



Robotic Process Automation (RPA)

Day 1

2020. 6. 13

정 준 수 Ph.D.

SW Bot & RPA 란?



소프트웨어 로봇(SW Bot)이란?

답) 몇 시간만 투자하면 평생 손과 발이 편해질 수 있는 도구(SW)

RPA란?

답) 소프트웨어 로봇(Robotic)을 통한 프로세스 자동화(Process Automation)

역사는 흐른다!

Digital divide -> RPA divide -> AI divide



RPA는 손과 발, Machine Learning은 머리

RPA와 AI의 차이점은?

AI와 AI를 지원하는 기술(RPA,머신러닝,NLP)은 실제로 의미하는 것과 다르게 인식되고 있어 많은 혼란과 오해를 야기했습니다. 많은 오해 중 하나는 AI와 RPA (Robotic Process Automation)입니다.

두 기술 모두 BPA (비즈니스 프로세스 자동화)에 큰 지분을 차지하고 있지만 각각 다른 방식으로 작업을 수행합니다. 간단히 말하자면 RPA를 행동하는 자로 생각하고 AI를 생각하는 사람으로 생각하십시오.

RPA - 로봇프로세스자동화

RPA (Robotic Process Automation)는 반복적인 작업을 자동화하여 비즈니스 프로세스를 간소화하기 위한 소프트웨어입니다.

RPA는 규칙기반 소프트웨어입니다. 화면 스크래핑, 워크 플로우 자동화 등 소프트웨어가 데이터를 집계하고 응답을 트리거하며 작업을 실행합니다.

RPA는 인간 행동을 모방하여 생산성을 높이기 위해 일관된 루틴화 된 워크 플로우를 자동화합니다. 규칙이 프로그래밍되고 봇은 Excel과 같은 애플리케이션에서 구조화 된 입력을 추출하여 SAP에 입력할 수 있습니다.

RPA는 사람처럼 사용자 인터페이스에서 작동하며 ERP 나 CRM에 완벽하게 적응하죠.

민첩하고 가벼운 RPA는 반복적인 프로세스를 지속적으로 완료 할 수있어 엄청난 인기와 성공을 거두었습니다. 직원들을 반복적인 작업에서 해방시키고 인적 오류를 제거했습니다.

RPA의 최대장점은 기존 방법보다 적은 비용으로 적은 시간으로 고품질의 수율을 달성할 수 있다는것입니다. RPA는 IT 지원 프로세스, 워크 플로우 프로세스 및 백 오피스 업무를 모두 처리하는 동시에 정규 직원을 고용하는 것보다 저렴한 비용으로 생산성을 향상시킵니다.

RPA는 트랜잭션 및 반복 작업 속도가 빠른 보험 및 은행업과 같은 산업에 적합하지만 그게 전부입니다. 지능이 없기 때문에 지시된 규칙이 없으면 작업을 더 이상 수행하지 않습니다.

여기에 인공 지능이 사용됩니다.

시연 1: RPA를 이용한 오픈마켓 판매가 자동화

시연 2: RPA를 활용한 입·출금대사업무

시연 3: 시계열 분석을 통한 예측분석 자동화

RPA (Robotic Process Automation) 맛보기

rpa (Robotic process Automation)는 단순반복 업무를 손쉽게 해주는 프로그램으로 기존 프로그램에 내장된 매크로(macro). 클라우드서비스프로그램의 스크립트(script)도 rpa 와 같은 기능을 보여줄수 있지만, UIPath, Automation Anywhere 등 독립된 프로그램을 주로 가리킵니다.

MS Office 안에 내장된 Visual Basic Application (VBA), G-Suite에 포함된 google Adds script (GAS) 은 물론 Python으로도 원페이지 내용을 가지고 오거나 데이터를 처리하는 단순반복 업무를할 수 있습니다.

script 등 컴퓨터 언어를 알아야 하는 것과는 달리 rpa 전문 프로그램은 반복되는 업무를 잘 정의하는 것이 중요합니다.

예를 들면, 평소 자료를 엑셀에서 관리하다가 특정한 웹페이지에 하나하나 입력해야 할 때 RPA 프로그램으로

엑셀과 웹을 연동시켜 자동으로 입력하고 제출할수 있도록 해줍니다.

Powerful Automation

RPA와 AI 기술은 각각 비즈니스 프로세스 자동화에 훌륭한 도구이지만 AI가 RPA와 합쳐지면 보다 강력한 자동화 프로세스가 완성됩니다. AI로 선택과 결정을 하고 RPA로 작업을 완료합니다. USE CASE 측면에서 업무 시간 이외의 시간에 발생한 고객의 대기시간을 줄이는 것만큼 간단할 수 있습니다.

