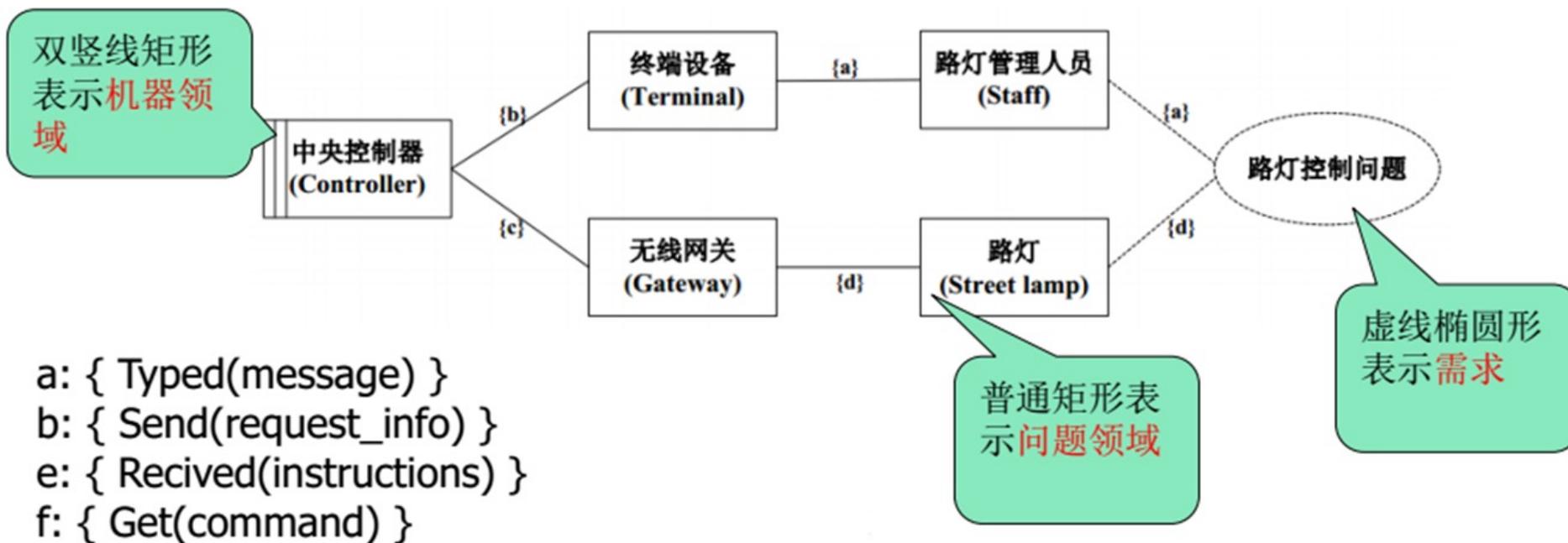
增加和定位需求



无人机控制问题图

▶ 概念介绍-问题框架理论

问题框架 (Problem Frames) 是一种用于需求分析和系统设计的方法论。它提供了一种结构化的方式来描述和分析复杂系统中的问题和需求。下面通过一个经典问题图了解问题框架如何描述信息物理融合系统的需求。



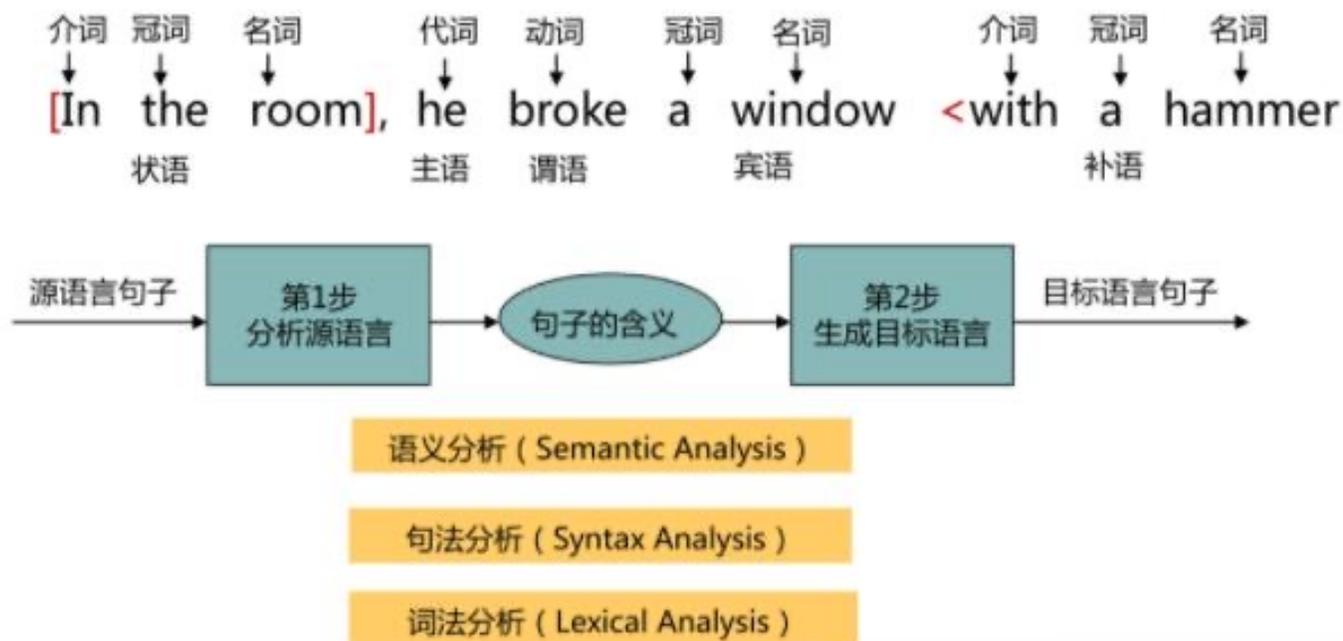
PART 02

需求文档预处理及信息抽取

▶ 需求文档预处理-分词

分词是中文文本处理的首要步骤，它将连续的字序列按照一定的规范重新组合成词序列的过程，例如，句子“我爱北京天安门”可以分词为“我/爱/北京/天安门”。

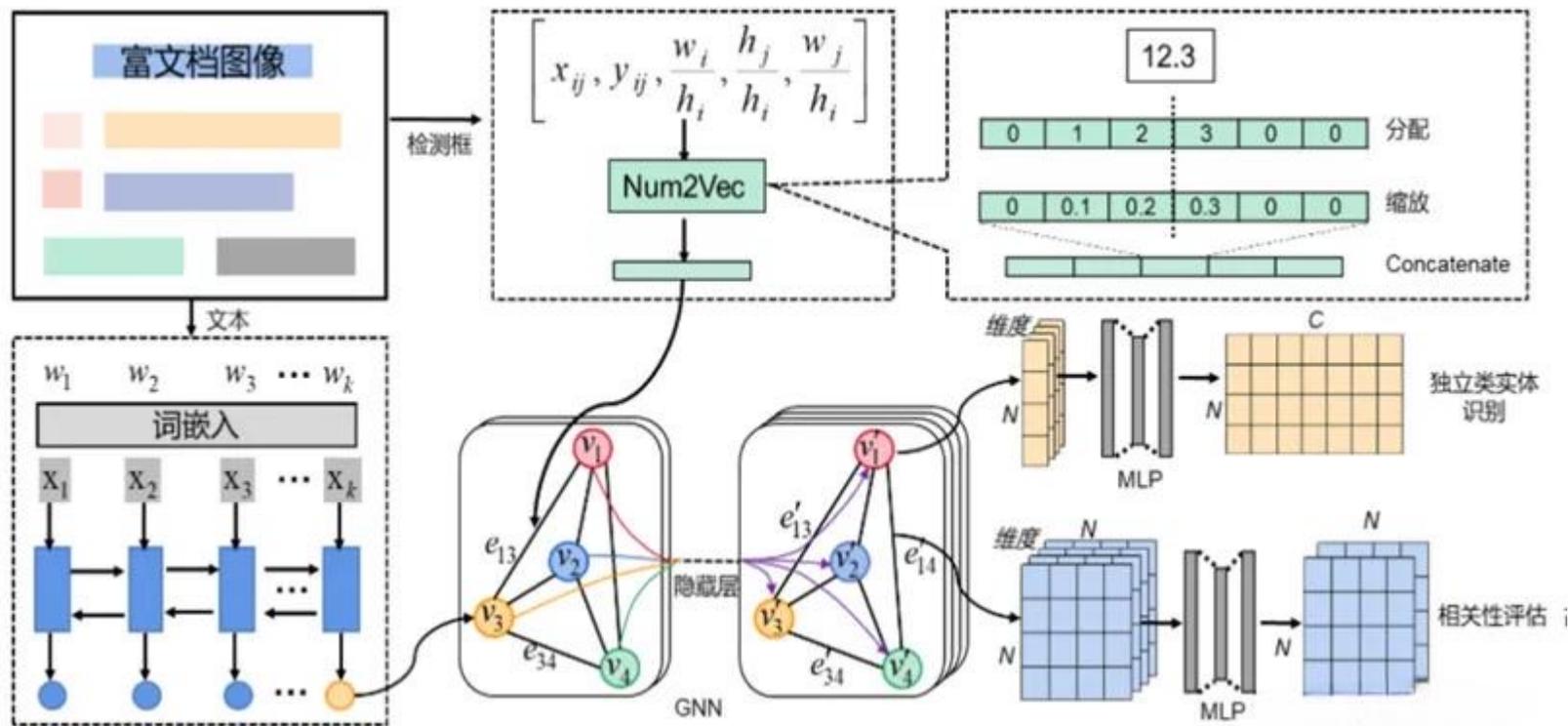
分词方法包括基于规则的方法、基于统计的方法和基于机器学习的方法。基于规则的方法依赖于词典和一定的规则，而基于统计的方法则通过分析大量文本数据来学习词汇之间的关联性。



▶ 需求文档预处理-去除停用词

停用词是指在文本处理中被认为没有实际意义的词汇，如“的”、“是”等。去除这些词可以减少后续处理的复杂度。

去除停用词可以通过编写正则表达式或者使用现成的停用词库来实现。



▶ 需求文档预处理-词干提取与词形还原

词干提取 (Stemming) 是去除单词的**后缀**，还原到基本形式的过程，如 “running”变为 “run”。

词形还原 (Lemmatization) 则是将单词还原到词典中的形式，它比词干提取更精细，因为它考虑了词的词源。

```
see -> see
saw -> saw
cat -> cat
cats -> cat
stem -> stem
stemming -> stem
lemma -> lemma
lemmatization -> lemmat
known -> known
knowing -> know
time -> time
timing -> time
football -> footbal
footballers -> footbal
```

▶ 需求文档预处理-词性标注

词性标注是为文本中的每个词标注词性，如**名词**、**动词**、**形容词**等。

词性标注有助于理解**句子结构和语义**，对于后续的文本分析和信息抽取非常重要。

Tag	Description	Example	Tag	Description	Example
a	adjective	美丽	ni	organization name	保险公司
b	other noun-modifier	大型, 西式	nl	location noun	城郊
c	conjunction	和, 虽然	ns	geographical name	北京
d	adverb	很	nt	temporal noun	近日, 明代
e	exclamation	哎	nz	other proper noun	诺贝尔奖
g	morpheme	茨, 甥	o	onomatopoeia	哗啦
h	prefix	阿, 伪	p	preposition	在, 把
i	idiom	百花齐放	q	quantity	个
j	abbreviation	公检法	r	pronoun	我们
k	suffix	界, 率	u	auxiliary	的, 地
m	number	一, 第一	v	verb	跑, 学习
n	general noun	苹果	wp	punctuation	, . !
nd	direction noun	右侧	ws	foreign words	CPU
nh	person name	杜甫, 汤姆	x	non-lexeme	萄, 翱

▶ 信息抽取-实体识别

实体是指文本中具有特定意义的名词或名词短语，如人名、地名、组织名等。

可以利用正则表达式或预定义的词典库进行实体识别。

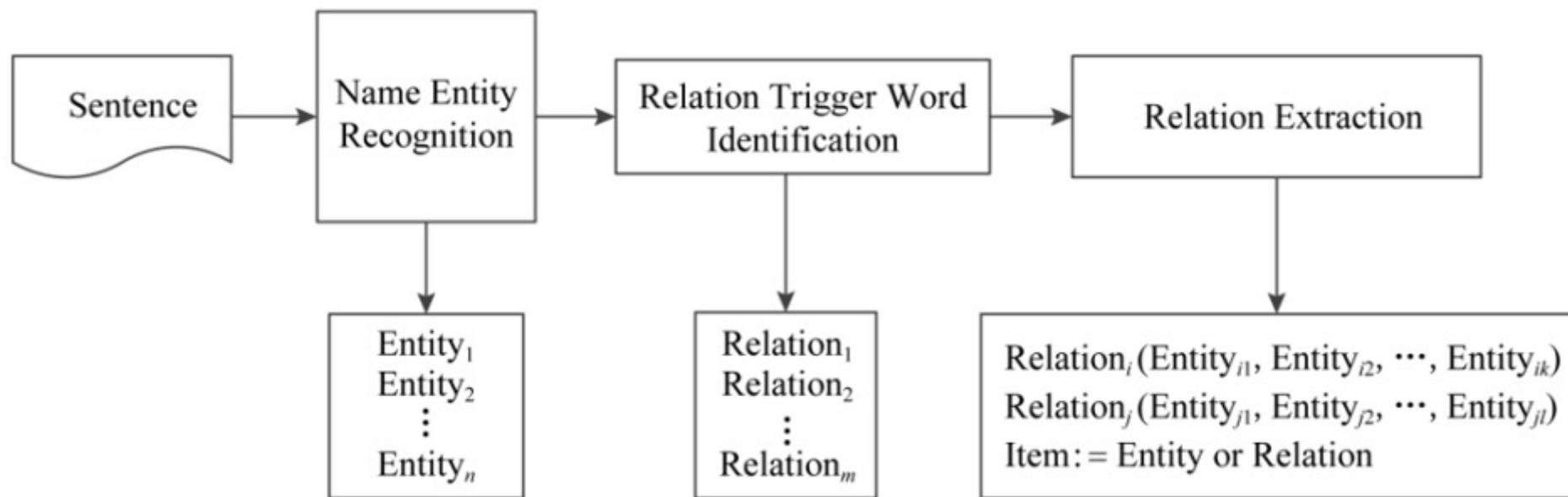
Person p Loc l Org o Event e Date d Other z

Barack Hussein Obama II (born August 4, 1961) is an American attorney and politician who served as the 44th President of the United States from January 20, 2009, to January 20, 2017. A member of the Democratic Party, he was the first African American to serve as president. He was previously a United States Senator from Illinois and a member of the Illinois State Senate.

