

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розроблення автоматизованої системи передачі знань для роботів-манипуляторів

Виконав(ла): студент(ка) IV курсу, групи КТ-41

спеціальності 151 – Автоматизація та комп'ютерно-

інтегровані технології

(шифр і назва спеціальності)

Пшеничняк В.І.

Керівник

(підпис)

Стухляк П.Д.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Чихіра І.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Голотенко О.С.

(прізвище та ініціали)

Рецензент

(підпис)

Белякова І.В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)
Кафедра комп'ютерно-інтегрованих технологій
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Голотенко О.С.
(підпис) (прізвище та ініціали)
« » 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)
за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва спеціальності)
Студенту Пшеничний Владислав Ігорович
(прізвище, ім'я, по батькові)
1. Тема роботи Розроблення автоматизованої системи передачі знань для роботів-маніпуляторів
Керівник роботи доктор технічних наук, професор кафедри КТ Стухляк Петро Данилович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом ректора від «14» квітня 2026 року № 4/9-183
2. Термін подання студентом завершеної роботи 25 червня 2026р.
3. Вихідні дані до роботи Технічні вимоги до роботи роботів-маніпуляторів

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

Аналітична частина

Проектна частина

Спеціальна частина

Безпека життєдіяльності, основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Титульний слайд

Актуальність роботи

Завдання роботи

Основна частина

Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності,			
основи охорони праці			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.05.2026	
2.	Підбір джерел по темі роботи	22.05.2026-24.05.2026	
3.	Опрацювання публікацій та збір даних по темі роботи	24.05.2026-27.05.2026	
4.	Виконання роботи згідно мети	27.05.2026-29.05.2026	
5.	Оформлення першого та другого розділів	29.05.2026-31.05.2026	
6.	Оформлення третього розділу	31.05.2026-11.06.2026	
7.	Оформлення четвертого розділу	11.06.2026-13.06.2026	
8.	Підготовка презентації	14.06.2026	
9.	Підготовка доповіді	15.06.2026	
10.	Оформлення кваліфікаційної роботи	15.06.2026	
11.	Нормоконтроль	16.06.2026	
12.	Перевірка на плагіат	16.06.2026	
13.	Попередній захист кваліфікаційної роботи	16.06.2026	
14.	Захист кваліфікаційної роботи	25.06.2026	

Студент

(підпис)

Пшеничняк В.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Стухляк П. Д.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота бакалавра складається з пояснювальної записки та графічної частини (ілюстративний матеріал – слайди).

Об'єм графічної частини кваліфікаційної роботи становить _____.

Об'єм пояснювальної записки складає ____ друкованих сторінок формату А4.

В роботі використано ____ літературних джерел.

У роботі було розглянуто проблему негнучкості в сучасних виробничих процесах, що включають робототехніку та програмування роботів, і запропоновано рішення, в якому використовується фреймворк моделювання завдань на основі SysML для моделювання можливостей робота в заданій області застосування. Використовуючи цей фреймворк для моделювання навичок робота на вищому рівні абстракції, показано, що є можливість внести більше гнучкості у виробничий процес, розглядаючи програмування та взаємодію роботів на рівні завдань.

Підійшовши до проблеми таким чином, була отримана можливість абстрагуватися від багатьох труднощів, пов'язаних із сучасними методами взаємодії, і водночас створити структуру, яка може охопити простір взаємодії, і тим самим забезпечити передачу знань у нашій цільовій області.

Ключові слова: ЗНАННЯ. РОБОТ-МАНІПУЛЯТОР, ЗАВДАННЯ, АВТОМАТИЗАЦІЯ, СКЛАДАННЯ

ЗМІСТ

ВСТУП	5
1. АНАЛІТИЧНА ЧАСТИНА	6
1.1 Огляд стану проблеми керування роботами-маніпуляторами з переналаштуванням	6
1.2 Структури представлення та передачі знань для роботів-маніпуляторів	10
2. ПРОЄКТНА ЧАСТИНА	18
2.1. Система передачі знань у робототехніці	18
2.2 Мова системного моделювання (SysML).	19
2.3 Фреймворк на основі SysML для передачі знань у робототехніці.....	25
2.4 Абстракція навичок.....	33
3 СПЕЦІАЛЬНА ЧАСТИНА.....	41
3.1 Експериментальні результати	41
3.2 планування фреймворку	47
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	51
4.1 Організація охорони праці при роботі з системою управління	51
4.2 Електробезпека	53
4.3 Розрахунок заземлення	56
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ	60

ВСТУП

Сучасні технології досягли такого рівня, що справжній потенціал робототехніки починає розкриватися. Однак програмування роботів, щоб вони були надійними в різних середовищах та для різних цілей, доступним та інтуїтивно зрозумілим для більшості користувачів, все ще залишається складним завданням. Залишається низка незадоволених потреб. Наприклад, багато існуючих сьогодні рішень є власними, що ускладнює широке впровадження єдиного рішення. Крім того, більшість підходів чітко орієнтовані на конкретну реалізацію. Але незрозуміло, чи ці підходи будуть узагальнені на ширший спектр проблем та застосувань. Щоб вирішити ці проблеми, ми визначаємо простір взаємодії, або простір, створений взаємодією між роботами та людьми. Цей простір використовується для класифікації відповідних існуючих робіт та концептуалізації цих незадоволених потреб. GTax, фреймворк для передачі знань, представлений як рішення, здатне охопити простір взаємодії. Фреймворк базується на SysML, стандарті, що використовується в багатьох різних системах, який забезпечує формалізоване представлення та верифікацію. У цій роботі ми демонструємо, що шляхом узагальнення в просторі взаємодії ми можемо спростити програмування роботів та забезпечити передачу знань між процесами, системами та доменами застосувань.

1. АНАЛІТИЧНА ЧАСТИНА

1.1 Огляд стану проблеми керування роботами-маніпуляторами з переналаштуванням

Робототехніка має величезний потенціал. Протягом десятиліть люди з усіх верств суспільства уявляли собі безліч різних способів, якими роботи можуть принести позитивну користь суспільству – від роботів для обслуговування дому, що допомагають у виконанні домашніх завдань, до роботів-виробників, які працюють разом з людьми для виконання складних промислових завдань. Сьогоднішні технології досягли такого прогресу, що ми починаємо бачити, як цей потенціал реалізується різною мірою. Прикладами є Саймон, дослідницька платформа для вивчення взаємодії людини та робота, а також платформа Baxter, продукт для виробничої сфери. Baxter, нещодавній продукт Rethink Robotics, - це гуманоїдний робот із двома руками з 7 ступенями свободи, призначений для спільної роботи з людьми у виробничих умовах. Він оснащений датчиками та програмним забезпеченням, що дозволяє користувачам навчати робота виконувати певні завдання. Ці платформи є важливим кроком уперед до справжньої спільної роботи між людьми та роботами для досягнення складних виробничих цілей, що вже давно є святим Граалем робототехніки в промисловому виробництві.

Однак, незважаючи на ці численні досягнення в сучасному розвитку, програмування роботів для забезпечення надійної роботи в різних середовищах та для різних цілей, а також у спосіб, доступний та інтуїтивно зрозумілий для більшості користувачів, все ще залишається складним завданням, і залишається багато незадоволених потреб у існуючих технологіях. Наприклад, хоча Baxter має можливість програмування шляхом демонстрації для виконання певних завдань, якщо параметр завдання змінюється (наприклад, оновлення до іншого типу гвинта), робота потрібно буде перенавчити для успішного виконання

оновленого завдання; одні інші знання з попереднього випадку не можуть бути збережені чи використані повторно.

Ця проблема динамічно змінюваних вимог до завдань є поширеною у виробництві, і проблема крихких рішень не обмежується лише цією однією платформою. Це стосується більшості застосувань, пов'язаних з промисловими роботами, і є значною перешкодою для широкого впровадження робототехніки та технологій автоматизації в багатьох різних виробничих галузях. У дослідженні, проведеному Європейською Комісією щодо перспектив робототехнічної галузі в Європі було надано аналіз ланцюжка створення вартості роботів. У ньому зазначалося, що в структурі витрат на впровадження та розгортання робототехнічного рішення проблеми в промисловості, власне робот становить лише 25% від повної вартості, тоді як інші 75% припадають на витрати на інтеграцію. Найбільша окрема вартість (40%) насправді є програмуванням робота. А оскільки виробничі потреби вимагають використання цих роботів для вирішення різноманітних проблем, хоча деякі кошти можна заощадити за рахунок повторного використання самого робота, 75% вартості впровадження необхідно оплачувати знову і знову для кожної нової проблеми. Таким чином, у цьому контексті найбільшою перешкодою для робототехніки є не сам робот, а його розгортання в конкретних застосуваннях.

Суть цієї проблеми можна сформулювати як проблему передачі знань. В обох випадках існує неможливість представити знання в системі таким чином, щоб це сприяло модульності та можливості повторного використання в іншому контексті чи проблемі. Представляючи знання таким чином, щоб вони були портативними та придатними для повторного використання в різних сценаріях, це може зменшити вартість програмування та перерозподілу в різних додатках, потенційно заощаджуючи до 40% від загальної вартості системи. Важливо зазначити, що існують різні типи передачі знань, про які можна говорити. До них належать передача знань між людиною та роботом, коли певний тип знань передається між людиною та роботом (наприклад, як це спостерігається в програмуванні шляхом демонстрації, що майже виключно є передачею знань між

людиною та роботом), та передача знань між роботами, або коли знання передаються між роботами або системами без участі людського втручання. Для досягнення такого рівня передачі знань потрібен метод, який охоплюватиме простір взаємодії.

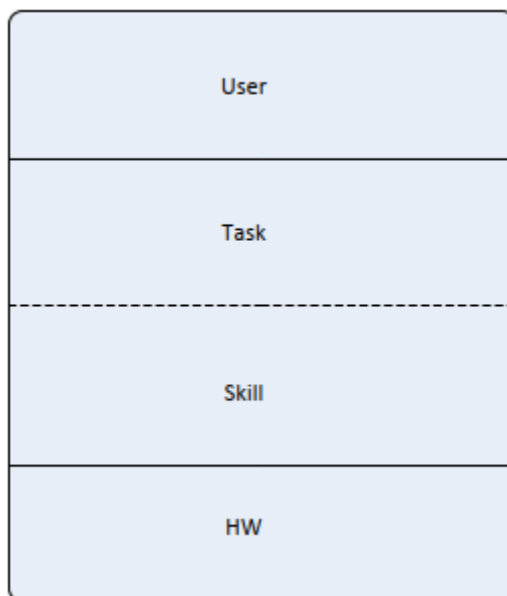
Технічний підхід.

Простір взаємодії.

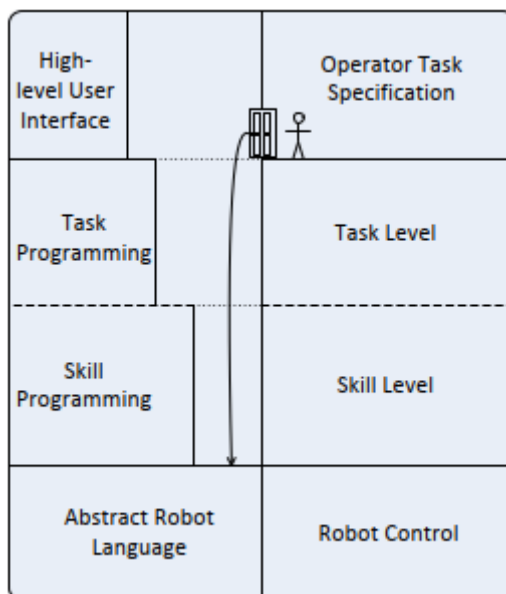
Простір взаємодії визначається як простір, створений перетином роботів, що працюють у людському середовищі, та співпрацею людини і робота. Рисунок.1показує Простір Взаємодії, розділений на різні області. На найнижчому рівні знаходиться рівень робота або пряме керування роботом. На найвищому рівні знаходиться рівень користувача. Решта простору розділена на дві різні області. Рівень навичок представляє абстракції низького рівня над рівнем робота. Рівень задач представляє абстракції та концепції вищого рівня. У контексті Простору Взаємодії різні типи передачі знань можна побачити досить легко. Знання можуть передаватися вертикально від людини до робота, роблячи робота доступним для людини, а людські знання відкритими для робота. Знання також можуть поширюватися горизонтально на графі між різними платформами та процесами роботів, роблячи існуючі знання робота відкритими та придатними для використання в різних контекстах та застосуваннях. Таким чином, можна побачити, що проблеми ефективної передачі знань та охоплення цього Простору Взаємодії, створеного співпрацею людини і робота, є одними й тими ж.

Рис. 1.1 показує різні рівні програмної взаємодії (ліворуч) у парі з відповідним рівнем специфікації (праворуч). На найнижчому рівні знаходиться абстрактна мова роботів або мова сценаріїв окремого робота для створення руху та дії в реальному світі. Багато з того, що відбувається в робототехніці сьогодні, відбувається на цьому рівні. Однак, якщо уявити користувача на верхньому рівні, рівні інтуїтивної взаємодії з системою, то до цього рівня, де відбувається більшість взаємодії, пролягає значна відстань, і цей перехід може бути складною перешкодою для багатьох користувачів. У міру того, як ми піднімаємося, ми бачимо більш абстрактні рівні взаємодії, а саме рівні взаємодії на рівні навичок

(такі як дискретно-подійні системи) та взаємодія на рівні завдань (наприклад, програмування шляхом демонстрації).



a)



б)

Рисунок 1.1 - Простір взаємодії.

Вершина складається з високорівневого, інтуїтивно зрозумілого інтерфейсу користувача. Було проведено багато робіт (включаючи попередній приклад Бакстера), які намагалися створити зв'язок між інтуїтивно зрозумілою

взаємодією та низькорівневим керуванням, що дозволило б оператору легко запрограмувати систему для виконання завдання. Однак, як загальне рішення, ці спроби зрештою зазнають невдачі через те, що реалізації не узагальнюються на весь простір взаємодії, а оновлення специфікації проблеми вимагають іншого рішення.

1.2 Структури представлення та передачі знань для роботів-маніпуляторів

GTax: на основі SysML Структура передачі знань

Щоб вирішити ці проблеми, у цій дисертації ми пропонуємо використовувати SysML як мову для створення фреймворку, що використовується для моделювання можливостей роботів у заданій прикладній області. SysML має низку переваг як вибір мови моделювання, зокрема те, що він є галузевим стандартом для моделювання складних систем і як такий має добре вивчену формальну специфікацію. Сам фреймворк має форму ієрархії абстракції, де найнижчий рівень абстракції представлений як апаратно-незалежні атомарні дії, які називаються примітивами навичок, які система повинна виконувати для виконання завдань у прикладній області. Ця ієрархія абстракції дозволяє користувачам та розробникам динамічно вибирати відповідний рівень абстракції для потреб системи та конкретного завдання.

Використання цього узагальненого представлення дозволяє застосовувати ці навички до широкого кола проблем, від безлічі різних завдань та описів завдань до завдань на різних платформах та системах. Використання представлення, представленого в рамках, підтримує рівень абстракції достатньо високим для узагальнення на багато різних описів проблем, а також на різних платформах, оскільки примітиви навичок існують вище рівня апаратної реалізації. Представлення також підводить специфікацію цих завдань до більш концептуально простого вигляду рівня, що полегшує користувачеві взаємодію з роботом, без необхідності витрачати час на вивчення нижчого рівня опису

можливостей та специфікацій параметрів окремих роботів. За допомогою роботи, виконаної в цій дисертації, ми покажемо, як фреймворк можна використовувати для охоплення простору взаємодії та тим самим забезпечити передачу знань у сфері застосування роботів.