가트너에 의하면 전 세계 RPA 시장 규모는 13억 달러로 평가되었으며 2020년 ~ 2027년 동안 매년 31.1 %의 성장세를 달성할 것으로 예측됩니다. RPA의 사용을 통해 비즈니스 프로세스 자동화 (BPA)에 대한 수요도 대폭 증가했고 AI와 소프트웨어 로봇 시장의 성장에도 긍정적인 요인이 되고 있습니다. 더 많은 기업에서 생산성을 높이고 비용을 절감하는 솔루션을 계속 요구함에 따라 RPA와 AI의 미래는 점점 밝아지고 있습니다.

이제 RPA는 선택이 아닙니다. 더이상 늦출수 없는 거대한 파도입니다.



프레더릭 윈슬로 테일러 (Frederick Winslow Taylor)

출생: 1856년 3월 20일, 미합중국 펜실베이니아 주 필라델피아

프레더릭 윈슬로 테일러는 미국의 기술자이다.
세계 최초의 경영 컨설턴트 중 한 사람이기도 하다.
조직의 운영, 관리에 있어 처음으로 객관적 수치와
데이터를 도입하여 현대 경영학과 산업공학의 효시가 된
이론인 과학적 관리론을 정립한 사람이다.
대표 저서로는 1911년 발표한
'과학적 관리론'(*The principles of Scientific Management*)이 있다.

<https://namu.wiki/w/%ED%94%84%EB%A0%88%EB%8D%94%EB%A6%AD%20%EC%9C%88%EC%A6%90%EB%A1%9C%20%ED%85%8C%EC%9D%BC%EB%9F%AC?from=%ED%94%84%EB%A0%88%EB%8D%94%EB%A6%AD%20%ED%85%8C%EC%9D%BC%EB%9F%AC>

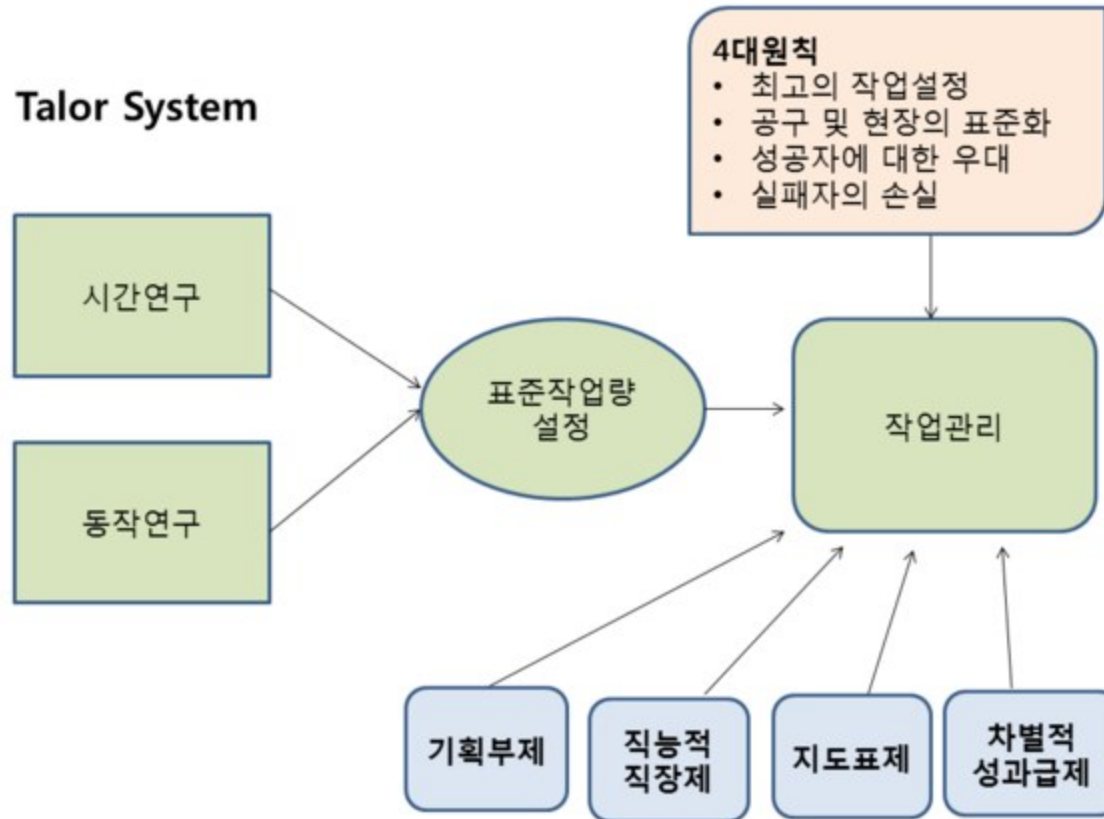
미국의 공학 기술자 [프레더릭 테일러](#)가 19세기 말부터 연구하고 발표한 이론. 발표 이후 1929년 [세계 대공황](#) 이전까지 전 세계적인 주목을 받은 [조직 이론](#)이자 [생산관리](#) 이론이다.