▶ 信息抽取-关系抽取

关系抽取是为了识别文本中实体对之间的**语义关系**，如“位于”、“属于”等。

抽取方法包括**基于特征的方法**、**基于深度学习的方法**和**远程监督方法**。



关系抽取系统框架

▶ 信息抽取-事件抽取

事件是指文本中描述的具有**特定意义的动作或发生的情况**，如“地震”、“会议”。

事件抽取包括**事件触发词识别**、**事件类型分类**和**事件论元抽取**。

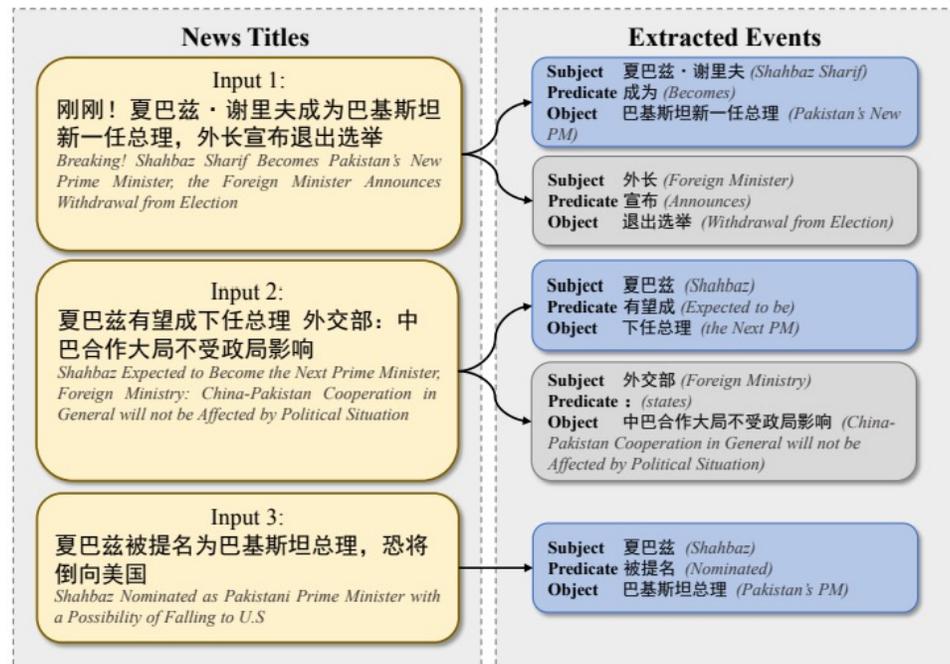


Figure 1: An example of event extraction on news titles where all factual events are extracted. Similar events are identified (in blue color) and could be used in aggregating relevant news.

PART 03

需求分类及提取

需求提取建模：在需求分析自动化中，首先利用NLP技术从需求文档中**自动提取功能**和**非功能需求**，识别**关键实体和关系**，生成初步的需求模型。以此为基础，才能通过问题框架（Problem Frames）方法对需求进行结构化建模，系统分析问题域及其上下文，**确保需求的完整性和准确性**。

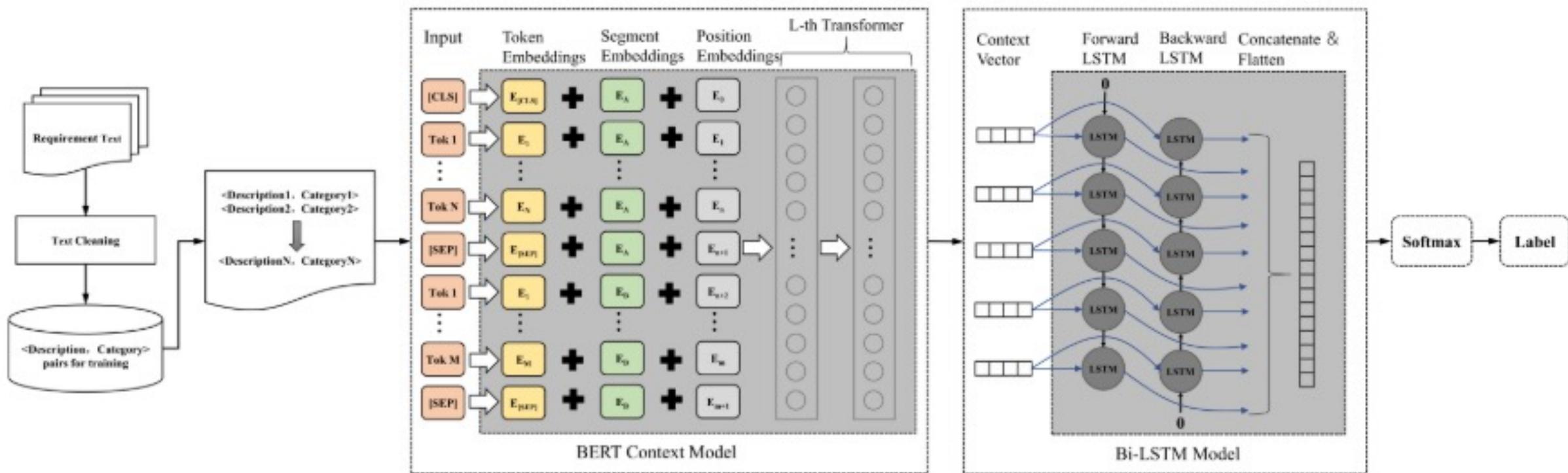
非功能需求：软件为满足业务需求而必须具备的那些**品质**（又称**质量需求**）。这些非功能性需求的引出需要专业知识、经验和领域知识，这对需求工程师和开发人员来说是具有挑战性和耗时的。如果能够从需求文档中自动提取非功能性需求，将是非常有益的，以减少人力、时间和精力上的疲劳。

数据分布长尾问题：虽然机器学习在需求工程研究与应用取得了一些令人满意的结果，但其中**许多方法**只是在**小规模**和**相对平衡的数据集**上进行评估，**无法反映它们在实际应用中的典型特征**。随着需求文档数量的增加和新类型的NFR的出现，NFR逐渐被细分为更细的类别，新类型往往代表用户最迫切的需求。因此，重要的是不仅要准确地对大量样本的NFR进行分类（即**头部类别**）而且还要处理具有较少样本的类别（即**尾部类别**）

► 研究内容——NFRNet

论文提出了一种深度神经网络模型NFRNet，由两部分组成：

1. 基于N-gram掩蔽的BERT单词嵌入模型，用于学习需求描述的上下文表示。
2. 双向长短期记忆网络Bi-LSTM,用于提取需求描述中包含的NFR。
3. 采用多采样Dropout技术，它在每个训练过程中选择多个丢弃样本进行隐藏，对所有样本的损失进



NFRNet整体架构图

► 实验结果

基于软件需求分类研究中广泛使用的数据集PROMISE。
将PROMISE数据集从原来的11类非功能需求类别扩展到32类，从原来的255个非功能需求描述语句扩展到6222个语句。

我们在SOFTWARE NFR数据集进行了10折交叉验证，以评估所提出模型的分
类准确性。从表中可以看出，我们设计的网络在所有模型中达到了最高的
分类准确率91.00%

Model \ Metrics	DPCNN	FastText	TextCNN	TextRCNN	TextRNN	TextRNN-Att	Transformer	LSTM	BERT	BERT-CNN	BERT-DPCNN	BERT-RCNN	BERT-RNN	NFRNet
Precision	0.8406	0.8611	0.8590	0.8677	0.7991	0.8193	0.8209	0.8299	0.8571	0.8873	0.8708	0.9056	0.8572	0.9100
Recall	0.8267	0.8554	0.8514	0.8634	0.7931	0.8131	0.8155	0.8171	0.8506	0.8858	0.8674	0.9034	0.8522	0.9265
F1-score	0.8276	0.8556	0.8520	0.8632	0.7934	0.8130	0.8161	0.8164	0.8508	0.8851	0.8668	0.9034	0.8520	0.9147

表：基准模型和我们提出的NFRNet模型进行NFR分类的实验结果

▶ 研究内容——NFRNet-LT

针对非功能性需求分类粒度大、数据长尾分布不均衡的特点，基于NFRNet，提出了一种新的深度神经网络模型NFRNet-LT，以解决从不同类型的文档中提取非功能性需求的难题。主要有以下改进

- 1.为了捕捉不同的语言特征并丰富语言信息，我们将滑动窗口N-gram机制引入TextCNN模型。这允许整个TextCNN模型考虑多个连续单词的语言特征，从而改善文本结构信息的捕获。
- 2.为了解决长尾分布问题，我们将Focal loss改进为适用于多个类别的Multi-Focal loss。这涉及动态计算和更新每个类的不同权重。此外，我们还引入了指数移动平均机制（EMA）来平滑权重更新并防止梯度爆炸，构成了我们的EMLFocal损失函数。

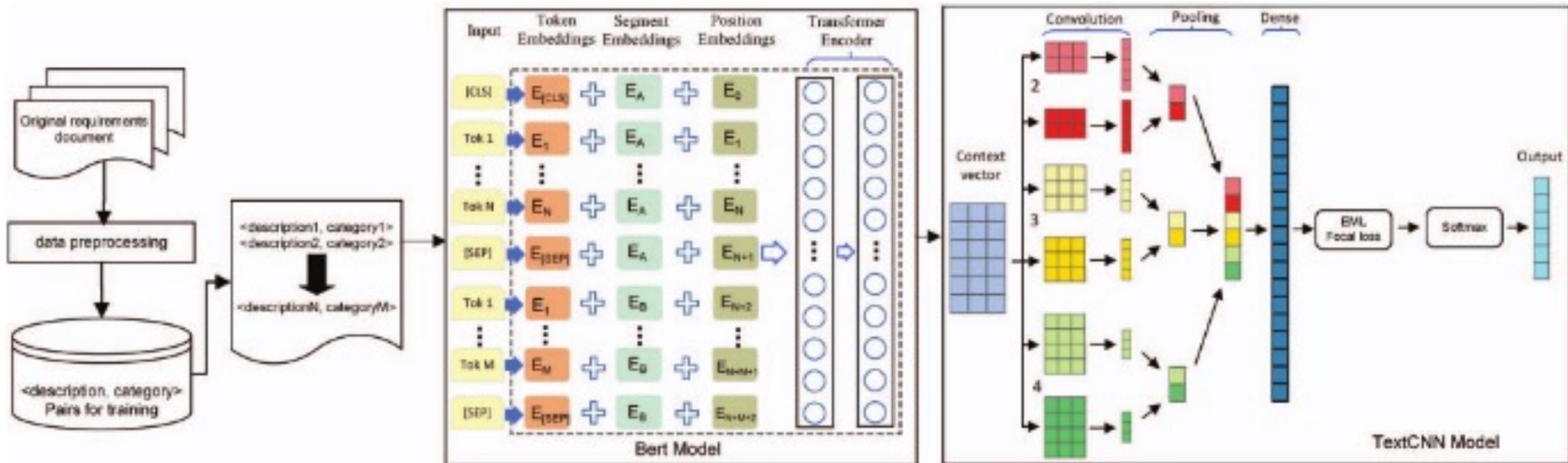


Fig. 1. Architecture of NFRNet-LT

▶ 研究内容——数据集及模型评价

我们收集了广泛用于软件需求分类研究的**Promise** NFR数据集，还包括与**电子健康记录**（EHR）相关的11个系列文档，以及**华为云系统文档**。我们将该集合过滤为一个命名的LT-NFR数据集，其中我们将原始Promise数据集中的11个NFR类别扩展为37个类别和19, 311个来自维基百科原始文档描述性语句的语句。

为了评估所提出的模型分类精度，我们对整个数据集进行了10倍交叉验证。我们使用宏F1得分（Macro F1），Top-1准确率和Top-5准确率作为模型的评价指标。

► 研究内容——实验结果

从表中可以看出，我们设计的神经网络模型达到了78.58%的最高宏F1。我们的方法在Top-1准确度和Top-5准确度方面明显优于其他机器学习方法。这表明我们的方法能够更好地关注和具有长尾分布数据集的尾类别的多样性和代表性特征。

Model \ Metrics	SVM	TextRNN	FastText	BiLSTM	TextCNN	Transformer	RCNN	BERT	BERT-BiLSTM	NFRNet-LT
Macro F1(%)	55.42	69.59	71.04	71.22	73.34	72.41	75.41	75.56	75.83	78.58
Top-1(%)	53.11	67.88	67.91	69.69	69.81	70.09	73.92	74.02	74.41	77.26
Top-5(%)	75.31	88.61	87.41	89.81	90.46	91.37	93.07	92.76	94.11	96.16

不同基线模型和我们提出的NFRNet-LT模型在NFR分类实验中的结果

J. Deng, Z. Li, X. Zhou and H. Xiao, "NFRNet-LT: Improving Accuracy in Extracting Long-tailed Non-functional Requirements," 2023 IEEE 31st International Requirements Engineering Conference (RE), Hannover, Germany, 2023, pp. 355-356, doi: 10.1109/RE57278.2023.00049.