У цьому розділі ми представляємо відповідні дослідницькі зусилля, що стосуються аспектів передачі знань та простору взаємодії (рис. 1.2).

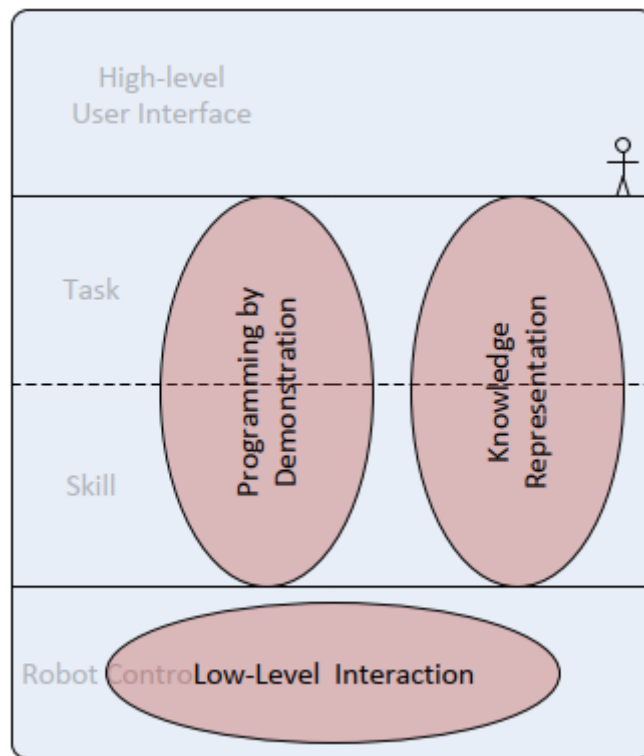


Рисунок 1.2 - Покриття простору взаємодії.

Обговорення розпочнеться з короткого огляду низькорівневих взаємодій з апаратним забезпеченням. Далі буде розглянуто важливу роль представлень знань та те, як вони використовувалися різними способами. Після цього будуть розглянуті підходи PbD та те, як ці різні методи застосовувалися. Нарешті, ми представимо кілька різних типів рамки передачі знань.

Взаємодія низького рівня.

Щодо простору взаємодії, найнижчий рівень можна розділити на дві частини (рис. 1.3). Перша секція представляє самого робота, тоді як друга секція представляє сприйняття для робота, таке як комп'ютерний зір. На цьому

низькому рівні існують певні моделі абстракції або системи, які роблять ці низькорівневі системи легшими у використанні.

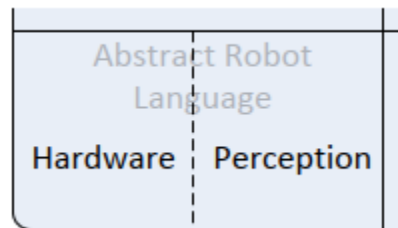


Рисунок 1.3 - Найнижчий рівень можна розділити на два розділи: апаратне забезпечення та сприйняття.

Що стосується апаратного забезпечення, то прикладами цього є операційна система робота, або ROS, система OROCOS та стандарт JAUS. ROS та подібні пакети працюють як модель абстракції робота (RAM) або як спосіб забезпечення спільного інтерфейсу для самого обладнання. Для сприйняття прикладами є пакет Open Computer Vision (OpenCV) та Бібліотека хмар точок (PCL). Такі пакети діють як модель абстракції сприйняття (PAM), або шар над певними датчиками для виконання корисних функцій сприйняття. Ці моделі абстракції є популярними та корисними інструментами для навченого техніків з робототехніки, але самі по собі все ще є досить низькорівневими. Незважаючи на це, багато реальних реалізацій роботів існують на цьому низькому рівні. Проблема полягає в тому, що цей найменший спільний знаменник є занадто низьким для такого високорівневого рівня взаємодії, необхідного для операторів-неекспертів, особливо коли йдеться про роботу в різних контекстах, таких як платформи та процеси. Тут нам потрібен спільний інтерфейс, де цей інтерфейс повинен абстрагувати деякі з цих часто складних технічних деталей, які можуть виникнути з моделями низькорівневої абстракції.

Окрім цих моделей абстракції, було створено низку мов програмування для спрощення програмування роботів та завдань. До них належать абстрактні, специфічні для роботів мови, такі як KRL для роботів KUKA та Karel для роботів Fanuc. Для загальних платформ використовуються такі мови, як мова опису завдань (TDL) та мова підтримки виконання (ESL) були створені.

Представлення знань.

З огляду на те, що програмування роботів є складним завданням, яке вимагає значних технічних знань, значна частина попередніх робіт була зосереджена на тому, як зробити програмування роботів більш інтуїтивно зрозумілим та доступним для операторів з меншим технічним досвідом. Значна частина цієї роботи була зосереджена на пошуку ефективних та результативних методів представлення системних знань для виконання цього завдання, з урахуванням певних вимог та обмежень на систему. Було запропоновано багато різних типів представлення знань.

Рівень майстерності.

Одна філософія працювала над кодуванням знань про задачу як функції руху. Прикладами такого типу представлення є динамічні системи та комплекси об'єкт-дія. Робота Лайонса та ін. визначили модель для обчислень роботів з використанням портових автоматів. Косецька та ін. використовував фреймворк дискретних подійних систем для моделювання завдань та поведінки робототехніки. До пізніших робіт належить робота Dantam, яка використовує граматичний підхід для представлення сенсомоторної інформації.

Рівень завдання.

Інша філософія дотримувалася точки зору, що важливі системні знання можна ефективно представляти символічно, наприклад, за допомогою дерев навичок або топологічні графи задач. Цей символічний підхід до представлення знань припускає, що система має більше властивих знань (вона знає, як поводитися зв'язки між символами та фізичними екземплярами), водночас вона дозволяє моделювати концепції вищого рівня, ніж представлення на основі руху. Робота Кресс-Газіта використовує лінійну часову логіку для моделювання специфікацій завдань, щоб створювати правильні контролери роботів для різних завдань. Деякі дослідники використовували цей символічний підхід для вирішення питання повторного використання знань або передачі знань у сферах поза виробництвом.

Також було проведено роботу з представлення конкретних цілей виробництва та складання, наприклад, із застосуванням мереж Петрі. Інший підхід – це використання графів І/АБО для перерахування всіх можливих шляхів через процес складання для досягнення загальної мети (наприклад, зібраного продукту). Далі пропонується використовувати алгоритм пошуку графів для знаходження відповідного шляху через граф на основі конкретних специфікацій проблеми.

Онтології використовувалися в багатьох різних фреймворках як інструмент представлення для обміну знаннями (в першу чергу семантичними або реляційними) між системами. Системи, такі як ConceptNet (який використовує базу даних MIT Open Mind Common Sense) та CYC - спроба змодельовати знання здорового глузду людини як реляційну онтологію з невеликим, заздалегідь визначеним набором семантичних зв'язків, що визначають, як поняття пов'язані між собою. KNOWROB – це специфічний для роботів фреймворк, який базує своє представлення знань на онтологічній реалізації, тут з використанням OWL2, як стандарту онтології. Онтологія визначається навколо концепцій та відносин, важливих для роботів, що працюють у реальному світі. У роботі Ліма та ін. запропоновано подібний підхід на основі онтологій для знань роботів у приміщенні.

Програмування шляхом демонстрації.

Програмування шляхом демонстрації (PbD) – це навчання через демонстрацію або імітаційне навчання – галузь досліджень, основною метою якої є передача знань між людиною та роботом. У цьому підході користувач або оператор має навчити робототехнічну систему виконувати певне завдання або мету, взаємодіючи з роботом певним чином. Ця взаємодія може приймати кілька різних форм. Часто оператор фізично демонструє виконуване завдання, а система використовує певний тип візуального сприйняття для отримання інформації від користувача, наприклад, камери або 3D-датчики, для фіксації руху користувача та відповідних особливостей навколишнього середовища. Потім ця інформація обробляється, щоб пов'язати фізичний рух оператора зі змінами в

навколишньому середовищі, а потім пов'язати цю інформацію з операціями, які необхідно виконати для виконання завдання. Таким чином, відповідні знання, пов'язані з виконанням завдання, передаються в систему. Інші методи взаємодії з системою включають фізичну взаємодію, або шляхом переміщення кінцевого ефектора робота вручну для демонстрації завдання, або шляхом керування роботом за допомогою іншого типу інтерфейсу (наприклад, джойстика, графічного інтерфейсу тощо). Подібна інформація отримується шляхом моніторингу власної роботи робота, і робляться висновки для узагальнення знань, щоб пов'язати це з виконанням завдання.

У рамках PbD (Personal Broadcasting Design) пропонується багато різних підходів та методів програмування робота через взаємодію з людиною. Ці різні підходи вписуються в різні місця в просторі взаємодії, показаному на рисунку 2. Слід зазначити, що хоча деякі конкретні роботи PbD явно розташовані в рамках низькорівневого навчання навичок або високорівневого опанування завдань, багато робіт важче класифікувати, як через відмінності у визначенні навичок та завдань, так і через зусилля різних дослідників, спрямовані на виконання обох визначень у цій роботі як навичок та завдань. Однак, хоча ці методи можуть охопити більше простору взаємодії вертикально, незрозуміло, чи узагальнюються ці методи горизонтально по всьому простору.

Рівень майстерності.

Деяка робота чітко відповідає рівню взаємодії навичок. Тут використовуються примітиви руху для категоризації людських рухів та виконання завдань, методи машинного навчання, такі як моделі гаусової суміші, для категоризації навичок з точки зору різних рухів, використання даних про силу для вивчення навичок складання низького рівня.

Рівень завдання.

Тут створюється рівень завдань у просторі взаємодії. Можна отримати описи завдань, спостерігаючи за тим, як люди виконують складальні завдання, використовувати аналіз людської діяльності для вивчення описів завдань для

домашніх справ, використання ключових кадрів для розуміння специфікацій завдань.

Програмування за допомогою інструкцій.

Іншим методом передачі людських знань у роботизовану систему є навчання через інструкції. Цей підхід базується на використанні попередньо визначеної мови для спілкування з системою щодо різних концепцій, визначених у цій мові. Багато систем PbD, наприклад, використовують обробку природної мови, щоб дозволити користувачам взаємодіяти з системою природним, інтуїтивним чином. Тут мовою для спілкування з роботом є фактична людська розмовна мова, хоча система часто налаштована на розуміння лише невеликої частини розмовної мови, обмеженої кількома простими концепціями. Були реалізовані й інші системи, які використовують людську мову у формі тексту. Наприклад, вчені використовували письмові інструкції, щоб повідомити робота про те, як виконати якесь завдання високого рівня. Хоча цей підхід схожий на використання розмовного діалогу, він використовує інший метод використання природної мови для взаємодії з системою. Інші ж використовували різні способи взаємодії з системою для надання інструкцій, такі як графічні інтерфейси.

Системи передачі знань.

Остання область дослідження, представлена в цьому розділі, стосується реальних систем або фреймворків, спеціально розроблених для передачі знань. З огляду на конкретне представлення знань, низка інших проектів розглядала проблему представлення, зберігання та передачі знань між різними роботизованими системами.

SIARAS

У проекті SIARAS (Перевірка та складання реконфігурованих систем автоматизації на основі навичок), Розроблено систему для сприяння автоматичному переналаштуванню систем автоматизації. Це зроблено через потребу в легких (з низькими накладними витратами) процесах для задоволення поточних виробничих потреб. Компоненти системи розроблені як для представлення навичок та параметрів, так і для відображення технологічного

процесу виконується за допомогою певної онтології. Сервер навичок призначений для того, щоб допомогти оператору-людині зіставити вимоги процесу з представленнями в базі даних.

ROSETTA

Ще один проект, що працює над розгортанням роботів у виробничому середовищі, в рамках проекту ROSETTA (керування роботами для кваліфікованого виконання завдань у природній взаємодії з людьми; на основі автономності, накопичених знань та навчання). Цей проект має на меті розробити промислові роботизовані системи, придатні для роботи поруч із людьми та співпраці з ними у виробничому процесі. Одним із важливих аспектів їхнього підходу є сховище навичок, де навички можуть обмінюватися між системами. Методи машинного навчання застосовуються до взаємодії з людьми для покращення цих навичок з часом.

RoboEarth.

RoboEarth – це проект, спрямований на створення глобального сховища всіх знань, що стосуються робототехнічних систем, не лише інформації, що стосується конкретних дій, але й усього, що потрібно роботу для функціонування в динамічному реальному середовищі, включаючи відповідну інформацію про навколишнє середовище, карти, моделі об'єктів та семантичну інформацію. Архітектура організована у три рівні, верхній рівень – це глобальна база даних, яка діє як інформаційний сервер, другий рівень містить апаратно-незалежні функції, такі як розпізнавання дій та семантичне відображення, а нижній рівень складається з реалізацій, специфічних для роботів. Представлення та обробка знань здійснюється системою KNOWROB.

2. ПРОЄКТНА ЧАСТИНА

2.1. Система передачі знань у робототехніці

Стан робототехніки сьогодні розвивається вражаючими темпами. Потреба в рішеннях для робототехніки та автоматизації продовжує зростати в багатьох різних галузях промисловості, від виробництва та польової робототехніки до охорони здоров'я та побутового обслуговування. І рішення, що пропонуються для цих потреб, також продовжують удосконалюватися. Оскільки робототехніка займає все більшу роль у цих різних галузях та сегментах, а потреби та взаємодія стають більш різноманітними та складними, самі системи робототехніки та автоматизації стають більш різноманітними та складними. Зі зростанням цієї складності зростає і потреба у розробці методів точного моделювання цих систем. Без відповідних моделей для систем стає все важче характеризувати ці системи та гарантувати їхню надійність та безпеку. Ця потреба йде пліч-о-пліч з необхідністю покращення стандартизації цих практик у галузі робототехніки. Галузі робототехніки та автономних систем ще багато чого потрібно зробити з точки зору стандартизації, хоча докладаються зусилля для просування у формулюванні стандартів. У цьому випадку також можна отримати багато переваг, максимально стандартизуючи моделювання цих складних автономних систем. У цьому розділі ми пропонуємо використовувати мову системного моделювання (SysML).

Як життєздатна мова для моделювання автономних систем та робототехніки. Як описова графічна мова моделювання, вона має низку особливостей, що роблять її привабливою для використання у складних системах. Це також стандартизована мова моделювання з ретельною формальною специфікацією, що потенційно спрощує спілкування та обмін інформацією щодо цих систем, а також створювати контрольні показники та порівнювати системи одна з одною.

2.2 Мова системного моделювання (SysML).

Мова системного моделювання, або SysML— це стандартизована мова графічного моделювання, розроблена для моделювання процесів у контексті системної інженерії. Зокрема, вона була розроблена для специфікації, аналізу та верифікації складних фізичних систем.

SysML базується на популярному стандарті UML (Уніфікована мова моделювання). UML — це узагальнена мова моделювання, розроблена спеціально для програмної інженерії, і хоча вона застосовувалася в системній інженерії протягом кількох років, існував ряд труднощів у моделюванні реальних систем за допомогою програмно-специфічної UML. Було запропоновано рішення у вигляді SysML, яка була представлена не стільки як нова мова для опису систем, скільки як розширення або модифікація існуючої специфікації UML. Рис. 2.1 ілюструє цей зв'язок.