19세기 말부터 본격적으로 시작된 [산업혁명](#)^[1]과 자연과학, 공학의 급진적인 발전, [엽관주의](#)의 타파^[2], 행정-경영의 일원화의 과정에서 만들어진 이론으로 과학적인 수치와 근거에 기초하여 조직을 관리하기 위해 탄생한 이론이다. 생산과정에 있어서 필요한 지식과 기술을 적절히 활용해 최소비용으로 최대 능률을 올리고자 하는 관리 이론. 여기에서 파생된 결과로는 프랭크 길브란트, 릴리안 길브란트 부부의 동작 연구(Motion Study)^[3], 헨리 간트의 시간 연구(Time Study)와 간트 차트(Gantt Chart)^[4], 그리고 독일의 전설적인 사회학자 [막스 베버](#)의 [관료제](#)가 있다. [프레더릭 테일러](#)가 '과학적 관리론'이라는 이론을 창안하고 발전시켰기에, 그의 이름을 따서 테일러 시스템(Taylor System), 혹은 아예 테일러리즘(Taylorism)이라고 하기도 한다.

사실상 현대 [경영학과 산업공학](#)의 효시가 된 이론이다. 과학적 관리론에 이론적 토대를 두고 객관적 기준에 의한 과학적이고 합리적인 방법에 따라 기업을 경영하는 방식이 미국을 시작으로 유럽까지 전파되었다. 그리고 이러한 경영 방식이 [제 2차 세계대전](#)의 [미국](#)을 거치며^[5] 더욱 정교하게 발전하여 사회과학의 한 분야로서 '경영학'이라는 학문이 정립되었다.

과학적 관리법(scientific management) 또는 **과학적 경영**이란, 창안자인 [프레더릭 윈즐로 테일러](#)의 이름 따, 테일러리즘(Taylorism)이라고 불리며, 20세기 초부터 주목받은 [과업](#)수행의 분석과 혼합에 대한 관리 이론이다. 이 이론의 핵심 목표는 경제적 효율성, 특히 **노동생산성 증진**에 있다.

이러한 [과학](#)적 관리법의 진행은 다음과 같이 수행된다. 우선 작업을 과업단위로 분류를 시도하며 동시에 해당 과업을 수행할 적합한 작업자를 체계적으로 선발을 한 뒤에 이러한 각각의 과업을 최대한 빠르고 효율적으로 수행할 수 있도록 시간연구와 동작연구를 시도한다. 연구를 통한 개선점이 도출 될 시에 이를 노동자들에게 작업수행의 표준화, 지침으로 교육을 하여, 이러한 과업수행이 효과적으로 나타나고, 목표량을 달성 시, 수행자에게 인센티브를 지급으로 마무리한다.