PART 04

需求文档自动生成问题图

问题	问题描述	自动生成需求模型的作用
面向问题的需求建模研究较少	目前NLP4RE的研究工作大多都集中与需求提取、需求分析（分类、标签化、追踪）、质量评估，对于 需求概念模型自动建模研究较少 。	更好的分析软件问题及其作用的环境上下文。
开源的需求自动建模的工具较少	目前能使用的需求半自动 辅助建模工具较少 。	通过自动化工具能提高效率，问题图也更能清晰表示需求之前的关系，有助于更好理解和管理需求。

▶ 研究内容——需求文档自动生成问题图

问题图是问题框架的一种可视化形式，它以领域和现象的形式来描述需求。

使用nlp技术可以提取需求文档中的**关键短语和关键关系**，通过设计基于问题框架的**元模型的语义规则**，可以将短语**映射到问题图**的节点，从而实现问题图的构建。

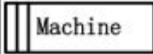
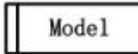
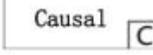
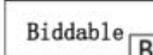
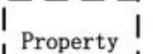
Name Of Nodes	Graphical	Meta-Model Element	Descriptions
Machine		MachineType	The machine is where the software to be run.
Designed Domain		LexicalType	Lexical domains provide physical space where the data is stored, thus with some causality in their storage behaviors.
Causal Domian		CausalType	Causal domains contain predictability in their behaviors
Biddable Domian		BiddableType	biddable domains (usually human beings) do not contain predictability in their behaviors.
Domain Property		Property	They describe internal or external causal relationships or other characteristics of the domain.
Requirement		Requirement	They describe the behavioral references or constraints, prescribed by the problem owners.

Fig. 2. How meta-model elements are mapped PF Nodes

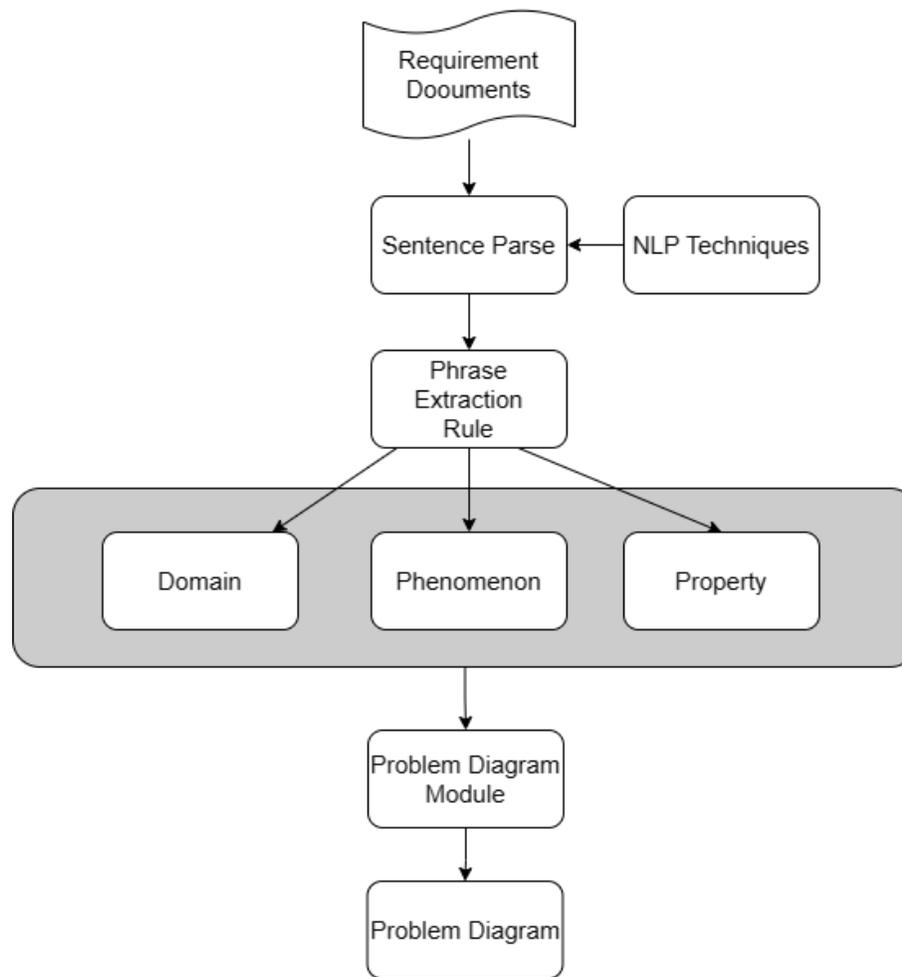
Xiao H, Li Z, Yang Y, et al. An Extended Meta-Model of Problem Frames for Enriching Environmental Descriptions[C]//2021 IEEE 29th International Requirements Engineering Conference Workshops (REW). IEEE, 2021: 428-434.

▶ 研究内容——需求文档自动生成问题图

主要研究内容：

1. 提取需求文档中的**关键短语和关键关系**。

2. 定义**转换规则**, 将短语转换为问题图。



生成问题图技术路线

研究内容——总体框架

(1) 需求文档预处理:

包括**共指消解和标点符号**处理, 以**降低句法分析的不确定性**, 确保信息提取的准确性。

(2) 信息提取:

从需求文本中抽取结构化信息。

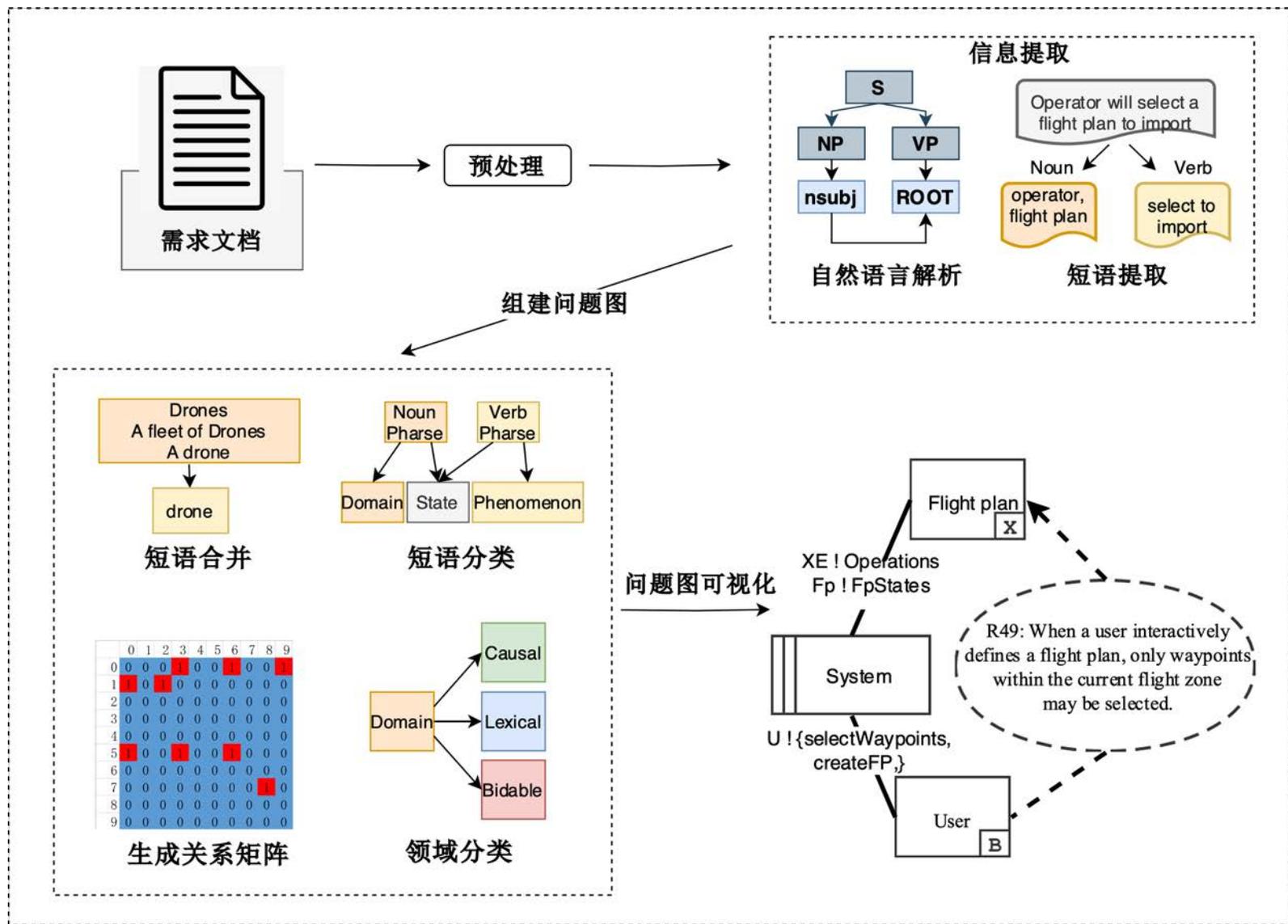
通过**启发式规则提取关系对**。先将并将其表示为〈实体, 关系, 实体〉三元组。

(3) 基于问题框架语义的元素转换:

将提取的三元组转换成问题图中的元素。根据**问题框架语义**和WordNet知识库进行**领域划分**, 填充问题图细节。

(4) 基于问题图元模型的可视化:

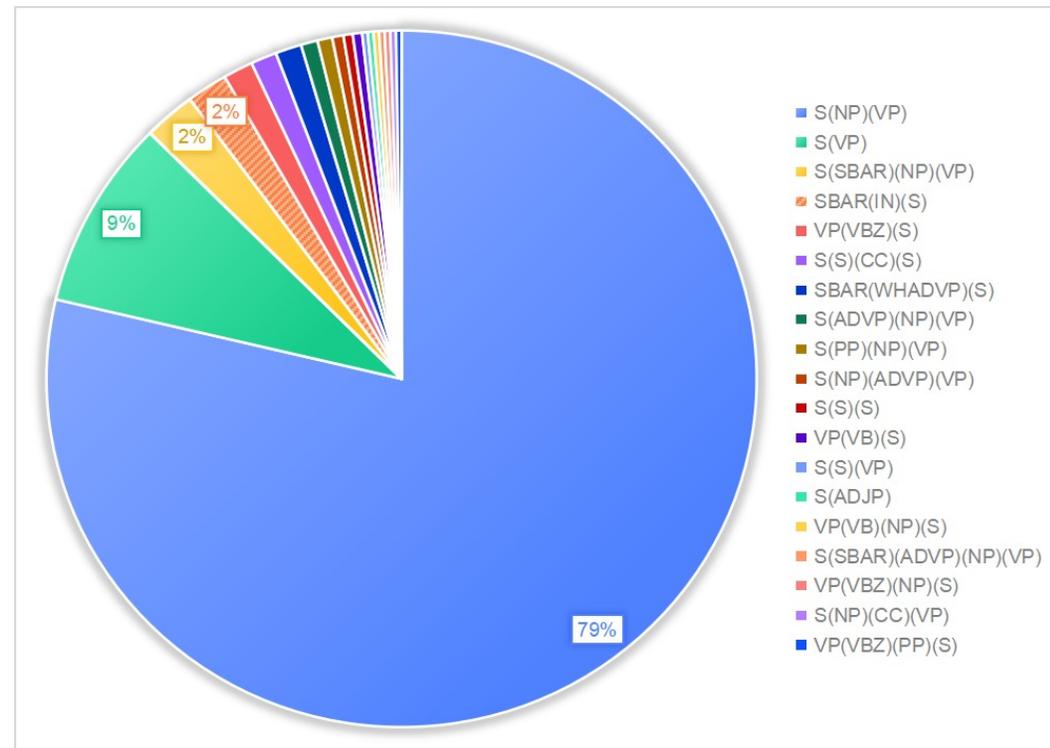
将识别的问题图元素转换成json文件, 并使用NL2PD平台进行**可视化**, 支持高效的问题建模和分析。