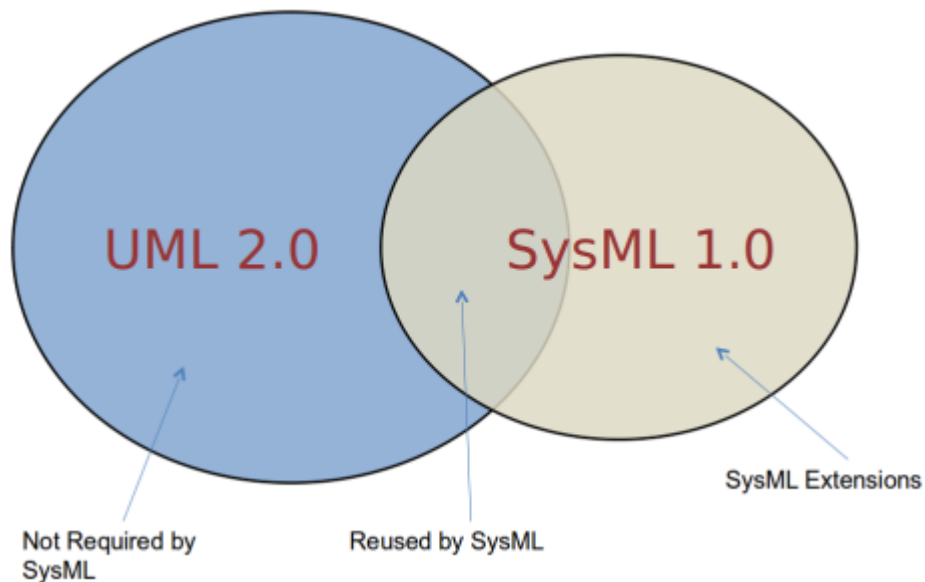


Рисунок 2.1 - Зв'язок між UML та SysML.

SysML запозичує значну частину своєї структури з UML. Він складається з набору різних діаграм (які використовуються для моделювання різних аспектів системи). У той час як UML містить тринадцять різних діаграм моделювання,

SysML містить лише дев'ять. Ці зв'язки показано на рис. 2.2. Зокрема, з дев'яти різних можливих діаграм, що використовуються в моделюванні систем за допомогою SysML, лише дві є унікальними для SysML (діаграма вимог та параметрична діаграма). Інші або модифіковані з діаграм UML, або перенесені безпосередньо зі специфікації UML.

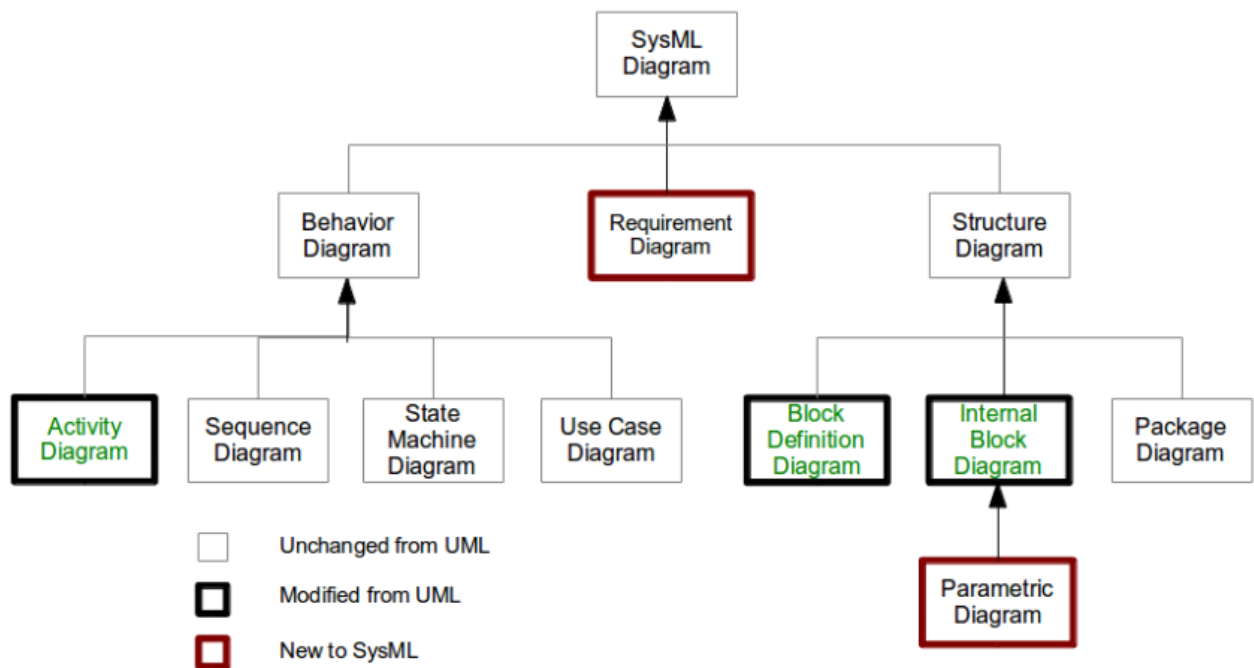


Рисунок 2.2 - Діаграма зв'язків між UML та SysML.

Дев'ять діаграм моделювання, доступних у специфікації SysML, можна згрупувати у чотири різні категорії, загальновідомі як чотири стовпи SysML: структурні діаграми, діаграми поведінки, діаграми вимог та параметричні діаграми. Нижче наведено короткий опис цих категорій та діаграм моделювання, які вони представляють.

Структурні діаграми.

Базовою структурною одиницею SysML є блок. Структурні діаграми описують статичні зв'язки між блоками в системі. Три структурні діаграми в SysML - це діаграма визначень блоків, внутрішня блок-діаграма та діаграма пакета.

Блок-діаграма визначення (BDD) — це основна діаграма в SysML. Вона визначає структурну ієрархію системи, а також її компоненти. Вона схожа на діаграму класів UML.

Внутрішня блок-схема (IBD) описує внутрішню структуру заданого блоку в системі. Це перевизначення складеної структурної діаграми з UML.

Діаграма пакета використовується для організації моделі та визначає її структуру. Вона залишається незмінною порівняно зі специфікацією UML.

Діаграми поведінки описують динамічні зв'язки між блоками в системі. Чотири діаграми поведінки - це діаграми варіантів використання, діаграми послідовностей, діаграми активності та діаграми машин станів.

Діаграма варіантів використання надає загальний опис взаємодії між системами. Вона не змінилася порівняно зі специфікацією UML.

Діаграма послідовностей описує взаємодію між частинами системи. Вона не змінилася порівняно зі специфікацією UML.

Діаграма діяльності описує потік даних та контроль між діями.

Діаграма машин станів описує переходи станів та дії, які система виконує у відповідь на події. Вона не змінилася порівняно зі специфікацією UML.

Діаграма вимог описує системні вимоги, ієрархії вимог та зв'язки, які задовольняють або підтверджують ці вимоги.

Параметрична діаграма описує кількісні обмеження на властивості системи, такі як продуктивність, надійність та масові властивості, і служить засобом інтеграції моделей специфікацій та проектування з моделями інженерного аналізу.

Робота, пов'язана з SysML.

Протягом багатьох років було запропоновано низку різних методів моделювання робототехнічних систем. Мережі Петрі були популярним вибором завдяки своїм математичним властивостям. Лайонс та ін. запропонував використовувати портові автомати для моделювання роботи як засіб розробки корисної мови програмування роботів. Косецька запропонували використання структури скінченних автоматів для моделювання систем дискретних подій для

автономних систем. Дантам та ін. визначив нову формальну граматику, відому як граматика руху, як засіб моделювання системи та її можливостей. Спільним для цих підходів є те, що основні представлення для кожного сумісні з формальними методами, що дозволяє використовувати їх для формулювання конкретних тверджень про деякі властивості систем, які вони моделюють (доцільність тощо). На жаль, ці методи також мають кілька недоліків. Незрозуміло, чи всі вони добре масштабуються для дуже складних систем. Деякі з цих підходів також не здатні повністю змоделювати багато робототехнічних систем, оскільки багато систем мають справу не лише з роботом, але й з взаємодією з навколишнім середовищем та людиною. Щоб вирішити деякі з цих проблем, деякі роботи були спрямовані на спробу поєднати ці підходи до моделювання з описовими моделями.

SysML, з іншого боку, моделює системи за допомогою описових графічних діаграм, які відображають структуру та поведінку системи з різних точок зору. Це формальна мова, оскільки описи повністю визначені, хоча вона не має тих самих властивостей, що й інші мови моделювання, згадані раніше. Однак вона має деякі переваги, зокрема здатність охоплювати широкий спектр системних абстракцій. SysML використовувався для моделювання низки різних застосувань у багатьох різних галузях промисловості. Популярні приклади варіюються від очищення води до проектування супутникових систем.

Хоча SysML ще не здобув широкого впровадження або популярності в робототехніці, існує кілька груп, які намагалися інтегрувати мову моделювання SysML у проектування своїх систем. Прикладами є робота Чханіяри та ін. У цій роботі автори представляють застосування SysML до моделювання складних космічних роботизованих систем, що використовуються в місіях з обслуговування супутників. Вони стверджують, що SysML є хорошим варіантом завдяки автоматизованій перевірці вимог та трасуванню моделі, можливим за допомогою SysML, що може бути використано для зменшення витрат та часу в системній інженерії. Рахман та ін. наводить аналогічні аргументи на користь використання SysML у проектуванні мобільних роботів. У цій роботі програмне

забезпечення для мобільного робота розроблено з використанням підходу системної інженерії на основі моделей (MBSE). Стверджується, що використання MBSE (з SysML як конкретною реалізацією) може дозволити створення багаторазових програмних модулів для програмування робота, що дозволяють платформонезалежне проектування та скорочують час розробки. Хоча ця робота демонструє, що SysML може бути застосований до деяких роботизованих систем з різним ступенем успіху, вони не описують, як його можна застосувати до інших областей застосування роботів.

Існує ряд важливих міркувань, які слід враховувати, розглядаючи вибір відповідної мови моделювання для таких систем, як робототехніка.

У вступі ми вказали на два важливі моменти в подальшому розвитку моделювання в робототехніці: здатність точно моделювати дедалі складніші системи та стандартизацію такого підходу. SysML унікально підходить для обох цих завдань. Як універсальна графічна мова моделювання, розроблена зокрема для систем, розробник може вибрати рівень абстракції та, таким чином, видиму складність системи в самому проектуванні моделі. Ця здатність вибирати відповідний рівень абстракції залежно від проблеми або системи, з якою мається на увазі, є важливою, оскільки вона також полегшує спілкування про систему з іншими. Це може бути особливо важливим при роботі з великими системами, оскільки зацікавленим сторонам часто потрібно спілкуватися щодо різноманітних компонентів системи. Використання SysML також буде кроком до стандартизації моделей у робототехніці.

Коли UML вперше представили розробникам програмного забезпечення, він швидко перетворився на стандарт для моделювання проєктів розробки програмного забезпечення. Так само SysML стає дедалі популярнішим як засіб опису системних моделей.

Верифікація та валідація.

Одне з звинувачень, висунутих щодо використання SysML у робототехніці, полягає в тому, що він у деяких аспектах поступається підходам, які можуть застосовувати формальні методи для перевірки певних властивостей

системи. Щоб правильно розглянути це твердження, спочатку визначимо верифікацію та валідацію у правильному контексті. У контексті системної інженерії, верифікація та валідація (V&V) зазвичай розуміється як таке:

- Верифікація — встановлення істинності відповідності між системою та її специфікацією.
- Валідація – встановлення придатності системи для її цільового призначення.

Неофіційно, верифікація розглядається як питання «Чи правильно будується система?», тоді як валідація запитує: «Чи правильно будується система?». Ці питання фундаментально пов'язані зі специфікацією та вимогами системи. Формальна верифікація, з іншого боку, пов'язана з доведенням правильності деякої властивості за допомогою формальних методів. Що стосується верифікації та валідації вимог, SysML здатний полегшити це завдяки специфікаціям мови та наданню правильної діаграми вимог. Слід також зазначити, що часто саме програмний інструмент, який підтримує моделювання SysML, визначає, які функції доступні, а не сама мова. Фактично, багато пакетів пропонують інструменти для верифікації та валідації вимог разом із моделюванням та генерацією коду, серед іншого. Формальна верифікація - це інша справа. Хоча представлення SysML за своєю суттю не підтримує формальні методи верифікації, було проведено багато роботи, щоб обійти це обмеження, використовуючи при цьому сильні сторони SysML. Наприклад, була проведена робота з забезпечення ймовірнісної верифікації моделей SysML. Також було проведено роботу з перетворення моделей SysML в інші представлення з метою формальної верифікації.

Автоматична генерація коду

Автоматична генерація коду – це ще одна важлива функція, доступна в багатьох програмних інструментах, що підтримують моделювання за допомогою SysML. Автоматична генерація коду може мати кілька різних форм. Багато інструментів підтримують пряме або зворотне проектування моделі. У прямому

проектуванні (у цьому контексті) код генерується на основі моделі SysML, яка визначається за допомогою різних використовуваних діаграм.

Під час інженерії код генерує (або частіше оновлює) модель на основі змін, що відбуваються в коді. Інший варіант автоматичної генерації коду відомий як циклічна інженерія. У циклічній інженерії система дозволяє вносити зміни як до моделі, так і до кодової бази, а також оновлює код або модель на основі внесених змін.

Можливість моніторингу та оновлення моделі таким чином має велике значення для робототехніки та великомасштабних систем загалом. Одним із значних потенційних джерел помилок може бути нехтування оновленням та поширенням змін по всій системі з дуже великою кодовою базою та значною кількістю моделей (або однією великою, дуже складною моделлю). Поширення змін вручну по всій складній системі займає багато часу та є схильним до помилок. Це може допомогти підтримувати системні моделі актуальними та точними для перевірки або моделювання системних завдань.

2.3 Фреймворк на основі SysML для передачі знань у робототехніці

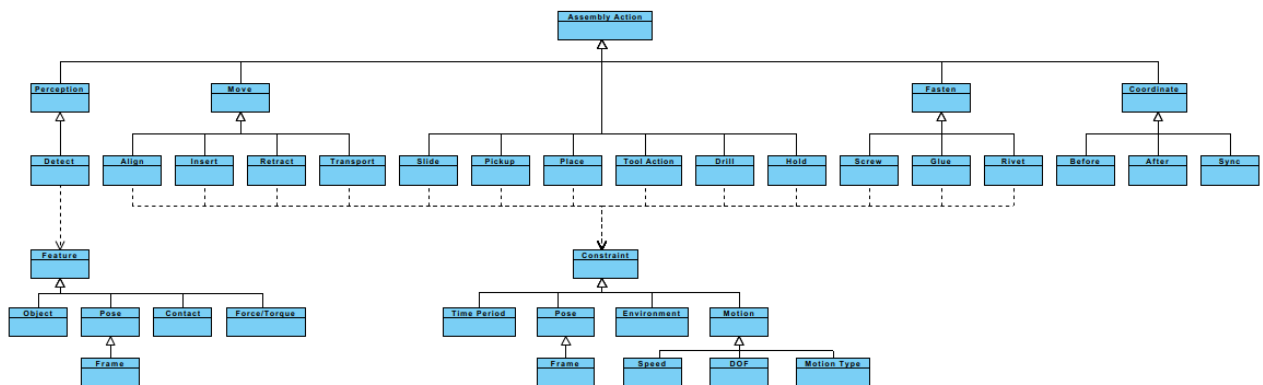


Рисунок 2.3 - Таксономія можливостей для домену складання.