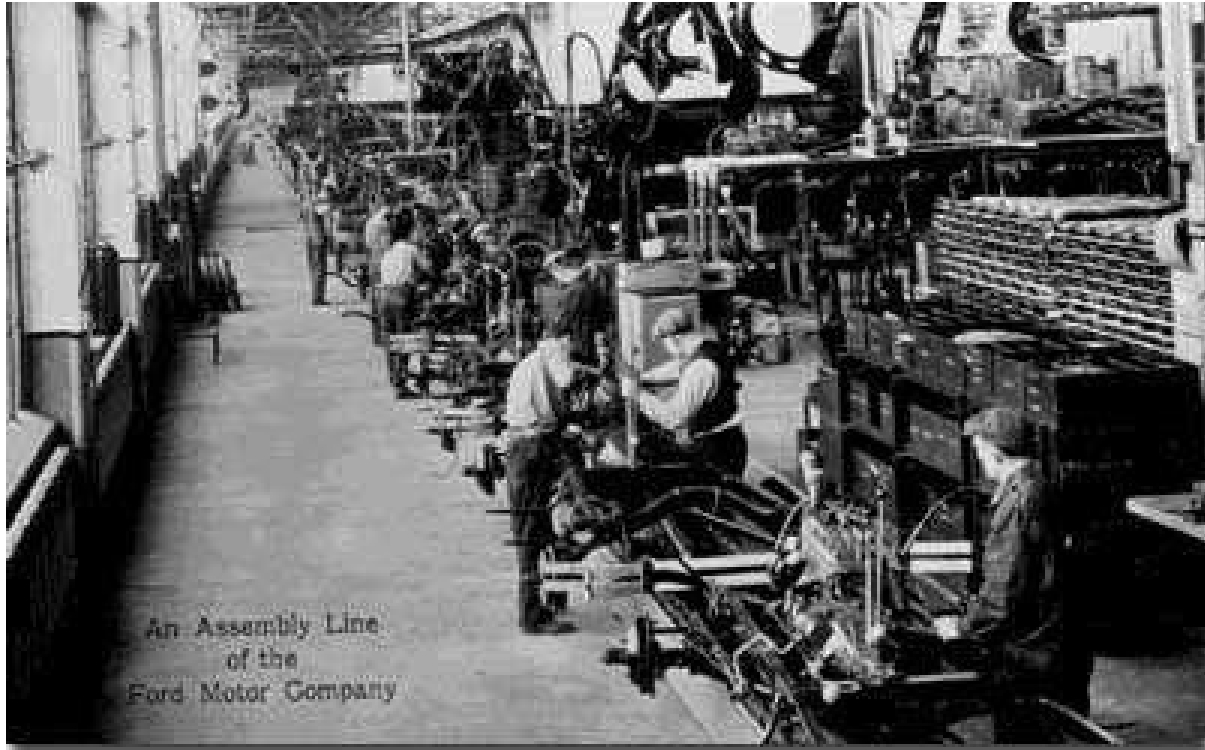
헨리 포드([영어](#): Henry Ford, [1863년 7월 30일](#) ~ [1947년 4월 7일](#))는 [미국의 기술자](#)이자 [기업인](#)으로 [포드 모터 컴퍼니](#)의 창설자이다. [미국 미시간주 디트로이트](#) 서쪽의 농촌에서 농부의 아들로 태어났다. 농업 노동의 합리화를 위한 운반의 개선을 위해 기계 기사를 지망, [디트로이트](#)의 작은 기계 공장에 들어가 직공으로서 기술을 배웠다. 5년 후 고향에 돌아가 농사를 돌보면서 공작실을 만들어 연구를 계속하였고, 1890년 [에디슨 조명 회사](#) 기사로 초청되어 근무하던 중 [내연 기관](#)을 완성하여 1892년 [자동차](#)를 만들었다. 1903년 세계 최초의 양산대중차 [포드 모델 T](#)의 제작을 시작하였다. [포드 모델 T](#)는 자동차의 대중화를 위해 헨리 포드가 실현한 대량 생산 방식의 자동차였다. 그는 특히 경영지도원리로써 미래에 대한 공포와 과거에 대한 존경을 버릴 것, 경쟁을 위주로 일하지 말 것, 봉사가 이윤에 선행할 것, 값싸게 제조하여 값싸게 팔 것 등 4개의 봉사원칙을 내세웠는데 이를 포디즘이라 한다. 한편 포드는 공장의 경영합리화를 위해 제품의 표준화, 부분품의 단순화, 작업의 전문화라는 3S운동을 전개하면서 이 원칙을 달성하기 위하여 누드젠콘이 창안한 컨베이어 시스템을 채용하여 흐름작업조직으로 노동생산성 고무에 이바지하였다. 이것을 '[포드 시스템](#)'이라 하는데 특별히 경영을 봉사 기관으로 보는 포드의 사상은 P. H. 드락카의 경영 이론에 계승되고 있다. 그러나 개성이 강한 그의 경영 스타일 때문에 말년에는 파탄지경에 내몰렸다. 저서에 《오늘과 내일(1926)》, 《나의 산업 철학(1929)》 등이 있다. [1947년 4월 7일](#), 향년 84세로 사망했다. 증손자는 윌리엄 클레이 포드 2세이다.

Henry Ford's Vision - his goal of bringing car travel to the masses!

over 12 hours building each car (EyeWitness to History, 2005).

This process was very expensive and time-consuming, thus making it impossible for Ford to mass-produce his cars at affordable prices. Ford hired management theorist Frederick Winslow Taylor to help map out possible solutions (The Franklin Institute).

궁극적으로 포드는 경영 이론가 Frederick Winslow Taylor를 고용하여 가능한 솔루션 (프랭클린 연구소)을 찾아 보았습니다.