▶ 关键短语提取

首先进行句子拆分将复杂需求语句拆分为**简单句**，然后针对单个简单句分别进行**NP提取**，**VP提取**。

Rule ID	描述	例子
S1	将所有存在VP子节点的S节点加入候选简单句列表。	S(VP)
S2	候选简单句的直接父节点为ROOT/SBAR/S节点，则确认为简单句	ROOT(S);SBAR(S);S(S);
S3	候选简单句的直接父节点为S或者SBAR节点，则确认为简单句	VP(S(VP)) replace subVP to S



需求语句中S节点分布

▶ 关键短语提取

Rule ID	描述	例子
R1	对于仅有一个连续名词的动词短语，确定为实体。	NNS,NN
R2	对于存在多个连续名词的名词短语，使用相似度分析确定实际名词。	similarity (NP Phase , [NNS,NNS])
R3	将动词短语VP直系动作短语中每个VB_加入动词组，作为动词候选	[VB_,VBN,VB.....]
R4	动词组中当前动词是系动词或助动词，将后续第一个动词确认为主要动词，如果主要动词为被动语态，则将被动状态设为True	is,have,does,are,... ...
R5	不存在VP节点后对NP, PP进行实体提取，如果存在介词to/for则将实际实体从介词短语中提取。	transmit gps to center command
R6	如果不存在宾语，或者动词候选仅有系动词或助动词，则设置动作类型为state，如果存在有效动词和宾语则设置为event。	is a feature

以无人机需求为案例提取结果

ID	Requirement
R1	A fleet of drones will be generated at simulation startup.
R2	The system will either be in physical mode or virtual mode but never simultaneous.
R3	When the system is in physical mode only physical drones will be used.
R4	When the system is in virtual mode only virtual drones will be used.
R5	The system will not switch modes at runtime.
R6	Flight plans will be specified in XML.
R34	Operator will select a flight plan to import.
R7	Each flight plan will include a set of waypoints specified in terms of coordinates.
R8	Each flight plan will be assigned to an available drone in the fleet.
R9	Each flight plan will be in one of three states: pending, current, completed.
R10	Flights will be assigned to drones in the order in which they are received.
R11	The bounds of the flight zone will be specified in terms of longitude and latitude.
R12	Drones will only be displayed on central command screens if they are located within the flight zone.
R13	A drone shall regularly transmit its current GPS coordinates to central command.
R14	Central command will track the current coordinates of all drones in flight.
R15	When a flight plan is assigned to a drone, the drone will fly to the targeted altitude.
R16	During the flight the drone travel from one waypoint to another as ordered by the flight plan.
R17	Upon reaching the final waypoint the drone shall land.
R18	A drone will always be in one of five flight modes: grounded, awaitingTakeOff, inFlight, onApproach, onLand.
R44	A drone will always be in one of three safety modes: normal, diverted, halted.
R19	Minimum separation distance shall always be maintained between all drones.
R20	A safety component shall direct the flight path of any drone which approaches a shared location.
R21	When drones take-off from a shared location, their flight shall be staggered to avoid collisions.
R22	When two or more drones approach minimum separation distance violations will be detected.
R23	Drones must maintain sufficient voltage to return to base.
R24	Drones shall ascend vertically during takeoff until the targeted altitude has been reached.
R25	The maximum number of drones flying in the flight zone shall be limited to two.
R26	While the system is in virtual model, a flight simulator will compute the current status of each drone.
R28	The current status of each flight plan including its waypoints, dispatch status, and assigned drone.
R29	Drones shall be represented visually as they move across the flight zone.
R30	Coordinates shall be specified using longitude and latitude degrees compatible with the flight simulator.
R31	Altitude shall be specified in meters.
R32	Drones shall maintain a minimum separation distance of 3 meters between waypoints.
R33	Drones shall maintain a minimum separation distance of 3.5 meters between drone launches.
R35	A central flight manager will coordinate all drone launches.
R36	Bases will be shown on the display.
R37	Each drone shall be assigned to a unique home base.
R40	While the system is in virtual mode, the system shall dynamically create drone instances.
R41	Bases shall not physically overlap other bases.
R42	While the system is in physical mode, an operator will place each drone physically on the display.
R43	While the system is in physical mode, the home base of each drone shall be assigned to a unique location.
R38	Drones shall return to their unique home base following each flight.

提取规则

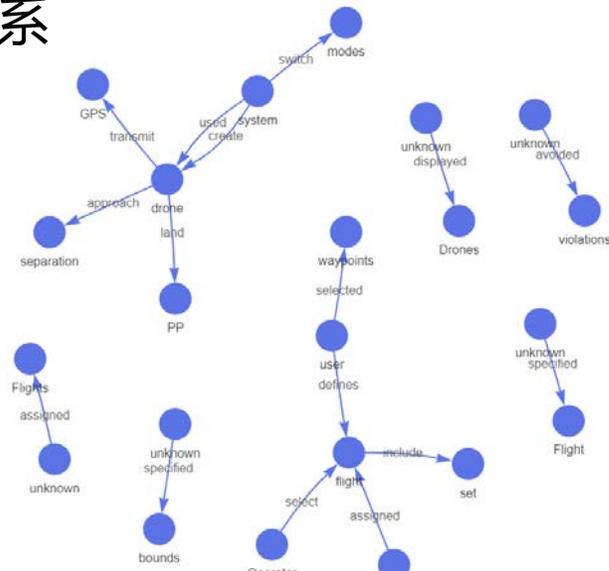


NLP技术

名词短语

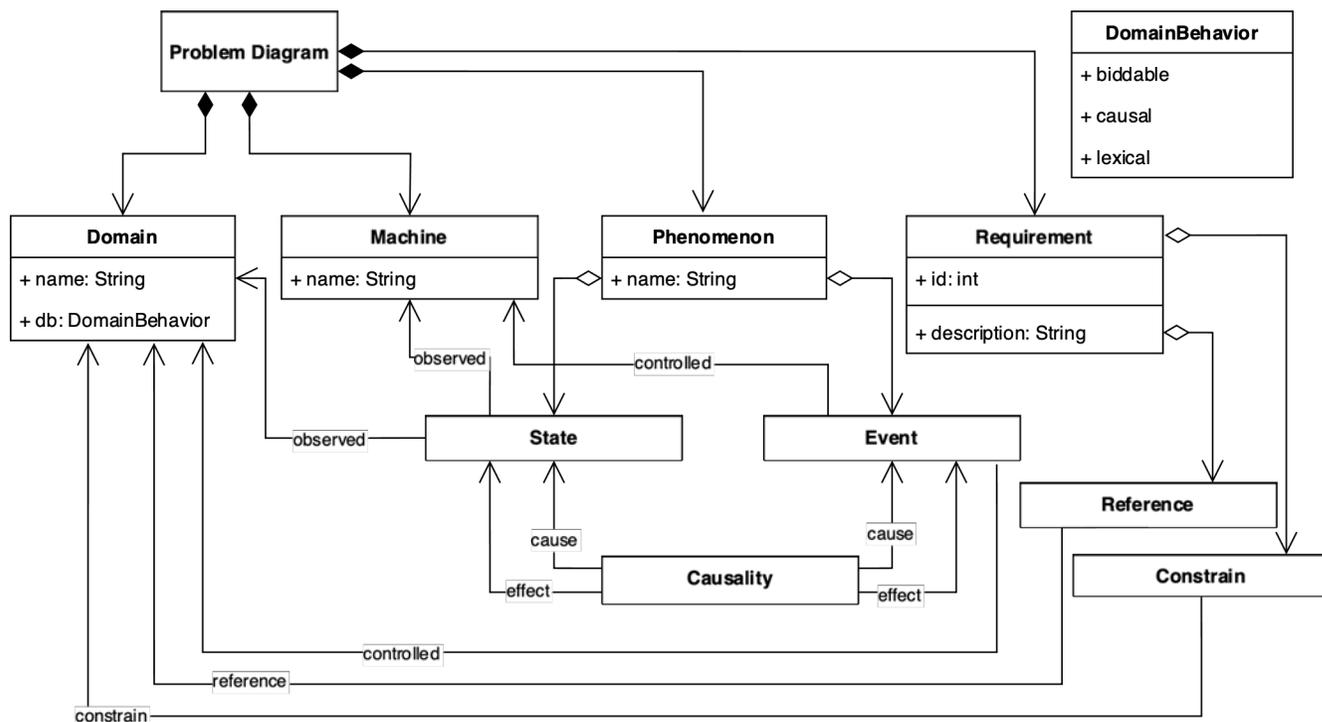
['user', 'flight plan', 'waypoints', 'drone', 'system', 'mode', 'flight zone', 'command screens', 'gps coordinates', 'safety component', 'flight path', 'separation distance', 'violation', 'flight simulator', 'location', 'altitude', 'home base', 'operator', 'xml format', 'longitude latitude']

动词关系



关系提取图数据展示

应用问题图元模型转换规则——以无人机需求文档为例



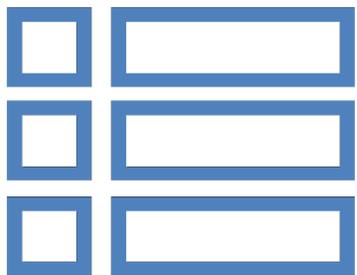
简化后的问题图元模型

规则

规则描述

- T1** 当三元组类型为事件时，将三元组的两个实体作为领域；当三元组类型为状态时，将三元组首个实体作为领域；当三元组的实体未出现过，创建一个新的领域对象；对于已经存在的领域实体，合并到同一个领域对象。
- T2** 对于领域间未存在现象的领域对，将事件作为现象，创建现象边；对于已存在现象的领域对，将新的现象并入同一边。
- T3** 当三元组类型为状态时，将领域和状态加入需求状态列表。
- T4** 对于每个需求对应的领域集合，如果不存在相应的需求列表，我们创建一个新的需求节点，将需求关联领域和状态列表作为属性加入；如果已存在领域集合对应的需求节点，我们更新需求列表和领域状态列表。