Для ефективного моделювання завдань маніпулювання роботами в заданій області важливо мати можливість охопити широкий спектр можливостей системи. З цією метою ми визначаємо таксономію, залежну від області

застосування, для опису завдань маніпулювання роботами. Щоб продемонструвати, як це працюватиме для конкретного випадку використання, ми використовуватимемо область виробництва та складання як приклад опис деяких переваг цього підходу. Рисунок показує таксономію, визначену для області виробництва та складання роботів. Одним з основних припущень є те, що таксономія визначена таким чином, щоб вона могла моделювати всі можливі завдання та цілі в області. Це важлива основа для твердження, що фреймворк здатний узагальнюватися для всіх завдань у заданій області. У цьому випадку таксономія визначається за допомогою блок-схеми SysML для опису цих системних можливостей.

Примітиви навичок, навички та завдання.

Сама таксономія складається з кількох різних компонентів. Верхня половина таксономії на рисунку складається з набору так званих примітивів навичок. Примітиви навичок визначаються як основні, атомарні дії, які робот здатний виконувати в предметній області, та представляють узагальнену здатність. Прикладами є примітив навичок транспортування для переміщення робота в середовищі, а також примітив навичок гвинтування для виконання завдань загвинчування за допомогою кінцевого ефектора (див. рис. 2.4).

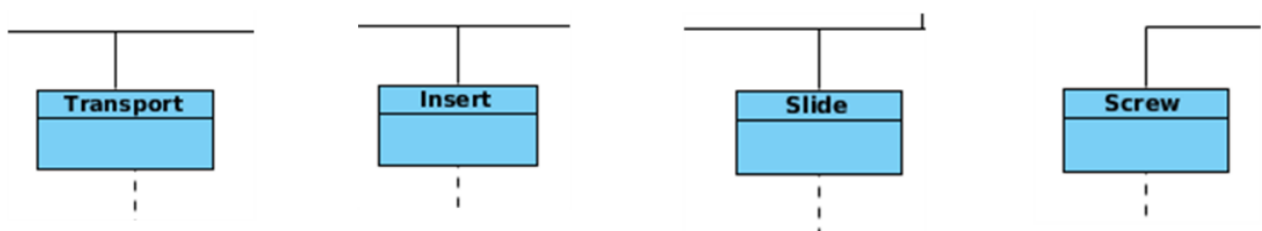


Рисунок 2.4 - Приклади різних примітивів навичок.

У термінології фреймворку, навичка визначається як абстракція примітивів навички. Одним із прикладів є навичка Fasten, яку можна розглядати як абстракцію кількох різних типів примітивів навичок в області складання (див. рис. 2.5)

Завдання стосуються концепцій високого рівня, які визначаються як сукупність навичок. Прикладом цього може бути складання колеса для моделі транспортного засобу.

Загальну мету або завдання тоді можна визначити з точки зору послідовності завдань.

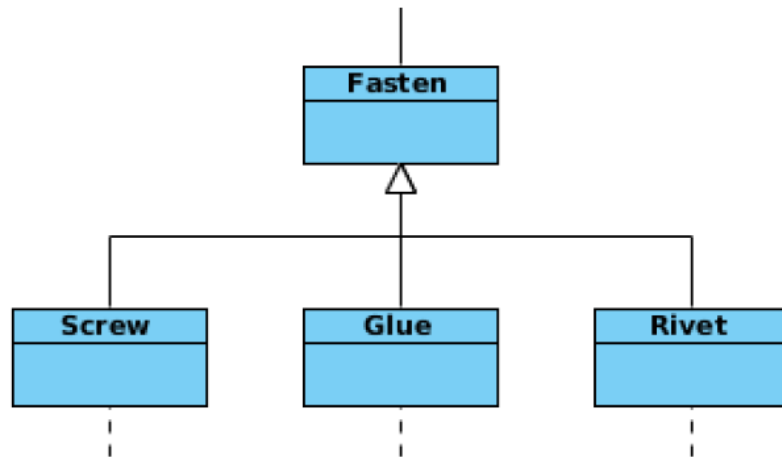


Рисунок 2.5 - Навичка «Закріпити» поділена на кілька різних примітивів навичок

Така ієрархія природно вписується в простір взаємодії, як видно на рис. 2.6. Обмеження.

Нижня половина таксономії представляє параметри або обмеження, що використовуються для визначення примітивів навичок (малюнок 10.) Примітиви навичок визначаються з точки зору параметрів, що накладаються на них системними вимогами до виконання дії. Наявність способу визначення цих вимог і параметрів є важливою для точного моделювання системи.

Наприклад, Період часу може використовуватися примітивом навички утримання (Hold) для опису того, як довго щось слід утримувати в певній позі, а також для позначення того, що певна дія має бути виконана за певний проміжок часу. Такі дії, як Вирівнювання (Align) та Транспортування (Transport), часто виконуються стосовно певної цільової пози (Pose).

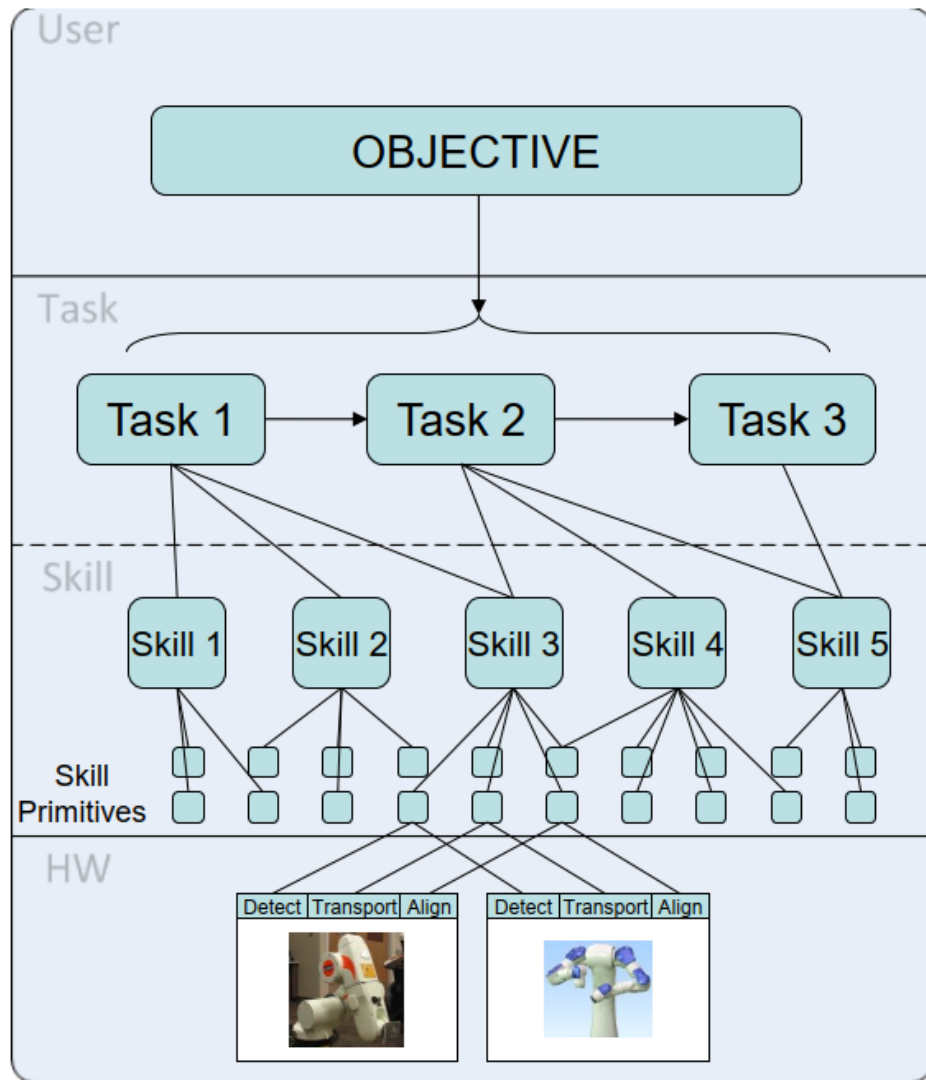


Рисунок 2.6 - Ієрархія декомпозиції завдань.

Приклади обмежень середовища включають обмеження робочого простору, необхідність слідування певній поверхні та обмеження, за яких планування руху в середовищі здійснюється за наявності перешкод. Обмеження руху стосуються руху робота та включають Швидкість (Single Fought), Глибочину свободи (Standing Scope) та Тип руху (Motion Type), які можна використовувати для визначення того, як робот повинен слідувати траєкторії з певним профілем швидкості, наприклад, у зварювальному завданні (This path of structure).

Розглянемо, як робот взаємодіє зі своїм оточенням. Роботи взаємодіють зі світом, працюючи над досягненням певної мети, наприклад, побудувати крило літака. Ці цілі можна розбити на послідовність завдань, які необхідно виконати

для успішного досягнення мети, наприклад, скріплення двох листів металу разом за допомогою болта. Завдання можна додатково розбити на навички, необхідні для виконання завдання, такі як нарізання різьби болта, та те, що називається примітивами робочих навичок, які є простими, специфічними для робота діями, такими як закриття захоплювача навколо об'єкта.

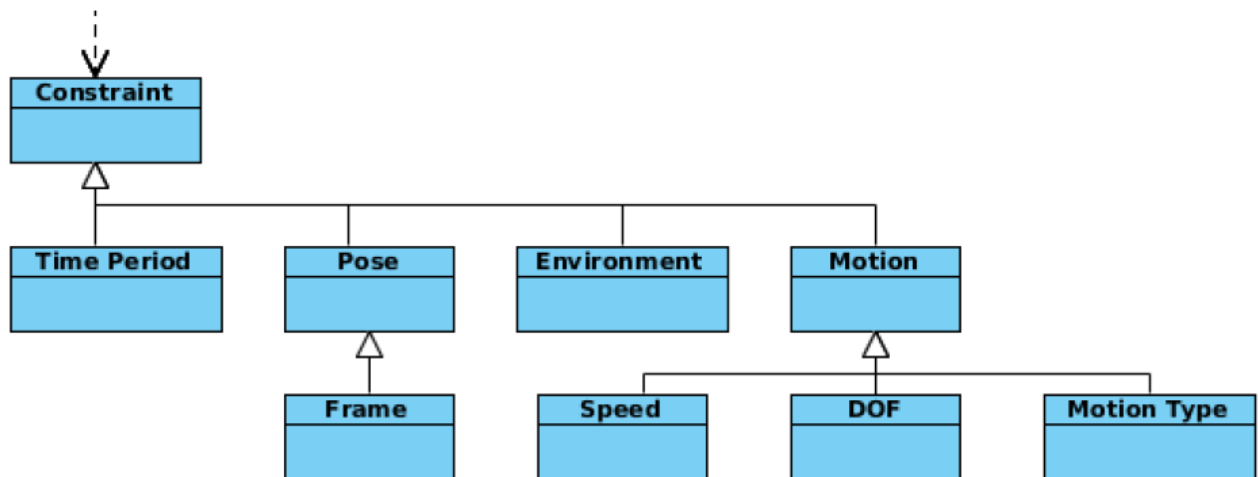


Рисунок 2.7 - Параметри для примітивів навичок.

Сприйняття.

Набір необхідних можливостей робота повинен включати всі дії, необхідні роботу для виконання завдання, включаючи перцептивні можливості. Цей тип представлення перцептивних дій є однією з функцій, якої бракує багатьом системам, згаданим у розділі про пов'язану роботу.

Для виконання завдання складання потрібно багато різних типів сприйняття, і таксономія здатна полегшити цю взаємодію. Примітив навички виявлення (рис. 2.8) є, по суті, інтерфейсним модулем для цих різних дій сприйняття. Обмеження або параметри цього примітиву навички визначаються як ознаки, що мають бути виявлені, і включають Об'єкт, Позу, Контакт та Силу/Крутний момент.

Ці типи ознак обробляють різні типи сприйняття, необхідні для складання. Це охоплює датчики та алгоритми, пов'язані з оцінкою пози та виявленням

об'єктів, такі як камери та 3D-сканери, а також інші методи пошуку та локалізації потрібних об'єктів для маніпулювання.

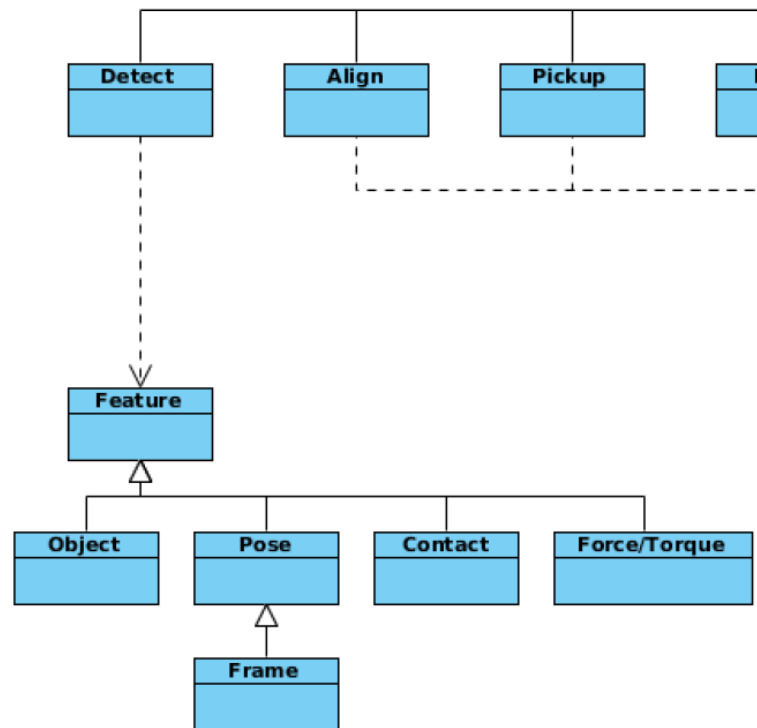


Рисунок 2.8 - Примітив виявлення навичок.

Інші датчики, що представляють інтерес, які у цьому контексті корисними є контактні датчики та датчики сили/крутного моменту для виявлення контакту в навколишньому середовищі та виконання складних, точних маніпуляційних завдань.

Контроль.

Так само важливою для виконання завдання складання, як і взаємодія з підсистемою сприйняття, є здатність таксономії взаємодіяти з підсистемою керування. За винятком «Виявлення» та «Координація», усі примітиви навичок взаємодіють з підсистемою керування, викликаючи певний контролер для виконання певної дії. «Вставка» викличе контролер для зміни положення кінцевого ефектора, «Слід» керуватиме контактними взаємодіями на кінчику інструменту для ковзання заготовки вздовж певної поверхні. Існує багато різних типів контролерів, які потрібні системі для виконання складання.

Слід зазначити, що таксономія розроблена незалежно від реалізації. Так само, як транспорт не залежить від алгоритму планування шляху, який використовується для планування маршруту, він також не залежить від типу контролера, що використовується для керування цим шляхом.

Підсистема керування та підсистема сприйняття тісно інтегровані. У більшості випадків контролери, що викликаються примітивами навичок, у свою чергу викликатимуть функції сприйняття для забезпечення зворотного зв'язку з системою. Це стосується як керування із замкнутим циклом, так і керування із розімкнутим циклом, оскільки специфікація пози необхідна для ініціалізації навіть у випадку із розімкнутим циклом. Наприклад, у випадку із замкнутим циклом функція Align залежатиме від підсистеми сприйняття для визначення пози робота, коли він досяг своєї мети. Система сприйняття часто надаватиме контролеру кінцеві умови, які, у свою чергу, сигналізуватимуть системі про завершення завдання і можна спробувати виконати нове.