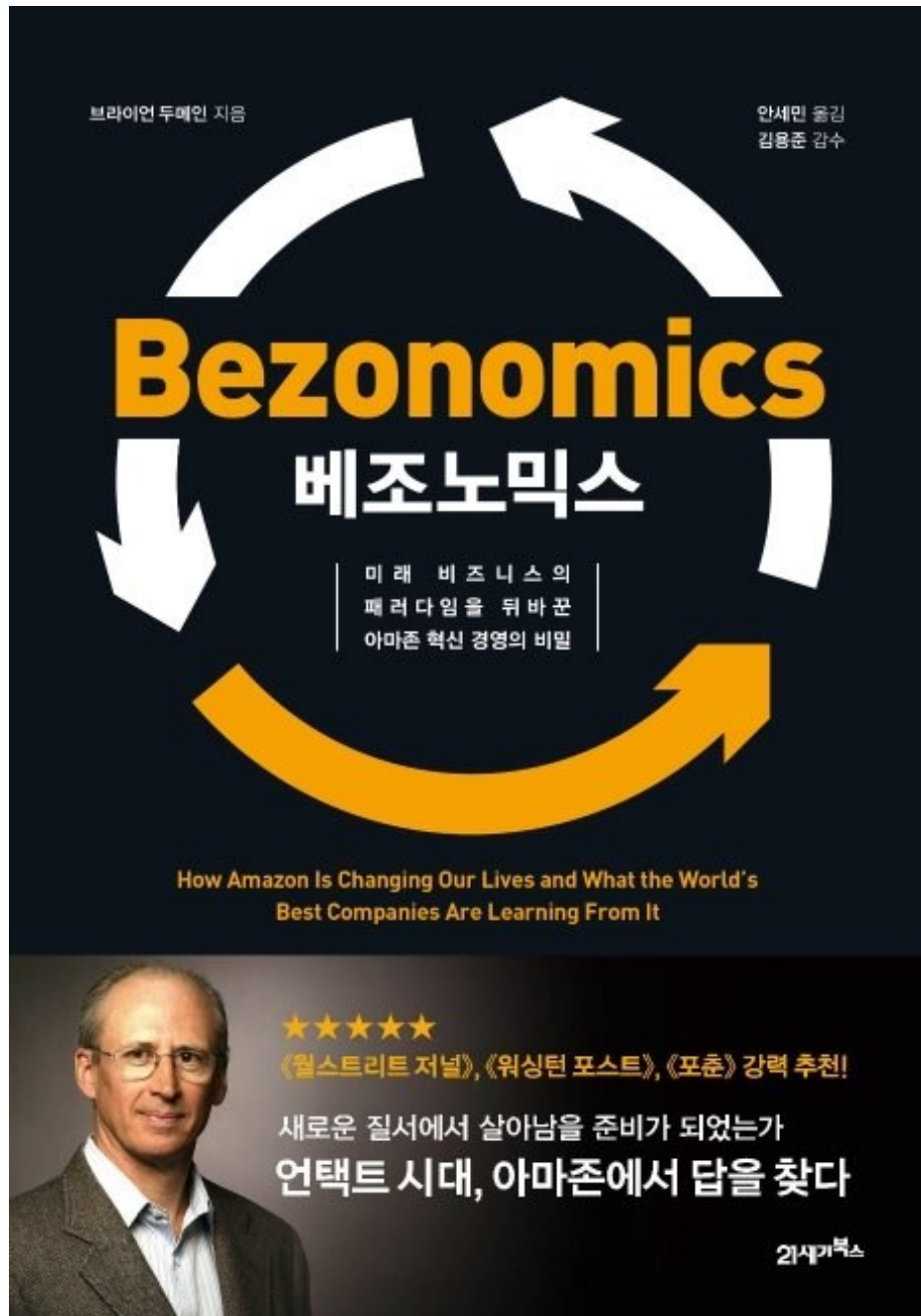
The founder of the Ford Motor Company, and sponsor of the development of the assembly line technique of mass production. The Model T was introduced on October 1, 1908.

After only a few short years, Ford had managed to bring the average time of production for a Model T down to 93 minutes, and as a consequence was able to lower the price down to \$575. By 1914, Ford had captured 48% of the automobile market (EyeWitness to History, 2005).

시사점

In the early days of 1914, Ford raised the wages that he was paying his workers from \$2.83 for a 9-hour day to \$5.00 for an 8-hour day (The Franklin Institute).

By 1924 the successes of Ford's practices were obvious, after just 16 years of implementing Taylor's scientific approach he had managed skillfully sell over 10 million cars (ibid).



저자는 "이제 베조노믹스를 이해하지 않고는 미래의 산업을 전망할 수 없게 됐다"고 말하며 베조노믹스의 3대 요소로 '고객 집중', '극단적인 혁신', '장기적 시각'을 꼽았다. 이를 통해 베조스는 최초로 **빅데이터**와 **인공지능**으로 움직이는 기업을 키웠다. 막대한 빅데이터를 기반으로 하는 아마존의 알고리즘은 세계의 어떤 기업도 넘볼 수 없는 강력한 경쟁력이 됐다. 아마존은 **로봇공학**을 선도한다. 이회사는 설립 이후로 2019년 까지 65만개가넘은 일자리를 창출했지만, 이제는 이 회사가 일으키는 자동화의 물결이 정부가 보편적 기본소득을 심각하게 고민해야 할 정도로 노동 시장을 일대 혼란에 빠뜨리게 될 것이다.