应用问题图元模型转换规则——以无人机需求文档为例



三元组列表

领域和现象列表

模型转换规则

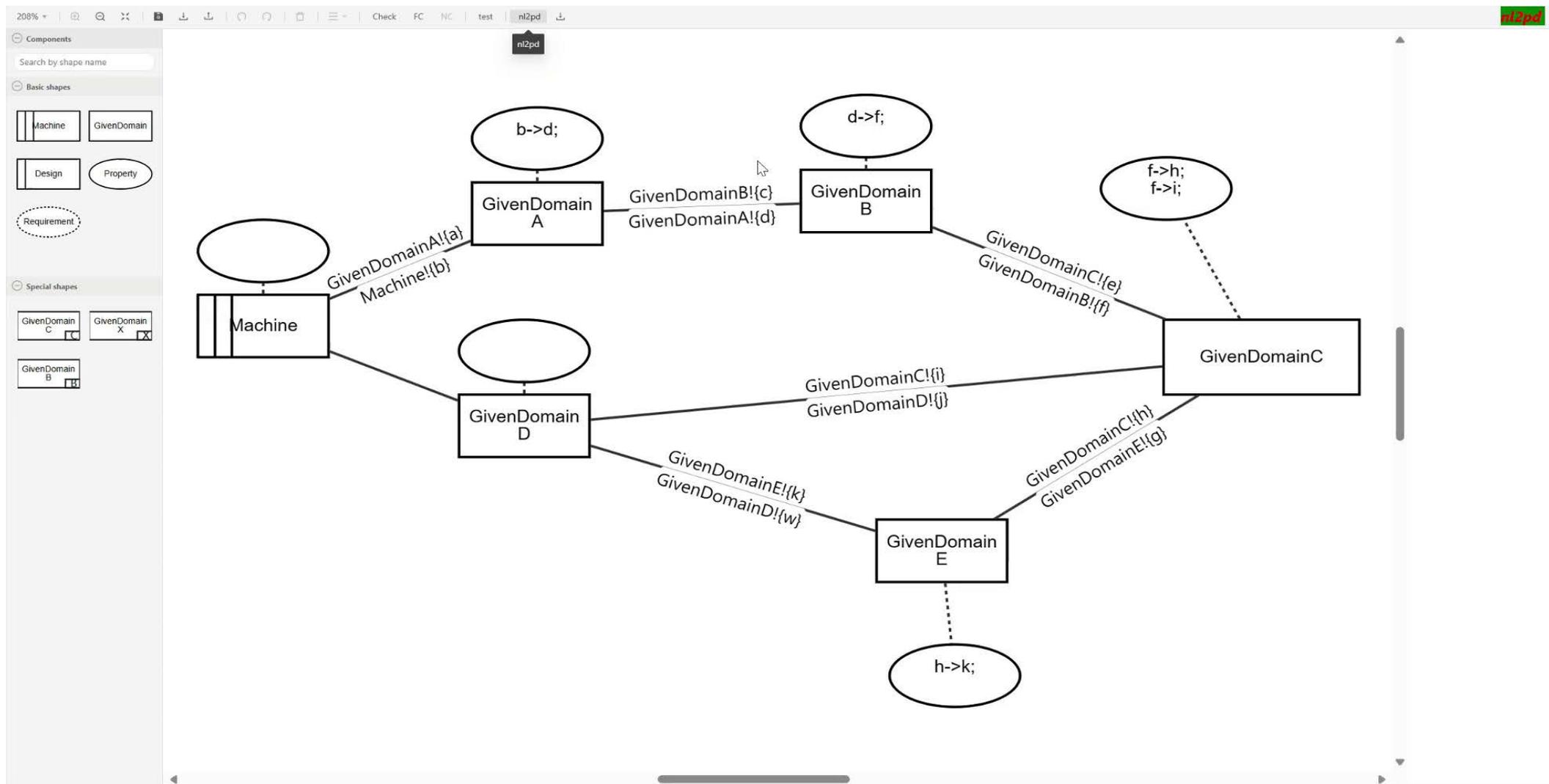


需求和状态列表

- name: drone, id: 0 - name: system, id: 1 - name: mode, id: 2 - name: flight plan, id: 3 - name: operator, id: 4 - name: waypoint, id: 5 - name: flight, id: 6 - name: flight zone, id: 7 - name: command, id: 8 - name: altitude, id: 9 - name: separation distance, id: 10 - name: safety component, id: 11 - name: voltage, id: 12 - name: base, id: 13 - name: flight simulator, id: 14 - name: longitude latitude degree, id: 15 - name: coordinate, id: 16 - name: flight manager, id: 17 - name: drone launch, id: 18 - name: basis, id: 19 - name: user, id: 20 - name: xml format, id: 21 - name: placeholder, id: 22 - name: location description, id: 23 - name: user longitude latitude, id: 24	1. source: system, target: mode, action: switch at runtime 2. source: operator, target: flight plan, action: select 3. source: flight plan, target: waypoint, action: include 4. source: drone, target: flight plan, action: be assigned / is assigned 5. source: drone, target: flight, action: be assigned in the order 6. source: drone, target: command, action: transmit gps coordinate 7. source: drone, target: altitude, action: fly 8. source: safety component, target: drone, action: direct 9. source: safety component, target: separation distance, action: approaches 10. source: separation distance, target: flight, action: be staggered prevent 11. source: drone, target: separation distance, action: approach / maintain at all times 12. source: drone, target: voltage, action: maintain 13. source: drone, target: base, action: return / return following each flight 14. source: flight simulator, target: drone, action: compute 15. source: longitude latitude degree, target: coordinate, action: be specified using 16. source: flight manager, target: drone launch, action: coordinate 17. source: system, target: basis, action: create within the bounds of the flight zone 18. source: base, target: basis, action: overlap 19. source: operator, target: drone, action: place at an operator launch location 20. source: user, target: flight plan, action: create / defines / is defining 21. source: xml format, target: flight plan, action: be saved 22. source: placeholder, target: flight zone, action: Show on the virtual display 23. source: user, target: waypoint, action: defines
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
{  
  "domainIds": "0",  
  "reqs": [  
    "R0: A fleet of drones will be generated at simulation startup.",  
    "R17: Upon reaching the final waypoint the drone shall land.",  
    "R18: A drone will always be in one of five flight modes grounded, taking off, flying, or landing.",  
    "R19: A drone will always be in one of three safety modes normal, diverted, halted.",  
    "R26: The maximum number of drones flying in the flight zone shall be limited to two.",  
    "R29: Drones shall be represented visually as Drones move across the flight zone.",  
    "R42: Drones shall land at Drones home base following each flight.",  
    "R43: The current voltage of each drone shall be tracked."  
  ],  
  "propertyList": [  
    {"idx": 0, "property": "be generated at simulation startup"},  
    {"idx": 0, "property": "land"},  