Інші функції.

Координація.

Однією з важливих можливостей для забезпечення гнучкості та надійності для представлення всіх типів завдань виробничого складання є здатність інкапсулювати координацію кількох систем, що працюють узгоджено для досягнення виробничої мети. В ідеалі представлення мало б бути здатним обробляти роботів різних конфігурацій та специфікацій (наприклад, як роботи з однаковими специфікаціями, що виконують різні ролі, такі як ідентичні роботи на протилежних сторонах конвеєрної лінії, так і роботи з абсолютно різними специфікаціями, такі як послідовні маніпулятори, що працюють з паралельними платформами або мобільними платформами).

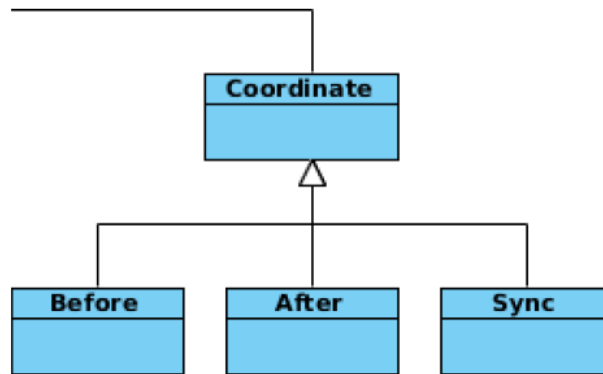


Рисунок 2.9 - Координаційні навички

Використання інструменту.

Ще один важливий аспект, який слід враховувати при роботі роботів у виробничих середовищах, – це переконатися, що представлення може враховувати можливість використання інструментів для виконання бажаного завдання. Інструменти необхідні для виконання найскладніших виробничих завдань. Ця важлива вимога в нашій системі виконується двома способами. По-перше, параметризація обмежень дозволяє задавати специфікації в різних системах відліку, що позначаються обмеженням системи відліку. Система відліку, пов'язана з певним інструментом, може бути задана щоразу, коли інструмент використовується роботом.

Важливою функцією, необхідною роботу для використання інструменту, є здатність активувати (або іншим чином активувати та використовувати) інструмент. З цієї причини в таксономії існує примітив навички «Дія з інструментом». Це дозволяє системі функціонально керувати інструментом, наприклад, активувати дріль або приводити в дію механізм нанесення клею на інструмент для склеювання.

3.5.5.3 Мобільність

Щоб запропонована структура охоплювала можливості різних роботизованих систем, необхідно також враховувати концепцію мобільності. У контексті завдання складання існує кілька різних типів роботів, які можуть

використовувати цю навичку, від простих мобільних платформ до повноцінних мобільних маніпулюючих систем.

Таксономію, яку обговорювали в попередніх розділах, також можна використовувати для моделювання мобільності, використовуючи примітиви навичок Transport та Hold. Використовуючи іншу параметризацію, Transport можна використовувати для переміщення мобільної платформи з однієї конфігурації в іншу, так само як він може переміщувати кінцевий ефектор (або якийсь об'єкт у кінцевому ефекторі) маніпулятора в потрібне місце. Аналогічно, Hold можна використовувати в тому ж контексті, що й маніпулятор, надаючи роботу (в цьому випадку мобільній платформі) інструкцію утримувати певну конфігурацію протягом певного часу.

2.4 Абстракція навичок

Часто знання, що існують у робототехнічних системах, представлені таким чином, що їх не можна використовувати повторно, і багато зусиль витрачається на повторення цих знань для нових систем або нових конфігурацій існуючих систем. Ця проблема особливо актуальна в галузі промислового виробництва. У цьому розділі ми представляємо платформу моделювання завдань на основі SysML, яка допомагає зробити знання стійкими в різних процесах і системах шляхом абстракції навичок робота. Ми представляємо цю платформу та обговорюємо її потенційні переваги. Ми також описуємо кілька експериментів, які були проведені для демонстрації ефективності підходу до узагальнення моделей для різних завдань і платформ.

Тут ми розглядаємо проблему негнучкості в сучасних виробничих процесах, що включають робототехніку та програмування роботів, і пропонуємо рішення, в якому наша система моделювання завдань на основі SysML використовується для моделювання можливостей робота в заданій області застосування. Використовуючи цю систему для моделювання навичок робота на вищому рівні абстракції, ми маємо намір показати, що ми можемо внести більше

гнучкості у виробничий процес, розглядаючи програмування та взаємодію роботів на рівні завдань.

Для ефективного моделювання завдань маніпулювання роботами в заданій області важливо мати можливість охопити широкий спектр можливостей системи. З цією метою ми визначаємо таксономію, залежну від області застосування, для опису завдань маніпулювання роботами. Для ясності ми використовуватимемо область виробництва та складання як приклад для опису деяких переваг цього підходу. Рис. 2.10 показує таксономію, визначену для домену виробництва та складання роботів, яку було детально обговорено в попередньому розділі. Таксономія визначена таким чином, що вона здатна моделювати всі можливі завдання та цілі в предметній області.

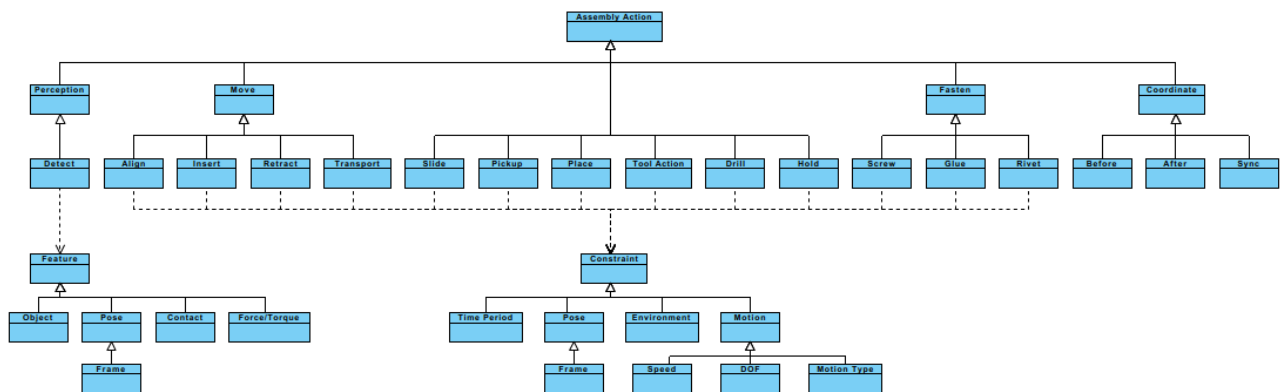


Рисунок 2.10 - Таксономія можливостей для домену складання.

Ми припускаємо, що використання такого узагальненого високорівневого представлення дозволяє застосовувати ці примітиви навичок до широкого кола проблем, а також описувати плани та цілі завдань незалежно від платформи. У наступних розділах ми описуємо набір експериментів, які демонструють, що за допомогою цієї системи моделювання завдань ідентичні примітиви навичок можна використовувати в різних комбінаціях для виконання широкого спектру завдань, а також ідентичні описи завдань можна використовувати для виконання одного й того ж завдання на кількох різних платформах роботів.

Мотивуючий приклад

У цьому розділі детально пояснюється, чому приклади завдань є гарною демонстрацією фреймворку GTax. У цьому обговоренні метою є навчитися збирати модель літака, як показано рис. 2.11 ,використовуючи складові частини, показані на зображенні ліворуч.

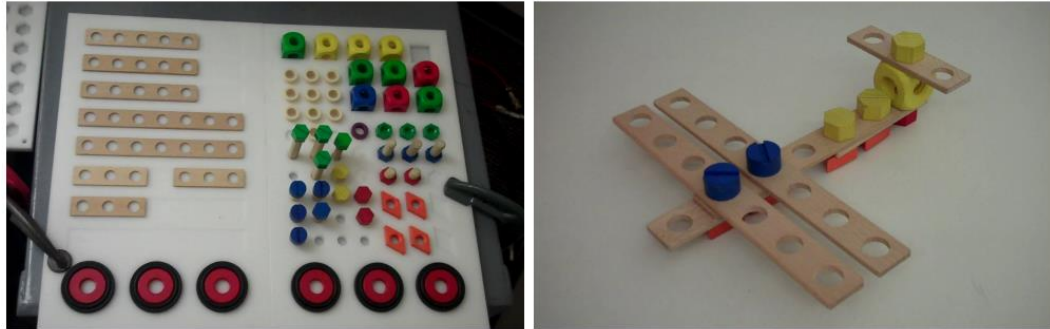


Рисунок 2.11 - Приклад завдання складання

Поставлене тут питання стосується того, які основні можливості потрібні роботу для виконання цього простого завдання. Щоб визначити це, ми простежимо кожен крок, пов'язаний зі складанням моделі.

По-перше, перш ніж щось можна буде зробити з самими деталями, необхідно розробити певні засоби, за допомогою яких робот зможе виявляти, де знаходяться деталі, середовище та розрізняти різні компоненти, щоб знайти першу деталь, яку потрібно помістити у пристосування для складання. Після того, як правильну деталь знайдено, робот повинен визначити її положення, щоб нею можна було маніпулювати. Після визначення деталі та завершення оцінки положення робот повинен переміститися в таке положення, щоб він міг підняти деталь та транспортувати її у правильне місце у пристосуванні. Потім робот вирівнюється з деталлю, щоб її можна було захопити. На цьому етапі роботу потрібно підняти деталь; або відкрити інструмент, який використовується для захоплення деталі, вставити інструмент у таке положення, щоб його можна було захопити, замкнути інструмент навколо деталі до відповідної ширини захоплення та відвести інструмент назад у вихідне положення.

Щоб визначити, куди тепер помістити деталь, робота знову потрібно визначити її розташування на пристосуванні та виконати оцінку положення, щоб визначити правильне розташування деталі в пристосуванні. Маючи деталь у руках, робот може транспортувати її до відповідного місця в пристосуванні. Тут робот знову вирівнюється з положенням розміщення, потім вставить деталь у потрібне положення, відкриє інструмент, щоб звільнити деталь, та відведе її у вихідне положення. Перша деталь тепер на місці. Цей процес повторюється для решти деталей, які вставляються безпосередньо в пристосування.

Наступні частини виконуються за аналогічною схемою, за винятком того, що замість визначення пози та вирівнювання щодо кріплення це робиться стосовно частини на кріпленні. Наприклад, для частини B5 (дошка з 3 частин) це робиться стосовно помаранчевих гайок, попередньо розміщених у кріпленні. Це продовжується для наступних 5 частин.

Після розміщення наступну деталь потрібно вкрутити в збірку на пристосуванні. Деталь витягується таким самим чином, як і інші деталі. Після цього робот повинен визначити місце розташування та вісь, вздовж якої потрібно вкрутити гвинт. Потім робот може транспортувати гвинт до кінцевого місця, вирівнювати його по осі гвинта та вставляти гвинт, виконуючи рух вкручування. Щоб визначити, чи завершено вкручування, робот визначає крутний момент кінця, вище деякого порогу. Цей процес повторюється на решті деталей, доки модель остаточно не буде зібрана.

З цього прикладу видно, що для виконання завдання потрібно лише кілька основних можливостей (Виявлення, Транспортування, Керування інструментом, Вирівнювання, Вставка, Відведення) та параметри, необхідні для визначення цих можливостей (Поза, Ширина захоплення, Вісь вирівнювання, Крутний момент, Поріг крутного моменту).

Абстракція завдання.

У цьому розділі ми описуємо експерименти, розроблені для демонстрації двох різних аспектів абстракції з використанням раніше представленої структури моделювання завдань. Перший експеримент має на меті продемонструвати

абстракцію для різних завдань, або те, що один робот, використовуючи ті самі примітиви навичок, змодельовані у структурі, здатний успішно виконувати два різні завдання.

У цьому експерименті ми використовуватимемо одного й того ж робота для виконання двох різних завдань зі складання робота. В обох завданнях робот буде відповідати за складання іграшкової моделі з використанням невеликих дерев'яних деталей. Ці деталі, показані на рисунку 17, включають дерев'яні шурупи, шайби, гайки та конструкційні блоки.

Першою обраною моделлю була модель літака, зображена на малюнку 17б. Як обговорювалося в попередньому розділі, це було обрано, оскільки завдання вимагало використання багатьох примітивів навичок, зазначених у моделі таксономії складання, а складання могло бути фізично виконано роботом простим способом. Деталі, що використовуються для складання літака, взяті з модельного набору Baufix і являють собою прості, яскраво забарвлені дерев'яні будівельні деталі, такі як гвинти, шайби, блоки, гайки та дошки. Модель літака була побудована з 17 окремих деталей. Збірка моделі літака виготовляється за допомогою одного маніпулятора та статичного пристосування, спеціально розробленого для допомоги роботу в цьому завданні складання літака.

Самі завдання моделюються за допомогою діаграми послідовностей SysML. Діаграма послідовностей призначена для демонстрації організації цілі складання в послідовність дій, які необхідно виконати, та використовує примітиви навичок, описані в таксономію як дискретні кроки в послідовності. Приклад однієї з цих діаграм послідовностей можна побачити на рисунку 18. Кожне повідомлення від робота до пристосування або контейнера деталей являє собою екземпляр примітиву навичок (повідомлення 1 – це Detect, повідомлення 2 – Align тощо). Незважаючи на те, що таксономія розроблена для сфери виробництва та складання, самі примітиви навичок однакові та просто використовуються на деталях іграшок замість їхніх більш традиційних виробничих аналогів лише для демонстрації.

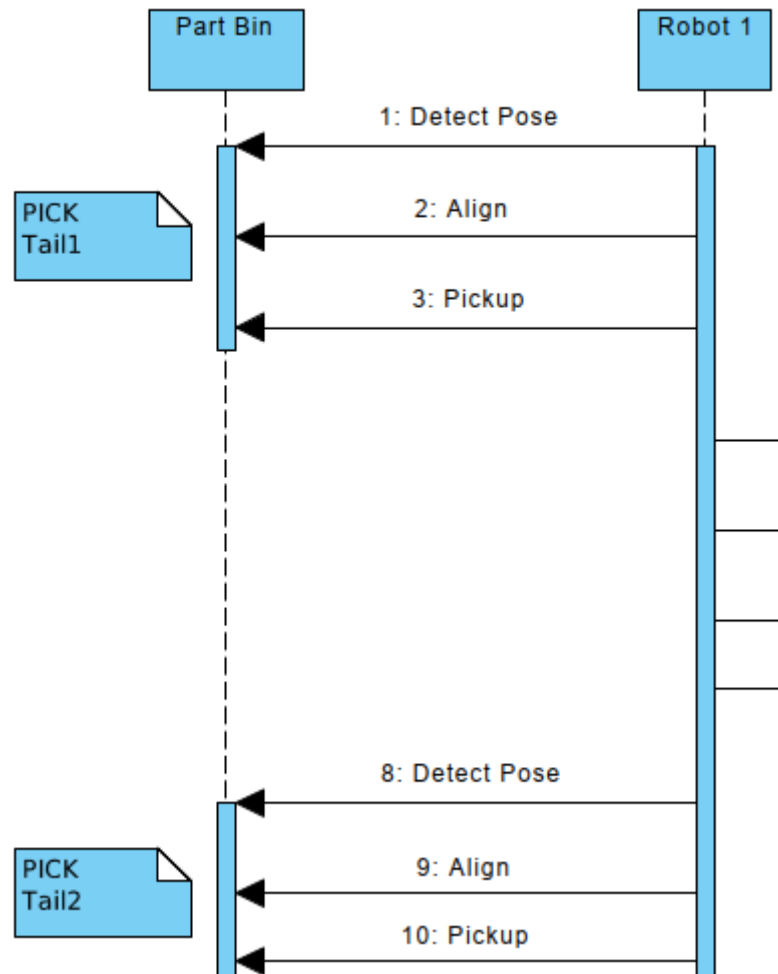


Рисунок 2.12 - Фрагмент послідовної схеми складання моделі літака.