ALGORITHMS AND COLLUSION

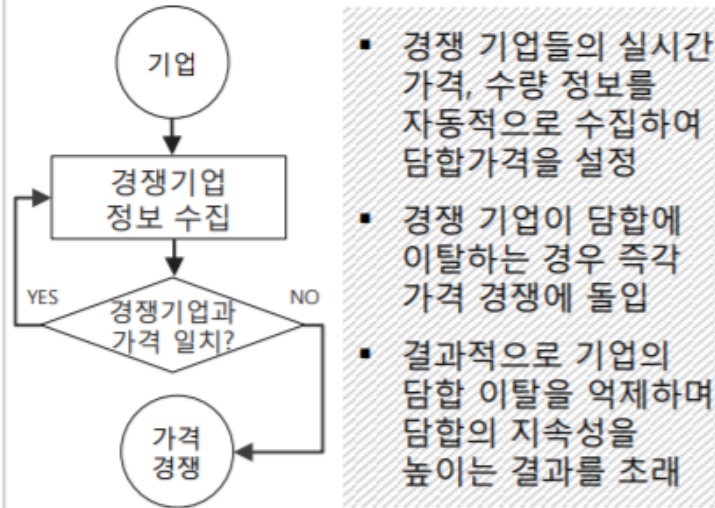
Competition policy in the digital age



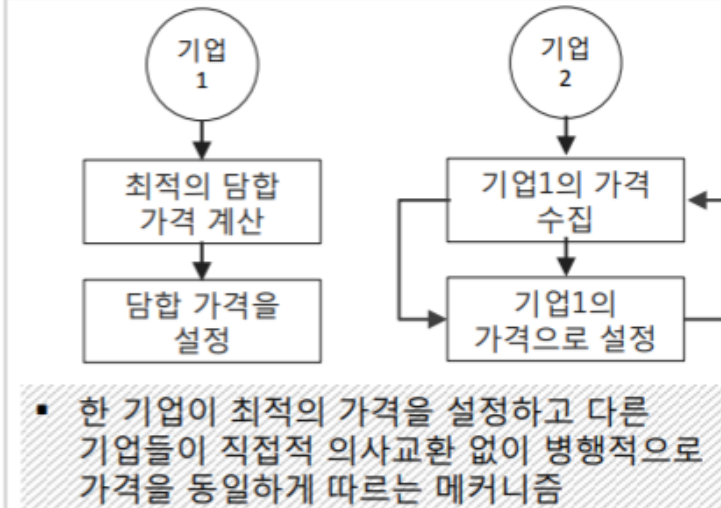
<http://www.oecd.org/daf/competition/Algorithms-and-collusion-competition-policy-in-the-digital-age.pdf>

가격 담합을 조장하는 4가지 알고리즘 유형

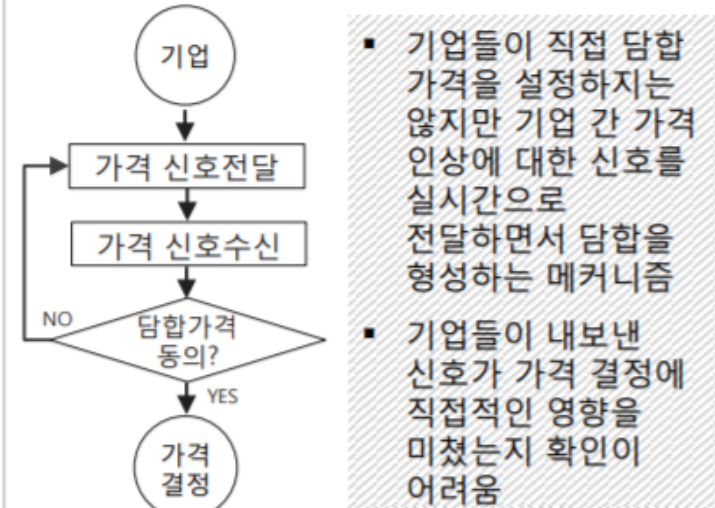
모니터링 알고리즘 (Monitoring Algorithms)



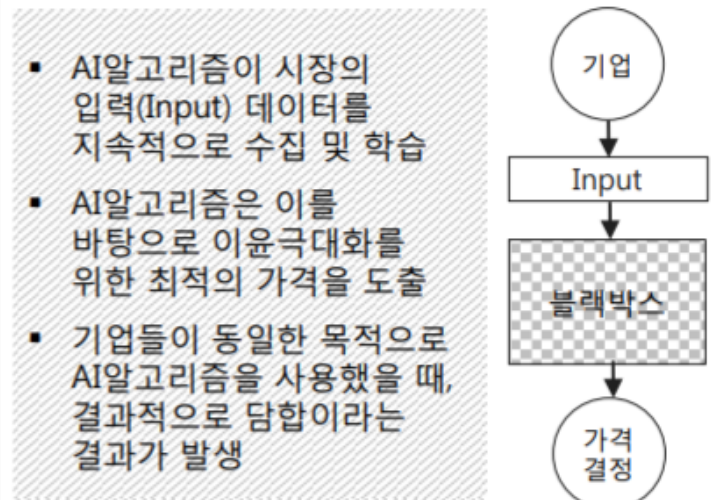
병행 알고리즘 (Parallel Algorithms)



신호 알고리즘 (Signaling Algorithms)

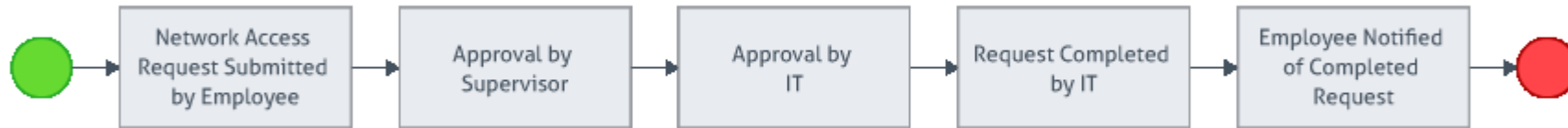


자가학습 알고리즘 (Self-learning Algorithms)



BPM이란?