    {"idx": 0, "property": "be in one of five flight modes grounded, awaiting TakeOff, taking off, flying, or landingflight mode"},  
    {"idx": 0, "property": "be in one of three safety modes normal, diverted, halted safety mode"},  
    {"idx": 0, "property": "be limited"},  
    {"idx": 0, "property": "be represented"},  
    {"idx": 0, "property": "move across the flight zone"},  
    {"idx": 0, "property": "land at Drones home base following each flight"},  
    {"idx": 0, "property": "be tracked"}  
  ]  
}
```


▶ 开发NL2PD平台可视化建模



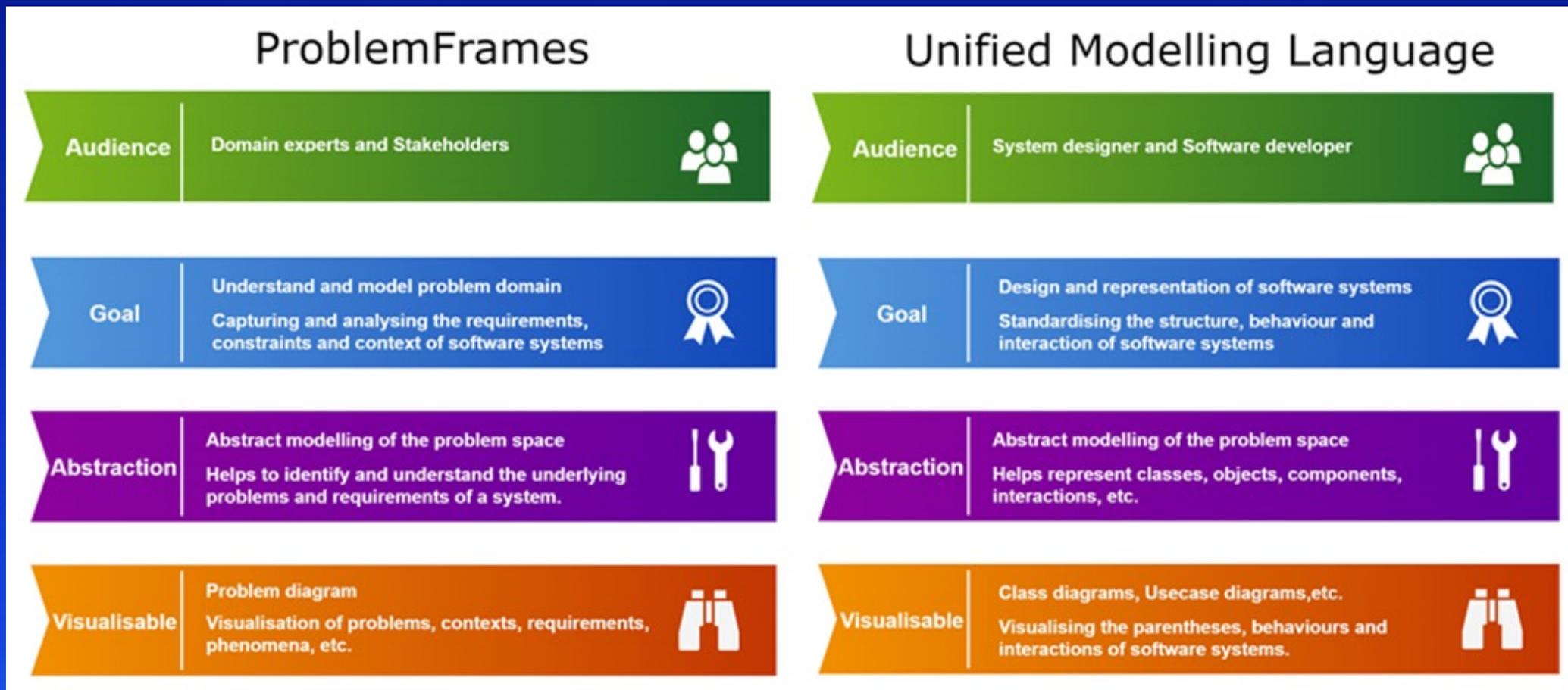
current path:

PART 05

需求模型至软件规格的转换

► 研究意义

软件应用设计阶段：UML模型可以通过自动化工具直接映射为可执行代码，实现从设计到代码的快速转换和迭代。UML还提供了**类图**、**用例图**、**顺序图**等多个视角的系统设计支持。



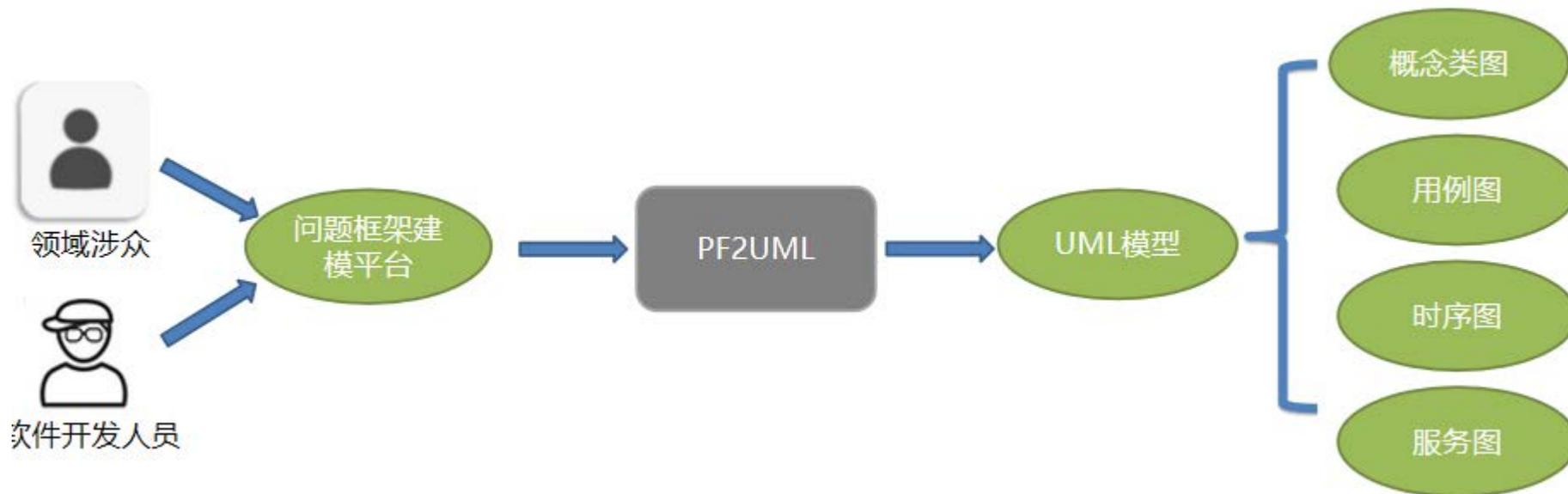
▶ 研究意义

研究方向:

使用自然语言处理 (NLP) 技术自动从需求文档中提取功能性和非功能性需求, 生成初步的需求模型。通过模型驱动工程 (MDE) 技术, 将提取的需求模型自动转换为UML模型, 确保从需求分析到系统设计的自动化衔接。

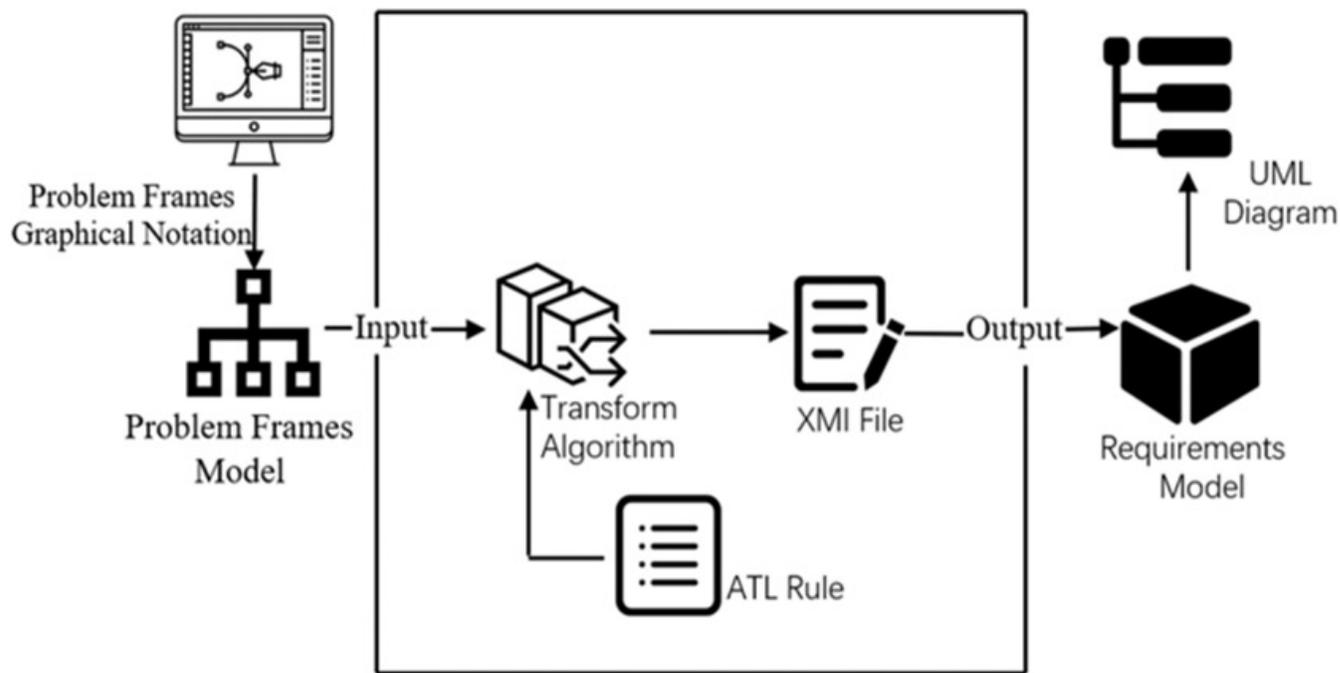
研究意义:

能够自动化地获取、分析和建模需求, 不仅提高设计的效率和一致性, 还降低了从需求到实现阶段的错误风险, 确保系统开发的高效性和可靠性。



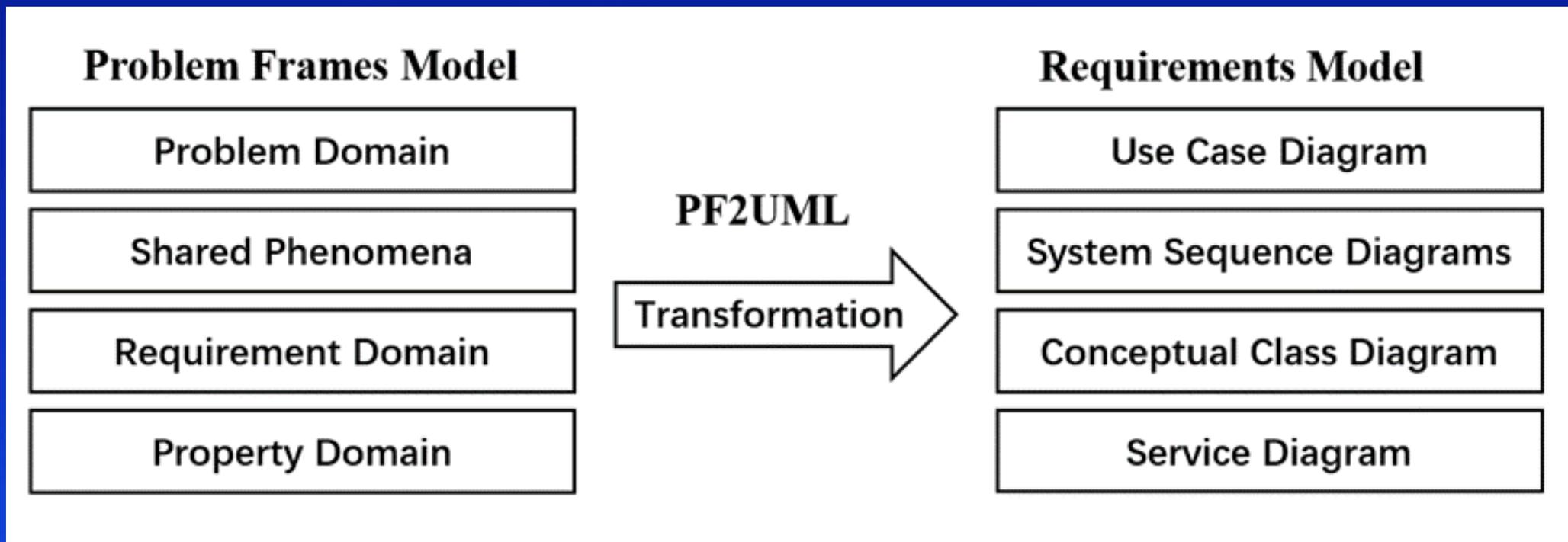
▶ PF2UML

使用包含**ATL转换规则**的自动化工具，处理需求中的**实体、关系、现象等信息**，将问题框架模型转换为UML模型。通过该转换算法，自动生成需求模型，**最终输出一个完整、可执行的需求规格说明**，并以4种UML图（**类图、用例图、顺序图、服务图**）的形式呈现，以支持后续系统设计和实现。



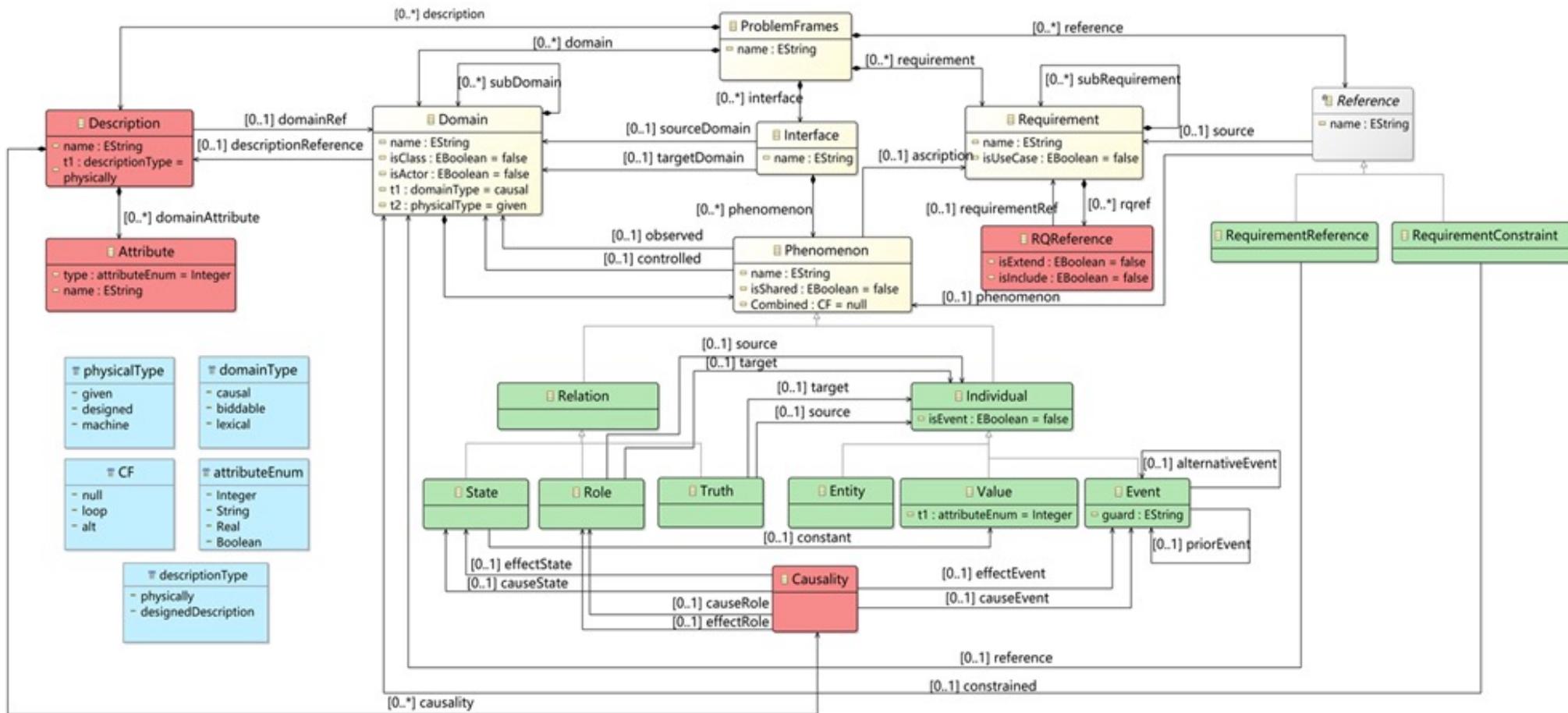
▶ PF2UML-模型转换

PF2UML 转换算法的设计思路如图：将问题模型中的问题领域、共享现象、需求领域以及属性领域的信息转换为 UML 中的概念类图、用例图和序列图。



PF2UML-问题框架元模型

问题框架方法元模型定义了用于描述问题框架的结构，元模型的根源称为**问题框架 (ProblemFrames)**，它代表了最基本的概念。该元模型包括**领域 (Domain)**、**接口 (Interface)**、**现象 (Phenomenon)** 和**需求 (Requirement)**等概念，描述了问题领域、问题现象和问题需求之间的关系，为问题分析和建模提供支持。

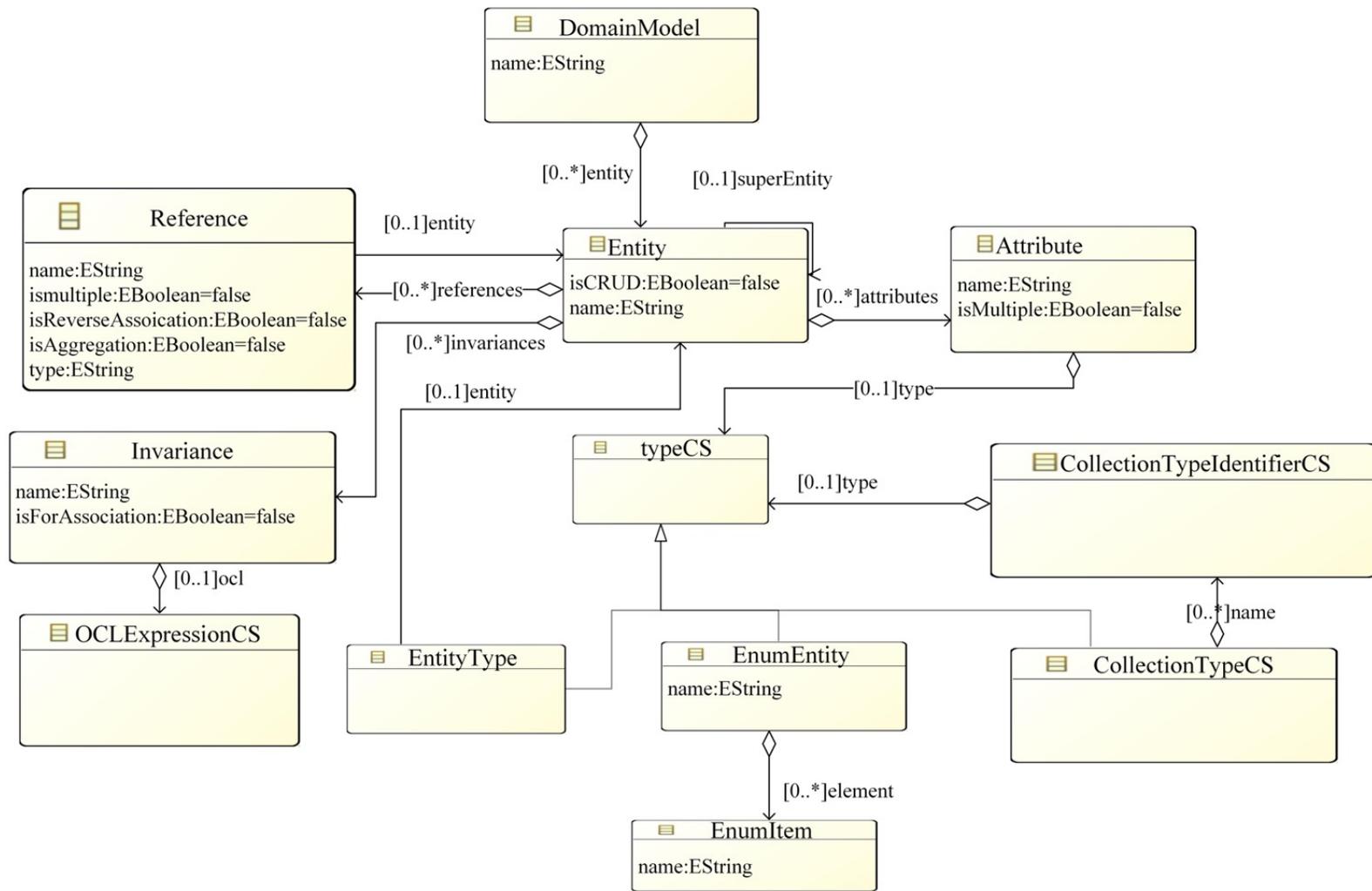


▶ PF2UML-概念类图