Зауважте також, що хоча показано лише частину діаграми послідовності, повна діаграма складається з 258 дій або екземплярів примітивів навичок для виконання завдання складання.

Рис. 2.13 показує діаграму обмежень траєкторії для моделі літака, або вимоги до послідовності, які необхідно виконати для правильного складання літака. Цілі в кожній коробці можна досягти в будь-якому порядку (у випадку верхньої частини фігури порядок не має значення для найнижчих деталей, встановлених у кріплення), але всі цілі в кожній коробці повинні бути досягнуті, перш ніж можна буде спробувати досягти цілей у наступній коробці (деталі кріплення повинні бути розміщені до того, як будуть покладені частини корпусу та крила).

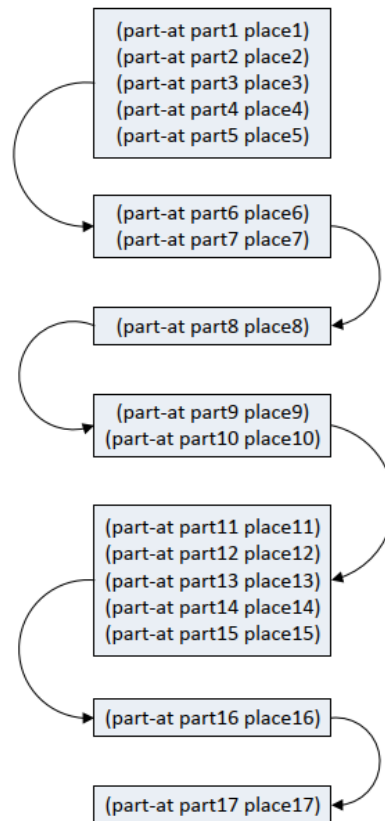


Рисунок 2.13 - Діаграма обмежень траєкторії для складання моделі літака.

Друге завдання, яке має виконати робот, — це складання моделі транспортного засобу, яку можна побачити на рис 2.14. Як і модель літака, модель транспортного засобу також є доречним вибором для цього прикладу, оскільки завдання вимагало використання багатьох примітивів навичок, зазначених у моделі таксономії складання. Це гарний приклад для використання разом із моделлю літака, оскільки, хоча обидві використовують подібні деталі та послідовності дій складання, самі можливі плани можуть набувати дуже різних форм. Ця модель побудована з використанням 23 дерев'яних конструкційних деталей.

Рис. 2.15 показує обмеження траєкторії для моделі транспортного засобу. Ці обмеження набувають іншої форми для цієї моделі порівняно з попередньою моделлю. У той час як попередня модель мала дуже чіткі області, де порядок дій не мав значення, тут є як вільні, так і суворі обмеження на порядок. У верхній

частині діаграми суворі обмеження спостерігаються між частинами {1,4,7}, {2,5,8} та {3,6,9}.



Рисунок 2.14 - Завдання складання моделі транспортного засобу, що використовується для демонстрації абстракції завдання

Тобто, частина 1 (болт) має бути розміщена перед частиною 4 (відповідною шайбою), які обидва мають бути розміщені перед деталлю 7 (шина), але доки цей порядок дотримано, не має значення, як це відбувається у переплетенні з іншими підвузлами (вільне обмеження). Пунктірний контур вказує на те, що всі ці кроки, в будь-якому порядку їх виконання, мають бути виконані до розміщення деталі 10.

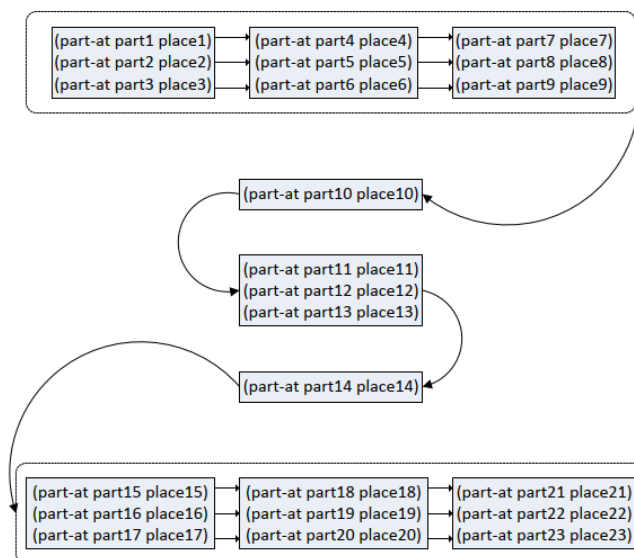


Рисунок 2.15 - Діаграма обмежень траєкторії для складання моделі транспортного засобу

3 СПЕЦІАЛЬНА ЧАСТИНА

3.1 Експериментальні результати

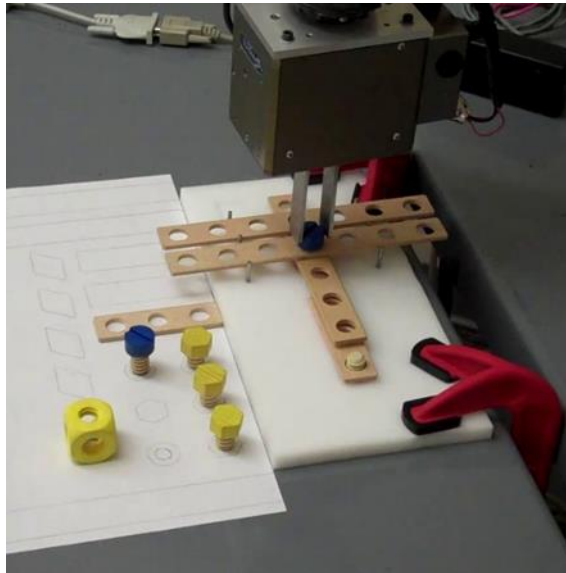


Рисунок 3.1 - Абстракція завдання – складання моделі літака роботом.



Рисунок 3.2 - Абстракція завдання – складання моделі транспортного засобу роботом.

У цьому експерименті було використано того самого робота (у цьому випадку KUKA KR5-sixx для виконання двох різних завдань. В описах завдань використовувалися ті самі абстрактні примітиви навичок, визначені в рамках

моделювання завдань. Рис. 3.1 і 3.2 показують, як робот успішно збирає кожную модель.

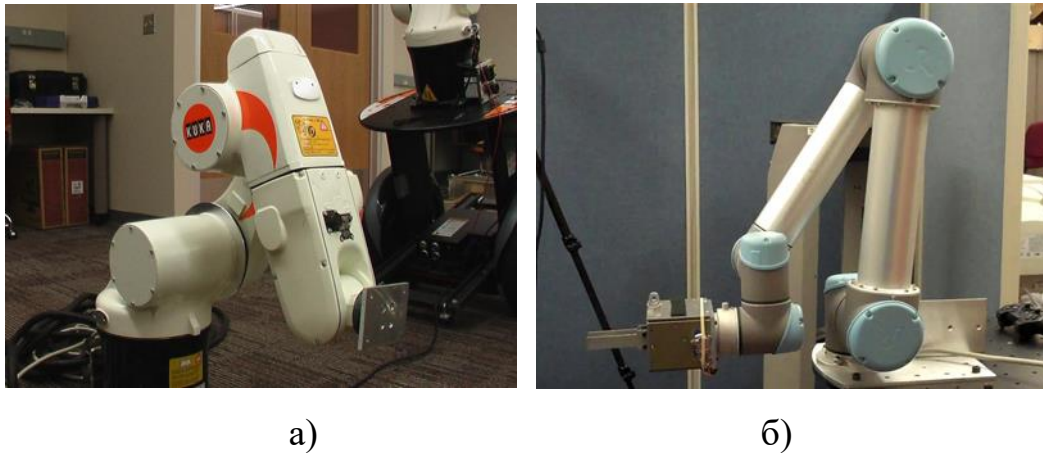


Рисунок 3.3 - Платформи, що використовуються для демонстрації абстракції платформи: а) Платформа 1 - KUKA KR5-sixx; б) Платформа 2 - Universal Robots UR5

У самому експерименті було кілька припущень. По-перше, ми припустили, що всі потрібні деталі в контейнері деталей видимі та доступні. Також припускалося, що сприйняття (виявлення об'єкта та оцінка пози) легкодоступне для системи та точне. Це моделювалося з використанням фактичних поз деталей для маніпулювання. Щоразу, коли Detect використовувався для визначення пози певного бажаного об'єкта для маніпулювання, замість отримання оцінки пози шляхом звернення до алгоритму оцінки пози на основі сприйняття, для цілей моделювання використовувалася доступ до істинної пози, визначеної до експерименту.

Абстракція платформи

Другий експеримент продемонструє абстракцію на різних платформах, або те, що один план завдання, описаний за допомогою примітивів навичок у фреймворку, може бути використаний на кількох різних платформах для виконання одного й того ж завдання.

У цьому експерименті ми використовуватимемо одне й те саме завдання складання робота на двох різних платформах роботів, щоб продемонструвати,

що абстракції, що використовуються в рамках моделювання завдань не залежать від платформи. Саме завдання є завданням складання моделі літака, представленим у попередньому розділі. На рис. 3.4 показано дві роботизовані платформи, що використовуються для демонстрації абстракції платформи. Перша платформа, яку ми використовуємо, це KUKA KR5-sixx . Це легкий промисловий маніпулятор, розроблений та призначений для виконання завдань легкого виробництва. KR5-sixx має шість ступенів свободи, корисне навантаження 5 кг та максимальну досяжність майже 1 метр.

Другий робот, використаний у цьому експерименті, – це платформа Universal Robots UR5. Цей робот аналогічно має корисне навантаження 5 кг, досяжність майже 1 метр і шість ступенів свободи, хоча конфігурація самого робота відрізняється від конфігурації KR5-sixx.

Для обох роботів маніпуляція здійснюється за допомогою паралельно-пластинчастого захоплювача Schunk.

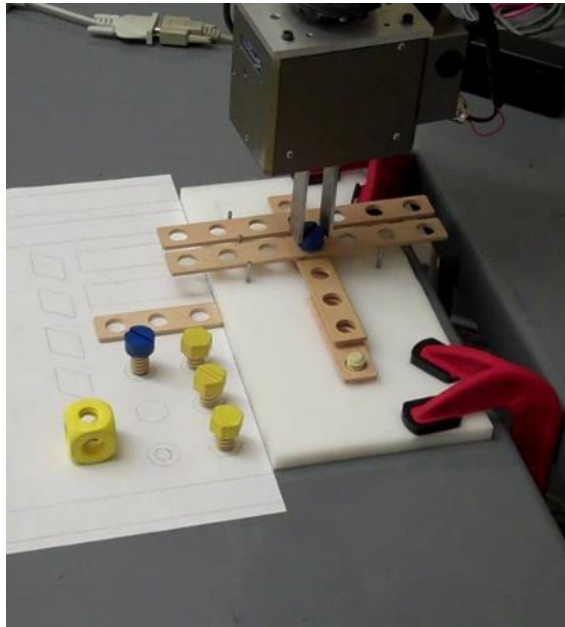


Рисунок 3.4 - Платформа 1 (KR5) виконує завдання складання.

Рис. 3.4 і 3.5 показують, як обидва роботи успішно збирають модель літака. Як описано в попередньому розділі, кожен робот виконував ідентичний опис завдання план (показано на діаграмі послідовності).

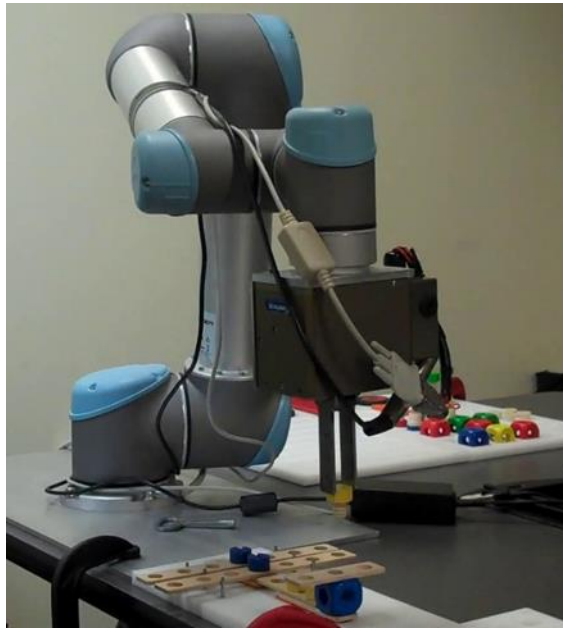


Рисунок 3.5 - Платформа 2 (UR5) виконує завдання складання.

Слід зазначити, що для успішної абстракції платформи робиться припущення, що для кожної платформи, яка буде використовуватися, існує повний набір реалізованих примітивів навичок, як описано в таксономії. Хоча це припущення вимагає певної роботи з реалізації перед виконанням, ця робота пропорційна кількості примітивів навичок у таксономії та повинна виконуватися лише один раз для будь-якої заданої моделі платформи. Після того, як примітиви навичок існують для платформи, вони просто будуть використані повторно для кожного виконання завдання.

У цьому розділі ми описали, як ми можемо застосувати структуру, детально описану в розділі 3 безпосередньо до робототехнічних застосувань, у цьому випадку в області виробництва та складання роботів. Такий тип представлення має низку переваг. Представлення можливостей робота на цьому рівні абстракції дозволяє представленню бути незалежним як від мети, так і від реалізації. Наприклад, навичка транспортування просто дозволяє моделі завдання вказати, що дію руху необхідно виконати, без турбот про деталі того, який алгоритм потрібно використовувати для виконання цієї дії, або яка загальна мета системи.

Використання цього узагальненого представлення дозволяє застосовувати ці примітиви навичок до широкого кола проблем, від завдань, подібних або дуже відмінних від тих, що описані в прикладах, а також дозволяє описувати плани та цілі завдань платформи-незалежним способом. Використання представлення, представленого в рамках, підтримує рівень абстракції достатньо високим для узагальнення для багатьох різних описів проблем, а також для різних платформ, оскільки примітиви навичок існують вище рівня апаратної реалізації. Це також звільняє кінцевого користувача від необхідності вивчати опис нижчого рівня можливостей окремих робіт та специфікацій параметрів.

У попередніх розділах ми показали, що кожен із цих експериментів був успішним, оскільки нам вдалося продемонструвати як абстракцію завдання, так і абстракцію платформи. Справжні результати експериментів полягають у тому, що вони показують, що ми можемо використовувати цей підхід високорівневої абстракції для виконання безлічі різних завдань, використовуючи ці узагальнені примітиви навичок та єдиний план виконання завдання на кількох різних платформах. Це означає, що маючи будь-який простий опис завдання, ми можемо виконати його на будь-якій платформі, для якої закодовано таксономічно змодельовані примітиви навичок (як згадувалося раніше, це кодування потрібно робити лише один раз для будь-якої заданої моделі платформи). Сам опис завдання може бути згенерований вручну або автоматично за допомогою різних методів, таких як планування (про що ми розглянемо детальніше в наступному розділі).