BPM의 핵심 역량 BPM은 프로세스를 발견 (discover), 설계 (design), 적용 (deploy), 실행 (execute), 상호작용 (interact with), 운영 (operate), 최적화 (optimize) 및 분석 (analyze)하는 종합적인 역량을 의미한다. 다시 말해, Business process management는 프로세스의 전 lifecycle을 완벽히 컨트롤하는 것이다 (출처: The Emergence of Business Process Management)



Business Process

Business Process는 다음과 같은 특징을 가진다 (출처. The Emergence of Business Process Management).

- 크고복잡함: Material, Information, Business Commitments를 포함
- 고객의 다양한 니즈와 변화하는 시장 상황에 대한 대응하기 위한 역동성
- 비즈니스 Entity내 또는 Entities들간에 걸친 넓게 분산되고 커스터마이즈 되어 있으며 때로 다양한 Application과 다양한 기술플랫폼에 걸쳐 있음
- 하나의 프로세스는 일반적으로 몇 달 혹은 몇 년 간에 걸쳐 운영됨
- 속도와 신뢰성을 위하여 가능하다면 자동화됨
- 인간의 지적능력과 판단에 의존함
- 보이지 않음: 일부 회사에서는 프로세스는 인식되지도 않고 명시적이지도 않으며 문서화되어 있지 않고 암시적이며 조직의 역사에 포함되어 있음

이런 비즈니스 프로세스를 관리한다는 것은 무슨 의미일까? 실제로 비즈니스 프로세스의 관리는 명시적이지는 않지만 기업 내에서 계속 이루어지고 있는 것이다. 예를 들어, 어느 기업이 컨설팅회사로부터 BPR (Business Process Reengineering) 프로젝트를 수행한다고 가정하자. 이것은 현행 비즈니스 프로세스의 성과를 분석 (analysis)하고 프로세스를 최적화 (optimization)하기 위하여 재설계 (redesign)하는 과정을 의미한다. 이런 프로세스의 분석, 최적화 및 재설계와 같은 과정이 바로 BPM인 것이다.

비즈니스 어플리케이션은 프로세스 관리를 위한 지원도구이며 이것을 지속적으로 발전시켜 나가는 노력에 의하여 새로운 CRM 솔루션, SCM 솔루션, ERP, EAI 및 Workflow 솔루션이 개발되어 왔다. 이런 솔루션들은 비즈니스 프로세스를 부분적으로 지원하며 특히 프로세스 관리의 지원 기능은 부족하며 프로세스 관점 보다는 어플리케이션 수준에서 기능이 대부분이다. 이런 점에서 비즈니스 프로세스를 설계/ 정의하고 실행하고 모니터/분석하고 최적화해 나갈 수 있는 비즈니스 역량인 BPM의 등장은 프로세스가 고객, 비즈니스 파트너, 공급자에게까지 확대되고 있는 비즈니스 환경 하 - 예를 들어 Collaboration 및 Ubiquitous 통합과 같은 현상 - 에서 어쩌면 당연한 결과라고 할 수 있다. 다음 본론에서는 현재까지 진행되어 온 많은 기술 요소들이 어떤 의미에서 BPM과 관련을 가지고 있는지 살펴보도록 하겠다.

BPR

BPMN - <https://www.slideshare.net/gisoopa/ch7-bpmn>

SOA (Service Oriented Architecture)

SOA는 내부의 단위 업무를 서비스로 정의하고 이를 조합하여 하나의 어플리케이션을 구성하는데 이러한 어플리케이션을 컴포지트 어플리케이션(Composite Application)이라고 한다.

<https://camunda.com/>

<https://medium.com/@goldfing/%EB%B9%84%EC%A6%88%EB%8B%88%EC%8A%A4-%ED%94%84%EB%A1%9C%EC%84%B8%EC%8A%A4%EB%A5%BC-%EA%B7%B8%EB%A6%AC%EC%9E%90-bpmn-2-0-3730b3295dcf>

BPMN 예제

BPMN의 구성

BPMN은 3가지로 구성되어 있다.

Basic

가장 기초적인 구성이다. 도형도 간단하고 기술자가 아닌 사람들도 10분 정도만 인지하고 바로 적용할 수 있을 정도로 쉽다.