输入元模型：问题框架元模型

输出元模型：类图元模型

类图元模型顶层为DomainModel，包含有多个Entity，Entity包含多个Reference和Attribute。

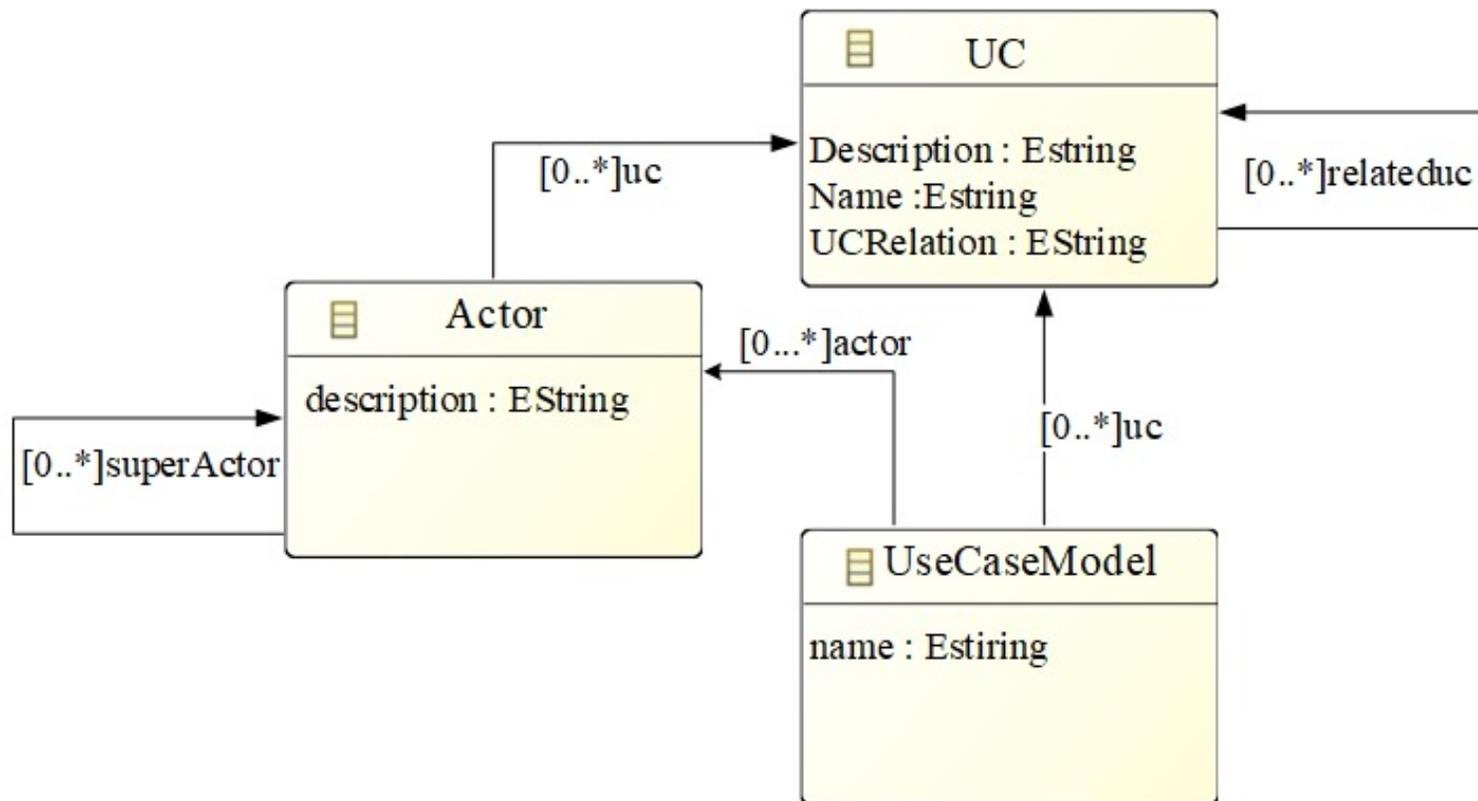


▶ PF2UML-用例图

输入元模型: 问题框架元模型

输出元模型: 用例图元模型

UseCaseModel为用例模型的根节点, UC为用例, Actor为参与者。

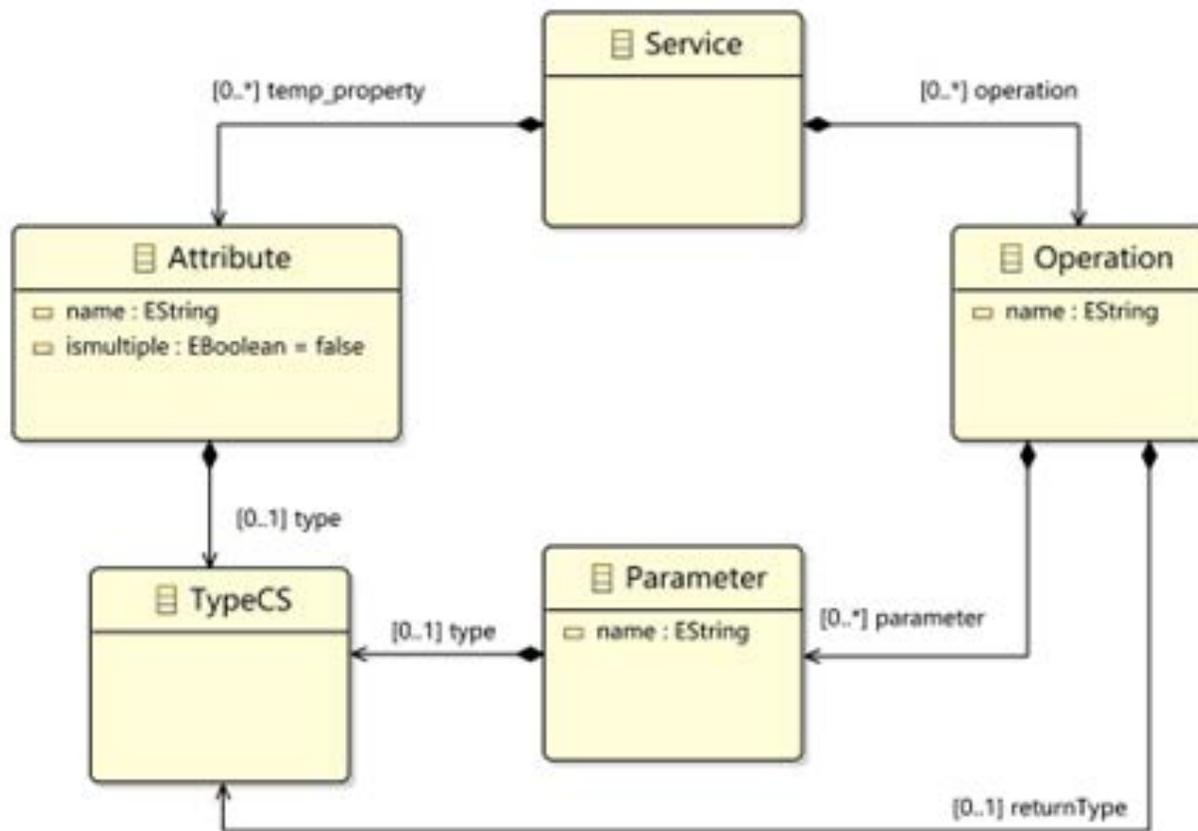


▶ PF2UML-服务图

输入元模型：问题框架元模型

输出元模型：服务图元模型

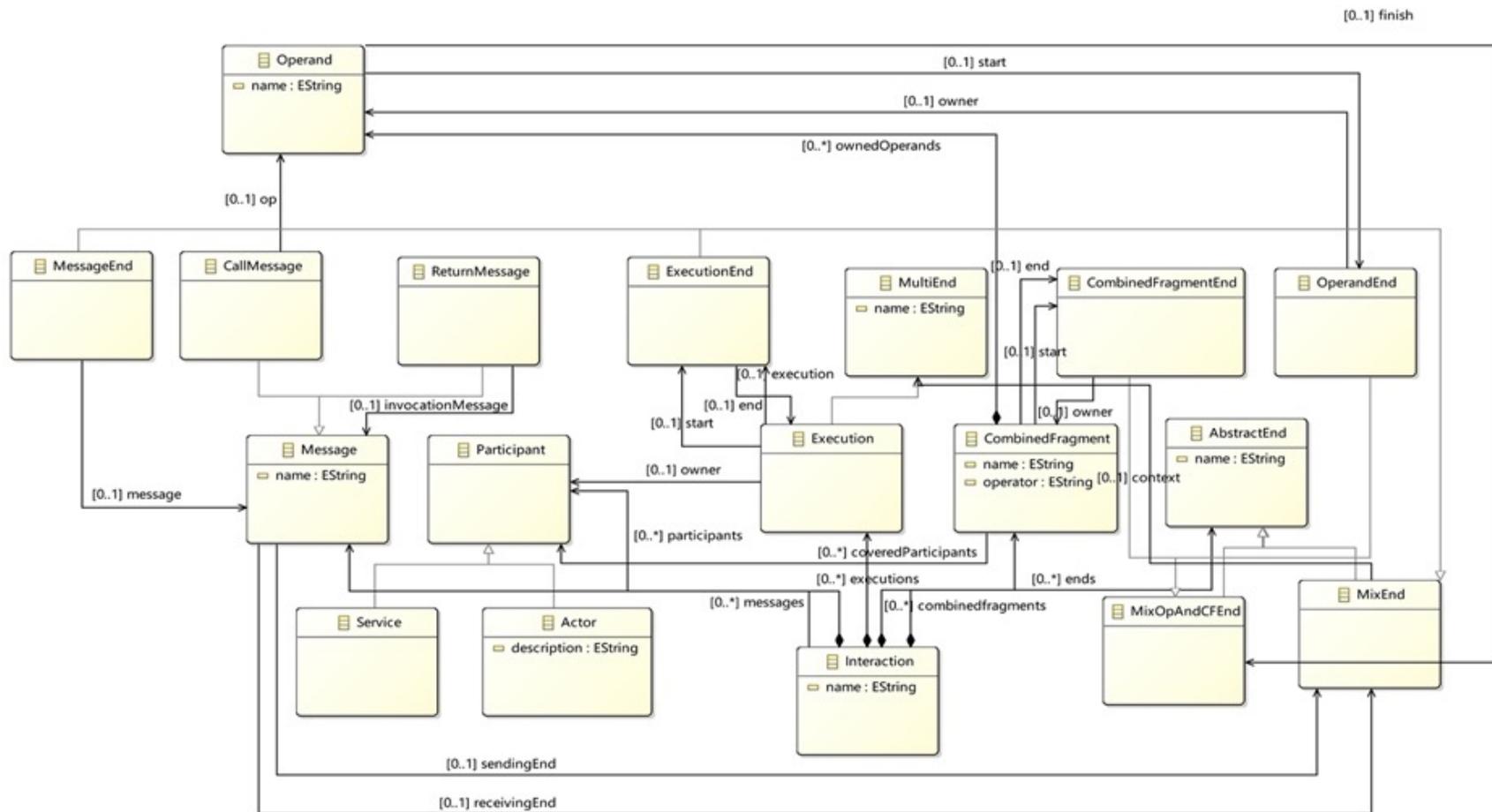
Service为服务图模型的根节点，
Attribute为属性，Operation为操作，
操作中包含Parameter。
TypeCS为Parameter和Attribute的
数据类型。



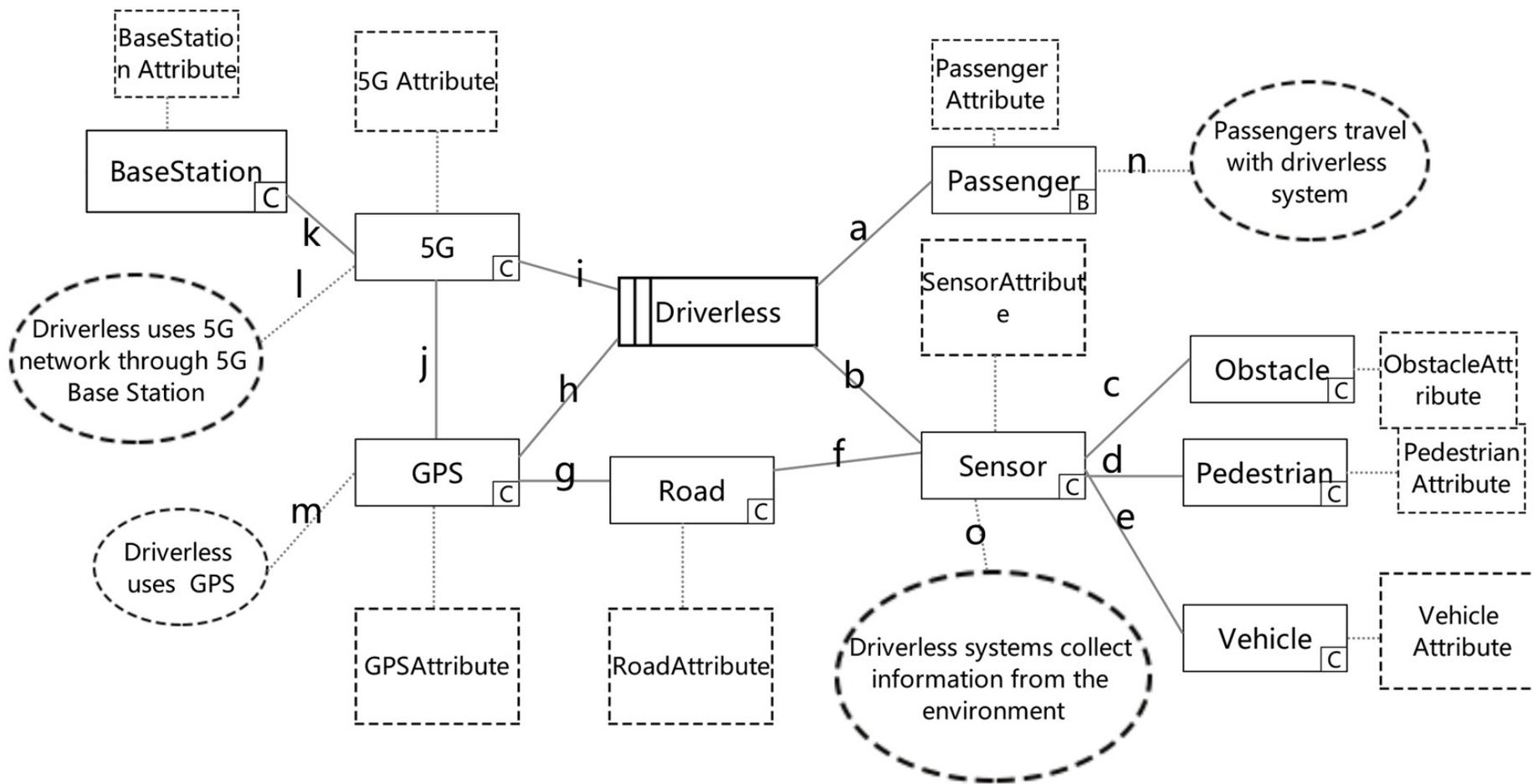
▶ PF2UML-顺序图

输入元模型：问题框架元模型
输出元模型：顺序图元模型

Interaction为用例模型的根节点，包含零至多个消息（Message）、执行动作（Execution）、组合片段（CombinedFragment）和结果（AbstractEnd），还包含一个指向参与者（Participant）的依赖（participants）。



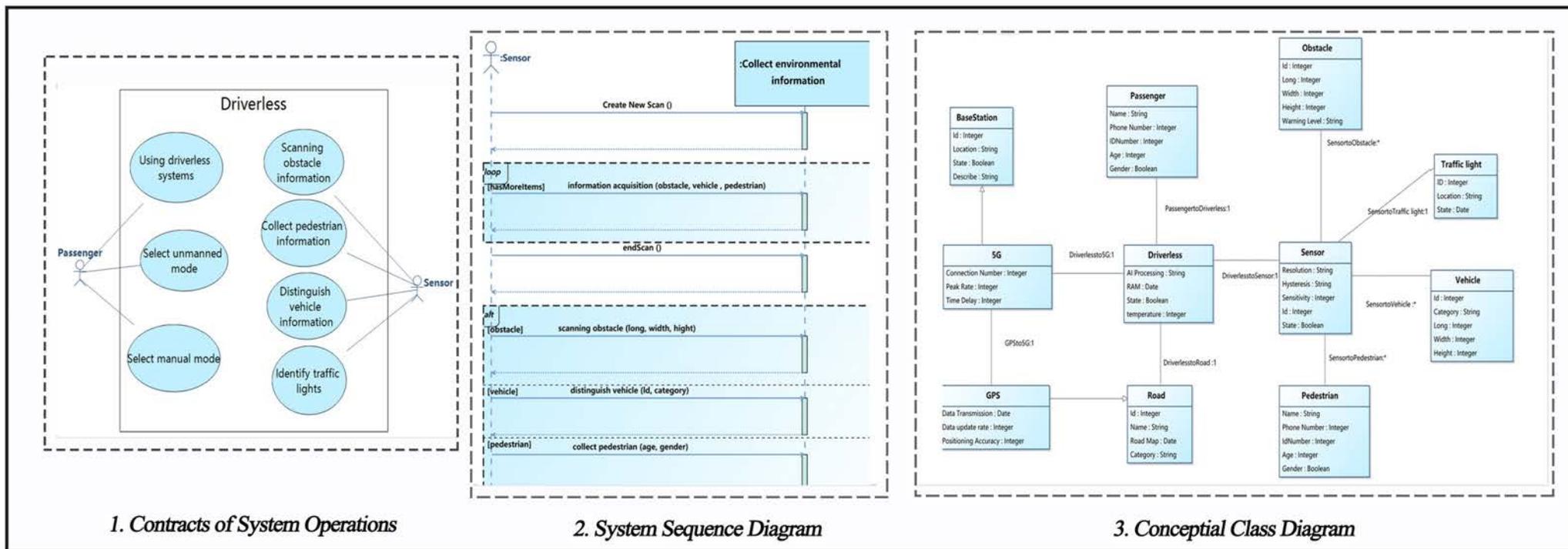
案例研究



由NLP2PF生成自动驾驶系统的问题图，展示了自动驾驶系统中各关键实体（如传感器、车辆、5G、GPS等）及其相互关系，反映了系统的基本结构。

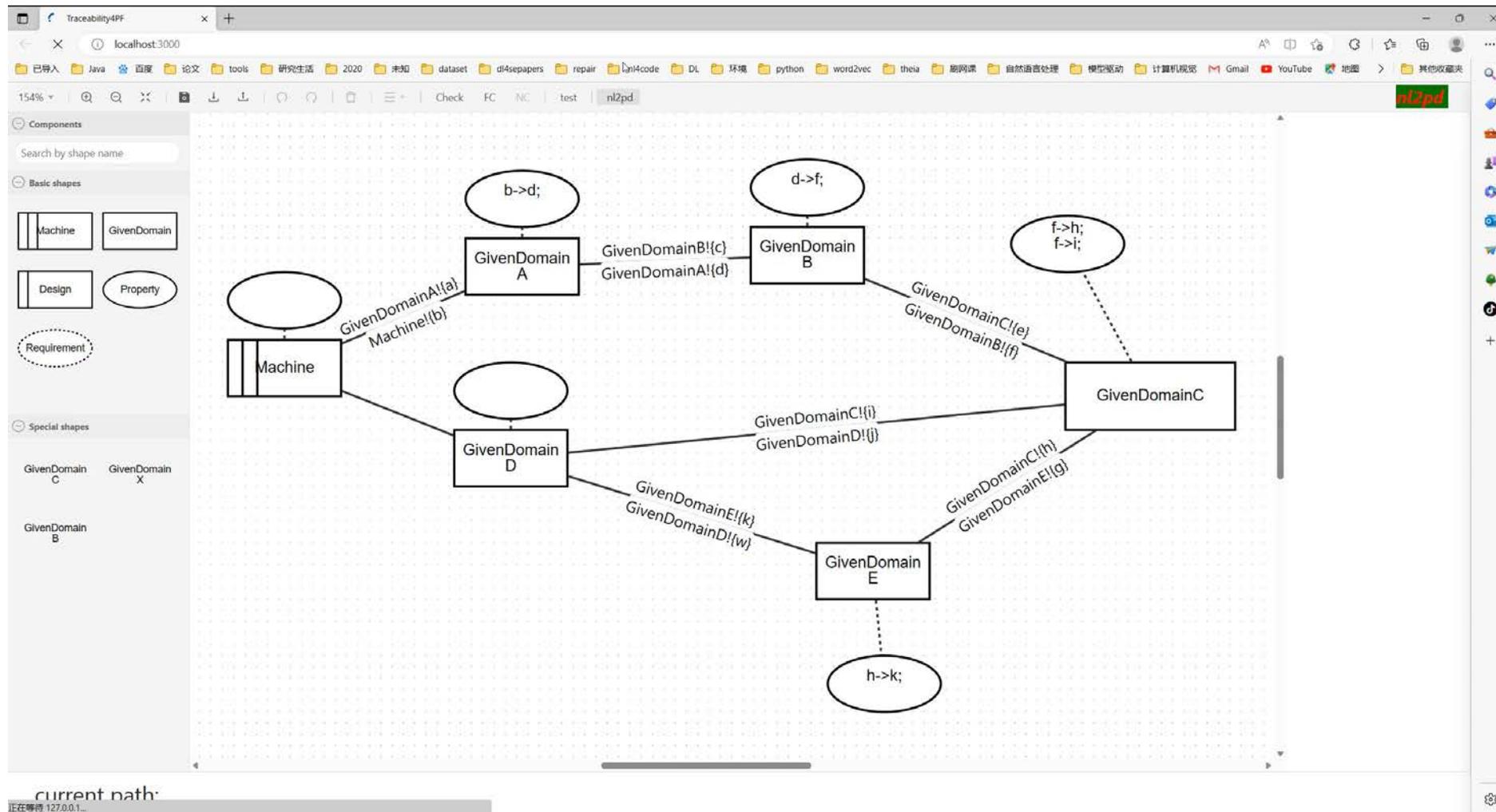
案例研究

Requirements Model



通过PF2UML，完成模型的自动转换。得到的需求模型通过Contracts of System Operations、System Sequence Diagram和Conceptual Class Diagram的组合，清晰地呈现了自动驾驶系统的主要操作流程、交互顺序以及系统中的核心实体。

RE自动化工具集视频演示1-需求语句自动生成问题图模型



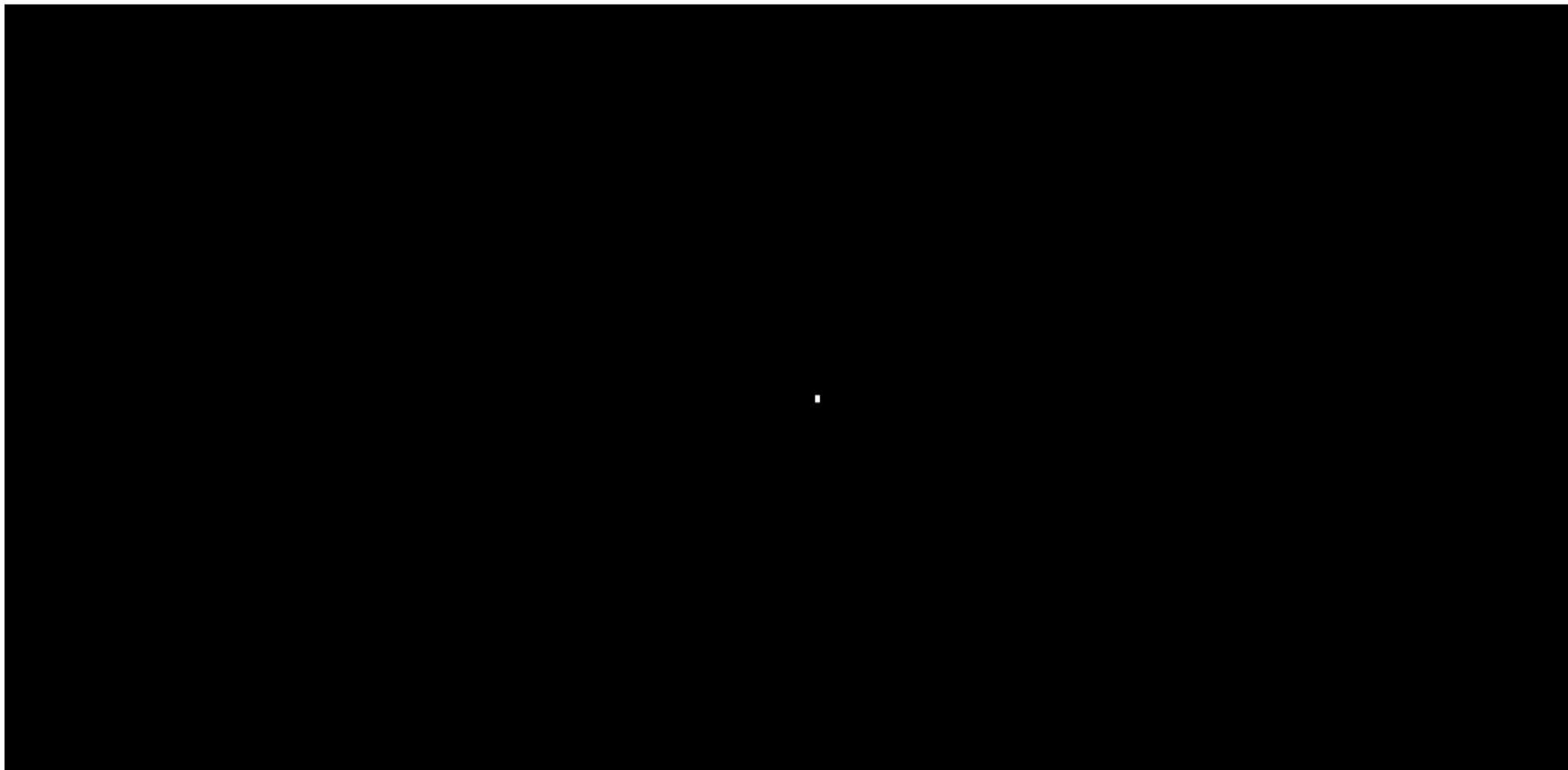
▶ 视频演示2-平板用户手绘建模及模型自动拆分及场景生成

Section 1

Hand Gesture Recognition

CARE 2.0

▶ 集视频演示3-网页版多利益相关者建模及自动拆分及场景生成



科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



 **K+峰会**  **敦煌站**

K+ 思考周®研习社

时间: 2025.08.29-30

 **K+峰会**  **上海站**

K+ 金融专场

时间: 2025.10.17-18

 **K+峰会**  **香港站**

K+ 思考周®研习社

时间: 2025.11.25-26



K+峰会详情



 **AiDD峰会**  **上海站**

AI+研发数字峰会

时间: 2025.05.17-18

 **AiDD峰会**  **北京站**

AI+研发数字峰会

时间: 2025.08.08-09

 **AiDD峰会**  **深圳站**

AI+研发数字峰会

时间: 2025.11.28-29



AiDD峰会详情



利用AI技术深化计算机对现实世界的理解

推动研发进入智能化时代