Бібліотека навичок.

Іноді робіт стикається з набором можливостей, які часто виконуються разом неодноразово, наприклад, ті, що необхідні для забирання деталі з контейнера для деталей. У таких випадках можна визначити можливість високого рівня, яка складається з цих можливостей нижчого рівня, параметри яких задовольняють параметрам можливостей кожного компонента. Приклад цієї концепції представлено тут.

SCREW-INSERT (X,T)

- 1: *DETECT*
 - 2: *TRANSPORT(X)*
 - 3: *ALIGN*
 - 4: *INSERT*
 - 5: *SCREW MOTION(T)*
 - 6: *UNGRASP*
 - 7: *RETRACT*
-

Для операції взяття об'єкта та його вкручування у відповідному місці визначено композицію «Вкручування-Вставка». У ній присутні необхідні можливості, включаючи переміщення деталі в позицію X (точку вставки) та вкручування деталі з параметрами T, що є порогом крутного моменту, що використовується для визначення завершення дії. Аналогічно, також можна визначити композиції «Піднімання та розміщення».

Це можна застосувати до наведених раніше прикладів. Наприклад, складання моделі транспортного засобу відбуватиметься майже так само, як і раніше, за винятком того, що в цьому випадку опис складання спрощується завдяки використанню визначених композицій. Робот підбирає першу деталь, а потім розміщує її в пристосуванні. Це триває доти, доки роботу не потрібно буде прикрутити блок для вузла колеса. Тут робот підбирає блок, а потім прикручує його в потрібному місці на вузлі колеса. Коли блоки готові, робот повертається до підбирання деталей та розміщення їх на пристосуванні, доки не потрібно буде прикрутити останні болти, після чого робот завершує складання колеса (і моделі), прикручуючи болти.

Кожну з цих новостворених навичок високого рівня можна залучити до нашого пулу доступних можливостей робота. За їх допомогою ми можемо створити Бібліотеку навичок, поєднавши всі наші вже існуючі низькорівневі примітиви навичок з цими високорівневими композиціями. Ця зростаюча Бібліотека навичок формує набагато багатший та гнучкіший метод опису можливостей робота, а також моделей завдань та планів дій.

3.2 планування фреймворку

У попередніх розділах ми продемонстрували, як фреймворк передачі знань GТах може бути використаний для моделювання завдань складання роботів на високому рівні та забезпечення передачі знань у формі абстракції завдань та абстракції платформи. Ми показали, як знання можуть бути представлені в предметній області та як їх можна використовувати для побудови діаграм послідовностей або планів дій щодо того, як робот має досягти загальної мети. Ці діаграми послідовностей описують послідовність примітивів навичок, які робот повинен виконати для успішного досягнення загальної мети завдання. Таким чином, вони можуть бути досить довгими. Наприклад, діаграма послідовності для завдання складання моделі літака має майже двісті п'ятдесят окремих примітивів навичок для виконання завдання. Створення цієї діаграми послідовності вручну може бути трудомістким та виснажливим. Хоча використання високорівневих навичок, таких як «Візьми та розмістіть», може зменшити розмір діаграми послідовності (а також скоротити час та збільшити читабельність), оскільки завдання складання масштабуються за межі складання простих моделей, діаграми послідовностей можуть стати великими та досить складними.

Було проведено дослідження з метою вивчення того, як представлення області планування разом із структурою моделювання завдань, представленою в попередніх розділах, може бути використано для автоматичної генерації цих діаграм послідовності виконання, тим самим зменшуючи обсяг роботи, пов'язаної зі створенням плану для складних виробничих завдань. Це дає можливість полегшити певне навантаження з користувача робота та підвищити надійність цих планів. Враховуючи обсяг загального дослідження, ця робота зосереджена на базовій функціональності роботів-маніпуляторів у виробничих умовах. Дослідження показує, що останні версії так званого Визначення домену планування мова (PDDL) і планувальники програмного забезпечення, що відповідають їм, підходять для представлення відповідних проблем та пошуку

рішень автоматизованим способом. У цій роботі визначено домен планування BasicAssembly PDDL, який охоплює основні можливості виробничих робіт. Зокрема, домен представляє примітиви навичок, які абстраговані в таксономії можливостей SysML, описаній у попередніх розділах. Ці дії представлені з точки зору їхніх передумов та наслідків. Наприклад, наступне твердження є схемою дій домену BasicAssembly, яка визначає дію роботизованої руки ?a, що захоплює деталь ?p зі станції ?s.

```
(:action grasp
  :parameters (?a - arm ?p - part ?s - station)
  :precondition (and
    (arm-at ?a ?s)
    (part-at ?p ?s)
    (arm-active ?a grip)
    (arm-holding ?a no-part)
    (pose-detected ?a ?p ?s))
  :effect (and
    (arm-holding ?a ?p)
    (not (arm-holding ?a no-part))
    (not (part-at ?p ?s))
    (not (pose-detected ?a ?p ?s)))
)
```

Цю область планування було успішно застосовано до кількох варіацій на прикладі багатороботної системи в середовищі автомобільного виробництва. Очікується, що ця робота зробить внесок у область виробничої робототехніки, показавши, як цю структуру моделювання завдань можна використовувати разом із формальними методами планування для покращення процесу автоматизації та зменшення залежності від знань та зусиль користувача. Спільнота планування на основі штучного інтелекту очікує, що ця робота зробить свій внесок у обсяг знань, продемонструвавши відносно нові методи

планування, що враховують обмеження траєкторії, успішно застосовані до конкретних виробничих проблем.

У цьому розділі ми представляємо ідею про те, що використання планування на основі штучного інтелекту разом із формальним моделюванням завдань дозволяє зменшити накладні витрати, пов'язані зі створенням планів для складних завдань. Запропонований підхід використовує нашу таксономію SysML для моделювання можливостей системи та специфікації процесу, а також мову планування PDDL для визначення прийнятних об'єктивних рішень. Ця ідея застосовується до виробничої сфери, і наведено приклади моделювання багатороботної системи в середовищі автомобільного виробництва. Обговорюються переваги продемонстрованого підходу.

Система.

Моделі, процеси, і послідовності для виробництва

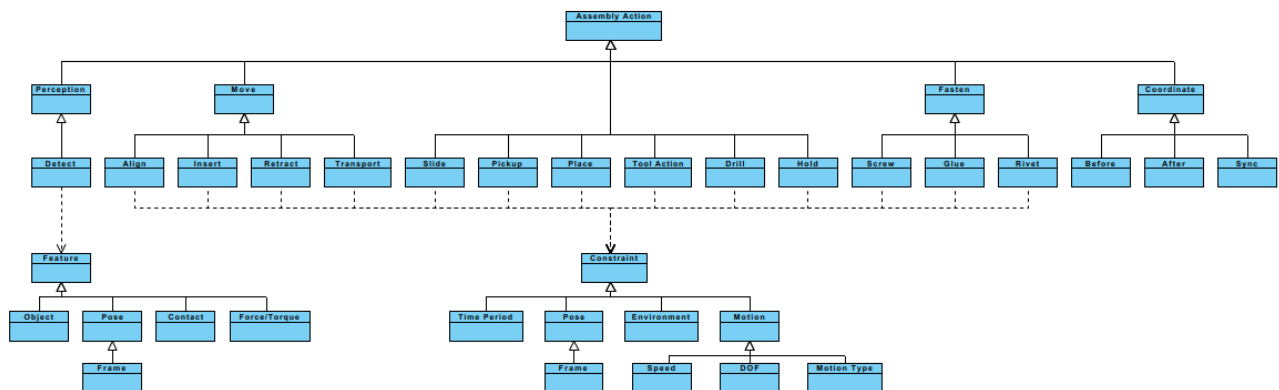


Рисунок 3.6 - Екземпляр моделі для завдань складання.

Одна з цілей цієї роботи — структурувати знання таким чином, щоб вони піддавалися модульності, щоб їх можна було використовувати в різних проблемах і контекстах. З цією метою, Ми визначаємо модельний простір як сукупність усіх можливих можливостей, які робот або роботизована система може використовувати для досягнення певної мети в реальному світі. Екземпляр модельного простору моделюватиме можливості, необхідні для виконання завдань у певній області застосування, такій як виробництво, автоматизація лабораторій тощо. Таким чином, модельний простір охоплюватиме всі

можливості, необхідні у всіх областях застосування роботів. У цій роботі екземпляр моделі має форму таксономії SysML, що визначає кожну з цих можливостей. SysML (або мова системного моделювання) – це загальна мова моделювання для систем та системної інженерії, яка визначається як розширення популярної мови моделювання UML. Переваги використання SysML включають виразну мову моделювання та інструменти для генерації коду, перевірки та валідації системної моделі.

Наприклад, рис. 3.6 показує екземпляр модельного простору для виробничого вузла *bly*. Найпростішими структурними блоками в таксономії є примітиви навичок, які є атомарними діями, які здатний виконувати робот. Примітиви навичок доповнюються певними параметрами або обмеженнями на дію, яку потрібно виконати. Для, наприклад, такої дії, як Вирівнювання та Транспортування, часто виконуватимуться стосовно певної цільової або кінцевої пози.

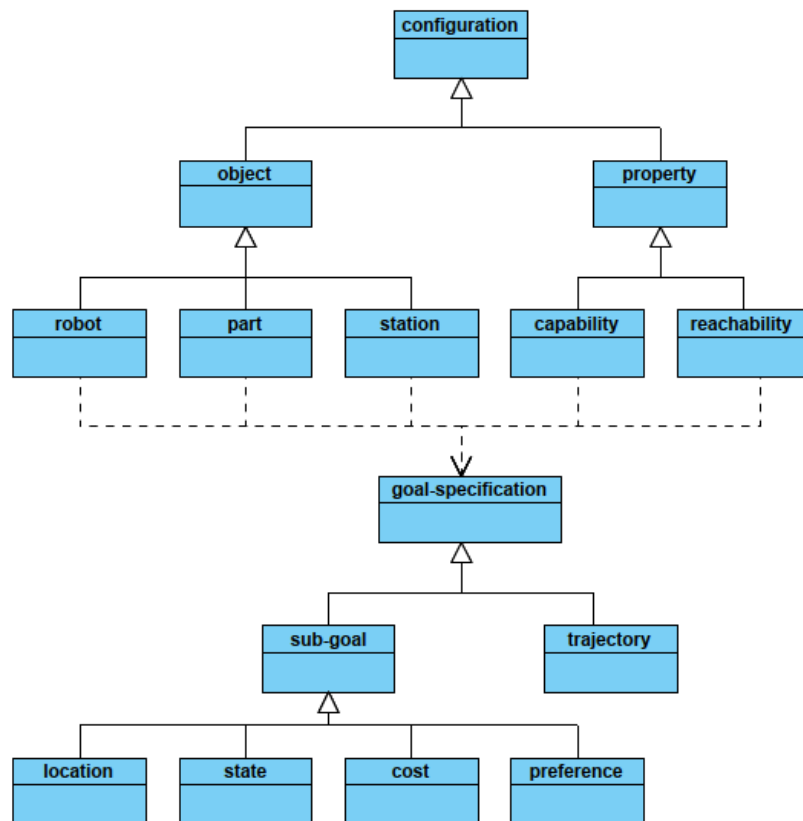


Рисунок 3.7 Екземпляр процесу для завдання складання.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Організація охорони праці при роботі з системою управління

Охорона праці розглядає проблеми забезпечення здорових і безпечних умов праці. Виявляє і вивчає можливі причини нещасних випадків, професійних захворювань, аварій, вибухів, пожеж і розробляє систему заходів і вимог з метою виключення цих причин і створення безпечних і сприятливих для людини умов праці.

Завдання охорони праці є зведення до мінімуму імовірності пошкодження або захворювання працівників з одночасним забезпеченням комфорту при максимальній продуктивності праці.

Навчання працівників безпеці праці проводять відповідно до вимог ГОСТ 12. 0.004 - 79, який встановлює порядок і види навчання. На всіх підприємствах і в організаціях незалежно від характеру і ступеню небезпеки виробництва навчання працівників проводять при підготовці нових робітників, проведенні різноманітних видів інструктажів і підвищенні кваліфікації.

Контроль за своєчасним і якісним навчанням виконує відділ охорони праці чи інженер з охорони праці, або ІТП, на якого наказом керівника підприємства покладено ці обов'язки. Ті, що вперше поступають на роботу, навчання проходять згідно з "Типовим положенням про підготовку і підвищення кваліфікації робітників". В журналі обліку навчальної роботи реєструють навчальну тему, за якою проводилось навчання.

Інструктаж працюючих поділяють на вступний, початковий, на робочому місці, повторний, позаплановий і початковий.

Вступний інструктаж з усіма, хто поступає на роботу незалежно від їх освіти і стажу роботи по даній професії, проводить інженер з охорони праці за

програмою, затвердженою головним інженером підприємства, про проведення вступного інструктажу з обов'язковим підписом того, хто проводив інструктаж і того, хто його отримував.

Початковий інструктаж на робочому місці, повторний, позаплановий і поточний проводить керівник робіт.

Початковий інструктаж на робочому місці проводять при прийомі на роботу нових робітників за інструкцією з охорони праці, розробленою для окремих професій або видів робіт. Всі робітники після цього інструктажу і перевірки знань 2-5 змін (залежно від навичок і стажу роботи) працюють під наглядом бригадира чи майстра, потім оформляється допуск до їх самостійної праці.

Повторний інструктаж проходять всі працівники незалежно від кваліфікації, освіти і стажу роботи через три місяці. Його проводять з метою перевірки знання робітниками правил і норм з охорони праці.

Позаплановий інструктаж проводять коли змінилися правила охорони праці або технологічний процес, обладнання, інструмент та інші фактори, що впливають на безпеку праці; коли працівники порушують правила охорони праці, що можуть призвести чи призвели до травм, аварій чи пожежі, вибуху. Його проводять індивідуально чи з групою робітників однієї професії за програмою початкового інструктажу на робочому місці. При його реєстрації вказують причину, яка спричинила його проведення.

Умови праці мають велике значення практично для всіх виробничих показників - продуктивності праці, якості робіт, безпеки працівників та інше.

Санітарно-гігієнічні умови праці характеризуються показниками виробничого середовища - рівнем освітлення, мікрокліматичними параметрами, загазованістю і запиленістю повітряного середовища, рівнем шуму і вібрації, наявністю іонізуючого випромінювання та інше.

4.2 Електробезпека

Електричні установки, з якими доводиться мати справу практично всім працюючим по встановленню та налагодженню засобів автоматизації, виявляють для людини велику потенційну небезпеку, яка збільшується у зв'язку з тим, що органи чуття людини не можуть на відстані виявити присутність електричної напруги на обладнанні.

Степінь ураження електричним струмом залежить від цілого ряду факторів: значення сили струму, електричного опору тіла людини та тривалості протікання через неї струму, виду та частоти струму, індивідуальних властивостей людини та умов навколишнього середовища.

Конструкція електроустановок має відповідати умовам їх експлуатації та забезпечувати захист персоналу від дотику з струмоведучими та рухомими частинами, а обладнання - від попадання всередину посторонніх твердих тіл та води.

Конструкція, вид виконання, спосіб встановлення, клас ізоляції застосовуваних провідників, кабелів, пристроїв та іншого електрообладнання відповідають вимогам електробезпеки. За ступенем ураження людей електричним струмом котельня відноситься згідно ПУЕ 1.1.13 до категорії приміщень з підвищеною небезпекою (висока температура, можливість одночасного дотику до металевих елементів технологічного обладнання або металоконструкцій будинку та металевих корпусів електрообладнання).

У нормальному режимі роботи обладнання - можливість ураження працівників електричним струмом виключена. Але на випадок аварії для запобігання ураження струмом людей передбачене захисне заземлення. Згідно ПУЕ 1.7.65 допустимий опір заземлення повинен бути не більшим 10 Ом.

При виконанні монтажних робіт використовуються переносні електроінструменти (електродрилі, електрошліфувальні установки, тощо).

Для забезпечення безпечної праці корпуси однофазних електроприймачів повинні занулюватись.

Захист людини від ураження електричним струмом в мережах з зануленням здійснюється тим, що при замиканні одної з фаз на занулений корпус в ланці цієї фази виникає струм короткого замикання, що діє на струмовий захист (плавкий запобіжник, автомат), в результаті чого відбувається відключення аварійної ділянки від мережі. Крім того, ще до спрацювання захисту струм короткого викликає перерозподіл напруги в мережі, що приводить до зниження напруги корпусу відносно землі. Таким чином, занулення зменшує напругу дотику та обмежує час, на протязі якого людина, що доторкнулася до корпусу, може потрапити під дію напруги.

Для того, щоб забезпечити швидке (на протязі декількох секунд) відключення аварійної ділянки, струм короткого замикання повинен бути достатньо великим. Відповідно до вимог ПУЕ струм короткого замикання повинен не менше ніж в три рази перевищувати номінальний струм плавкої вставки найближчого запобіжника або номінальний струм нерегульованого розчеплювача автоматичного вимикача. При використанні автоматичних вимикачів, що мають тільки електромагнітний розчіплювач (відсічку), струм короткого замикання повинен перевищувати значення струму встановлення миттєвого спрацювання в 1,25-1,4 рази в залежності від номінального струму.

В однофазних електроприймачів, що включені між фазним та нульовим робочим проводами, занулення корпусів слід виконувати з допомогою окремого (третього) провідника, який повинен з'єднувати корпус електроприймача з нульовим захисним проводом. В таких випадках під'єднувати корпуси електроприймачів для забезпечення електробезпеки до нульового робочого проводу недопустимо, оскільки при його розриві (перегоранні запобіжника) всі під'єднані до нього корпуси виявляться під фазною напругою відносно землі.

В мережі з зануленням недопустимо використовувати заземлення окремих електроприймачів, не під'єднавши їх перед цим до нульового захисного провідника. В цьому випадку при замиканні фази на заземлений, але не приєднаний до нульового захисного провідника корпус створюється коло струму через заземлення цього корпусу та заземлення нейтралі джерела струму. Такий випадок небезпечний, оскільки засоби захисту не зможуть відключити такий електроприймач через мале значення струму і тому небезпечна напруга на всіх корпусах може зберігатися тривалий період, поки заземлений приймач не буде відключений вручну.

Важливо відмітити, що якщо занулений корпус одночасно заземлений, то це тільки покращує умови безпеки, оскільки забезпечує додаткове заземлення нульового захисного проводу.

Для ізоляції людини від частин електроустановок, що знаходяться під напругою, використовуються основні та допоміжні ізолюючі засоби, а саме слюсарно-монтажний інструмент з ізольованими ручками, коврики, ізолюючі підставки, тощо.

У приміщеннях, де знаходяться вимірювальні прилади, необхідно забезпечити виконання заходів по боротьбі з статичною електрикою (тобто прилади повинні бути заземлені). Найпростішим засобом є підтримка відносної вологості повітря на рівні 50 - 60 % за допомогою побутового електрозволожувача.

Підлогу слід виконувати відповідно до ГОСТ 12.4.124-83, використовуючи антистатичне покриття на проходах і біля робочих місць.

Робітникам рекомендовано носити одержу з природних матеріалів або з комбінованих - природних і штучних волокон. Для зняття електростатичних зарядів з одержу слід використовувати антистатики побутового призначення.

Оскільки корпуси приладів виконані з металу, то для усунення небезпеки ураження людини електричним струмом (можливий пробій на корпус приладу) використовується захисне заземлення.

4.3 Розрахунок заземлення

Розрахуємо систему заземлення для електроустаткування, яке працює від напруги 220 В.

$$R_{\text{заз}} \leq \frac{U}{I_p} = \frac{220}{66} = 3.3 \leq 4 \text{ Ом}$$

Визначаємо опір ґрунту: $\rho = \kappa_n * \rho_n = 2 * 200 = 400 \text{ Ом м}$,

де κ_n - коефіцієнт підсилення;

ρ_n — питомий опір ґрунту (вибирається з довідкової літератури).

Визначаємо опір одиночного вертикального заземлювача:

$$R_B = \frac{\rho}{2\pi} \left(\ln \frac{2l}{d} + \frac{1}{2} * \frac{4t+1}{4t-1} \right)$$

де t - відстань від середини заземлювача до поверхні ґрунту, м;

l, d - довжина і діаметр стержня заземлювача, м;

$$R_B = 96 \text{ Ом.}$$

Визначаємо опір сталевій полоси, що з'єднує стержневі заземлювачі:

$$R_{II} = (\rho / 2\pi l) * \ln(l^2 / dt) = 61 \text{ Ом.}$$

Визначаємо орієнтовне число стержневих заземлювачів:

$$n = R_B / [r_B] \eta_B = 96 / 4 * 1 = 24 \text{ шт.};$$

r_B - допустимий по нормам опір заземляючого пристрою,

η_B - коефіцієнт використання вертикальних заземлювачів (для орієнтовного розрахунку приймається рівним 1).

Приймаємо розміщення вертикальних заземлювачів по контуру з відстанню між сталевими заземлювачами рівним 21. З довідкової літератури визначаємо $\eta_B = 0,66$ і $\eta_I = 0,39$.

Визначаємо необхідну кількість вертикальних заземлювачів

$$n = R_B / [r_B] \eta_B = 96 / (4 * 0.66) = 36$$

Розраховуємо загальний розрахунковий опір аземлюючого пристрою R з врахуванням з'єднувальної полоси

$$R = R_B R_{II} / (R_B \eta_I + R_{II} \eta_B n) = 3.9 \text{ Ом.}$$

Розрахунок проведено правильно, оскільки виконується умова $R \leq [r_B]$.

Розрахунок штучного заземлення:

Приймаємо, що опір захисного заземлення не повинен перевищувати 4 Ом:

$$R_{33} = \frac{R_c R_n}{R_c + R_n} \leq 4 \text{ Ом}$$

де R_{33} – опір захисного заземлення;

R_c – опір стержневих заземлювачів;

R_n – опір поперечних заземлювачів.

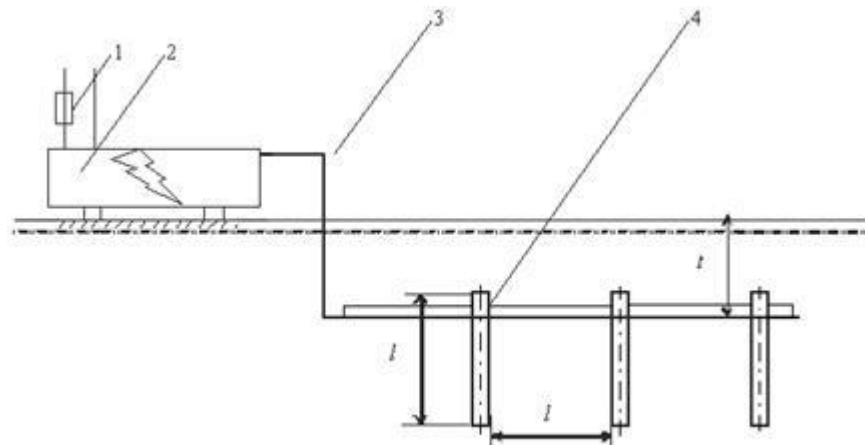


Рисунок 6.1 - Пристрій заземлення

4 – плавка вставка; 2 – електроустановка; 3 – з'єднувальна штаба; 4 – трубчастий заземлювач

Опір одиночного стержневого заземлювача розтіканню електричного струму:

$$R_{oc} = \frac{\rho_p}{2\pi l} \left(\ln \frac{2l}{d} + \ln \frac{4h' + l}{4h' - l} \right)$$

де h – відстань від поверхні ґрунту до заземлювача і становить 0,8 м;

l – довжина стержневого заземлювача 3 м;

d – діаметр стержневого заземлювача 50 мм.

$$R_{oc} = \frac{750}{2 \cdot 3,14 \cdot 3} \left(\ln \frac{2 \cdot 3}{0,05} + \ln \frac{4 \cdot 0,8 + 3}{4 \cdot 0,8 - 3} \right) = 39,8 \cdot (0,18 + 3,43) = 143,8 \text{ Ом}$$

Опір одиночного поперечного заземлювача:

$$R_{ок} = \frac{\rho_p}{2\pi l} \ln \frac{2l^2}{bh'}$$

де l – довжина поперечного заземлювача 2,5 м;

b – ширина полоси заземлювача 30 мм;

ρ_p – розрахунковий опір ґрунту: для поперечних електродів 1000 Ом·м, для стержневих електродів 750 Ом·м.

$$R_{ок} = \frac{1000}{2 \cdot 3,14 \cdot 2,5} \ln \frac{2 \cdot 2,5^2}{0,03 \cdot 0,8} = 63,7 \cdot 6,25 = 398,1 \text{ Ом}$$

В наслідок взаємовпливу вводимо коефіцієнт використання заземлювачів:

$$\eta = \frac{R_0}{nR_{\phi}}$$

де R_d – допустимий опір заземлення, що становить 4 Ом;

R_0 – опір одиночного заземлювача.

ВИСНОВКИ

У роботі було розглянуто проблему негнучкості в сучасних виробничих процесах, що включають робототехніку та програмування роботів, і запропоновано рішення, в якому використовується фреймворк моделювання завдань на основі SysML для моделювання можливостей робота в заданій області застосування. Використовуючи цей фреймворк для моделювання навичок робота на вищому рівні абстракції, показано, що є можливість внести більше гнучкості у виробничий процес, розглядаючи програмування та взаємодію роботів на рівні завдань.

Підійшовши до проблеми таким чином, була отримана можливість абстрагуватися від багатьох труднощів, пов'язаних із сучасними методами взаємодії, і водночас створити структуру, яка може охопити простір взаємодії, і тим самим забезпечити передачу знань у нашій цільовій області.

ПЕРЕЛІК ПОСИЛАНЬ

1. Abbas, T.; MacDonald, B. Generalizing topological task graphs from multiple symbolic demonstrations in programming by demonstration (PbD) processes // Robotics and Automation (ICRA), 2011 IEEE International Conference on. – May 2011. – P. 3816–3821.
2. Balakirsky, S.; Kootbally, Z.; Schlenoff, C.; Kramer, T.; Gupta, S. An industrial robotic knowledge representation for kit building applications // IEEE/RSJ International Conference on Intelligent Robots and Systems. – San Francisco, 2012.
3. Benton, J.; Coles, A.; Coles, A. Temporal planning with preferences and time-dependent continuous costs. – 2012.
4. Huckaby, J.; Christensen, H. I. A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics // 8th International Cognitive Robotics Workshop. – July 2012.
5. Jarraya, Y.; Debbabi, M. Formal specification and probabilistic verification of SysML activity diagrams // Theoretical Aspects of Software Engineering (TASE), 2012 Sixth International Symposium on. – July 2012. – P. 17–24.
6. Konidaris, G.; Kuindersma, S.; Grupen, R.; Barto, A. Robot learning from demonstration by constructing skill trees // The International Journal of Robotics Research. – 2011.
7. Kaelbling, L. P.; Lozano-Pérez, T. Hierarchical planning in the now // IEEE Conference on Robotics and Automation (ICRA). – 2011. – Finalist, Best Manipulation Paper Award.
8. Kress-Gazit, H.; Wongpiromsarn, T.; Topcu, U. Correct, reactive, high-level robot control // IEEE Robotics Automation Magazine. – Vol. 18. – Sept. 2011. – P. 65–74.
9. Krüger, N.; Geib, C.; Piater, J.; Petrick, R.; Steedman, M.; Wörgötter, F.; Ude, A.; Asfour, T.; Kraft, D.; Omrcen, D.; Agostini, A.; Dillmann, R. Object-Action Complexes: Grounded abstractions of sensory-motor processes // Robotics & Autonomous Systems. – Vol. 59. – Oct. 2011. – P. 740–757.

10. Lim, G. H.; Suh, I. H.; Suh, H. Ontology-based unified robot knowledge for service robots in indoor environments // IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. – Vol. 41. – May 2011. – P. 492–509.
11. Rusu, R. B.; Cousins, S. 3D is here: Point Cloud Library (PCL) // IEEE International Conference on Robotics and Automation (ICRA). – Shanghai, China, May 9–13, 2011.
12. Rahman, M. A. A.; Mayama, K.; Takasu, T.; Yasuda, A.; Mizukawa, M. Model-driven development of intelligent mobile robot using systems modeling language (SysML) // Mobile Robots – Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training. – InTech, 2011.
13. Waibel, M.; Beetz, M.; D'Andrea, R.; Janssen, R.; Tenorth, M.; Civera, J.; Elfring, J.; Gálvez-López, D.; Häussermann, K.; Montiel, J.; Perzylo, A.; Schiessle, B.; Zweigle, O.; van de Molengraft, R. RoboEarth – A World Wide Web for Robots // IEEE Robotics & Automation Magazine. – Vol. 18, No. 2. – 2011.
14. Huckaby, J.; Vassos, S.; Christensen, H. I. Planning with a task modeling framework in manufacturing robotics // IEEE/RSJ International Conference on Intelligent Robots and Systems. – 2013.
15. Chao, C.; Lee, J.; Begum, M.; Thomaz, A. L. Simon plays Simon says: The timing of turn-taking in an imitation game // RO-MAN, 2011 IEEE. – 2011. – P. 235–240.
16. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 1. [навчальний посібник] (Лист МОНУ №1/11-8052 від 28.05.12р.) - Львів, "Магнолія 2006", 2013. – 256 с.
17. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 2. [навчальний посібник] (Лист МОНУ №1/11-11650 від 16.07.12р.) - Львів, "Магнолія 2006", 2014. – 312 с.
18. Микитишин А.Г., Митник, П.Д. Стухляк. Комплексна безпека інформаційних мережевих систем: навчальний посібник – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. – 256 с.

19. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі : навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 384 с.
20. Введення в компютерну графіку та дизайн: Навчальний посібник для студентів спеціальності 174 "Автоматизація, компютерно-інтегровані технології та робототехніка"/Укладачі: О.В. Тотосько, П.Д. Стухляк, А.Г. Микитишин, В.В. Левицький, Р.З. Золотий - Тернопіль: ФОП Паляниця В.А., 2023 - 304с. <http://elartu.tntu.edu.ua/handle/lib/41166>.
21. Пилипець М. І. Правила заповнення основних форм технологічних документів : навч.-метод. посіб. / Уклад. Пилипець М. І., Ткаченко І. Г., Левкович М. Г., Васильків В. В., Радик Д. Л. Тернопіль : ТДТУ, 2009. 108 с. <https://elartu.tntu.edu.ua/handle/lib/42995>.