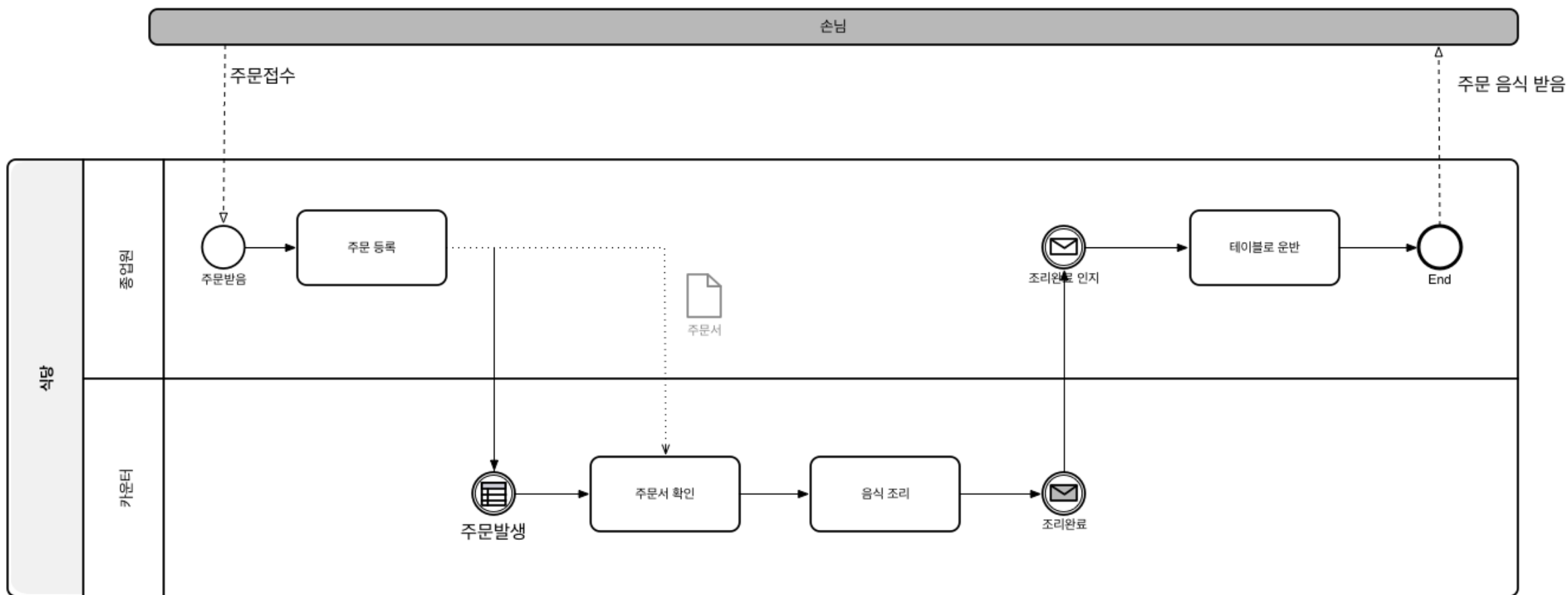
Core

SW 엔지니어들이 보기 시작하는 단계이다. Basic에서 좀더 세분화된 구성이다. 시스템을 구현할때 필요로 하는 Activity에 대한 세부 분류가 표현 가능하고, 예외 처리도 표현 가능하다.

Advaned

실제 코드를 다이어그램으로 표현하는 수준이다. Core보다 더 SW 엔지니어링에 가깝도록 세분화 되어 있다. 이 수준은 BPM 시스템에 구현하면 실제로 동작하는 단계이다.

이 포스팅에서는 가장 많이 사용하는 Element를 기준으로 기술한다.



Why should I care about DMN?

Decision Model and Notation (DMN) is an industry standard for modeling and executing decisions that are determined by business rules. DMN has been published in 2015 and is currently seeing a very quick adoption. These are the reasons:

Standard

DMN is not owned by a certain enterprise but by an institution ([OMG](#)), which is already established through other world-wide standards, e.g., BPMN and UML. The DMN standard is supported by several software products; you are less dependent on any particular vendor's products.

Direct Execution

In DMN, decisions can be modeled and executed using the same language. Business analysts can model the rules that lead to a decision in easy to read tables, and those tables can be executed directly by a decision engine (like Camunda). This minimizes the risk of misunderstandings between business analysts and developers, and it even allows rapid changes in production.

Experience

DMN as a standard is young, but it was developed by people having decades of experience with business rule management. Even though, the standard does not dictate any special implementation patterns, allowing for more modern and lightweight implementations than traditional business rule engines.

This tutorial provides a quick introduction into DMN, as it is defined in version 1.1.


<https://github.com/camunda/camunda-bpm-examples/tree/master/dmn-engine/dmn-engine-java-main-method>

<https://camunda.com/dmn/>

The example uses a decision table from the [DMN tutorial](#) to decide which dish should be served to our guests for dinner:

Dish
decision

Hide details



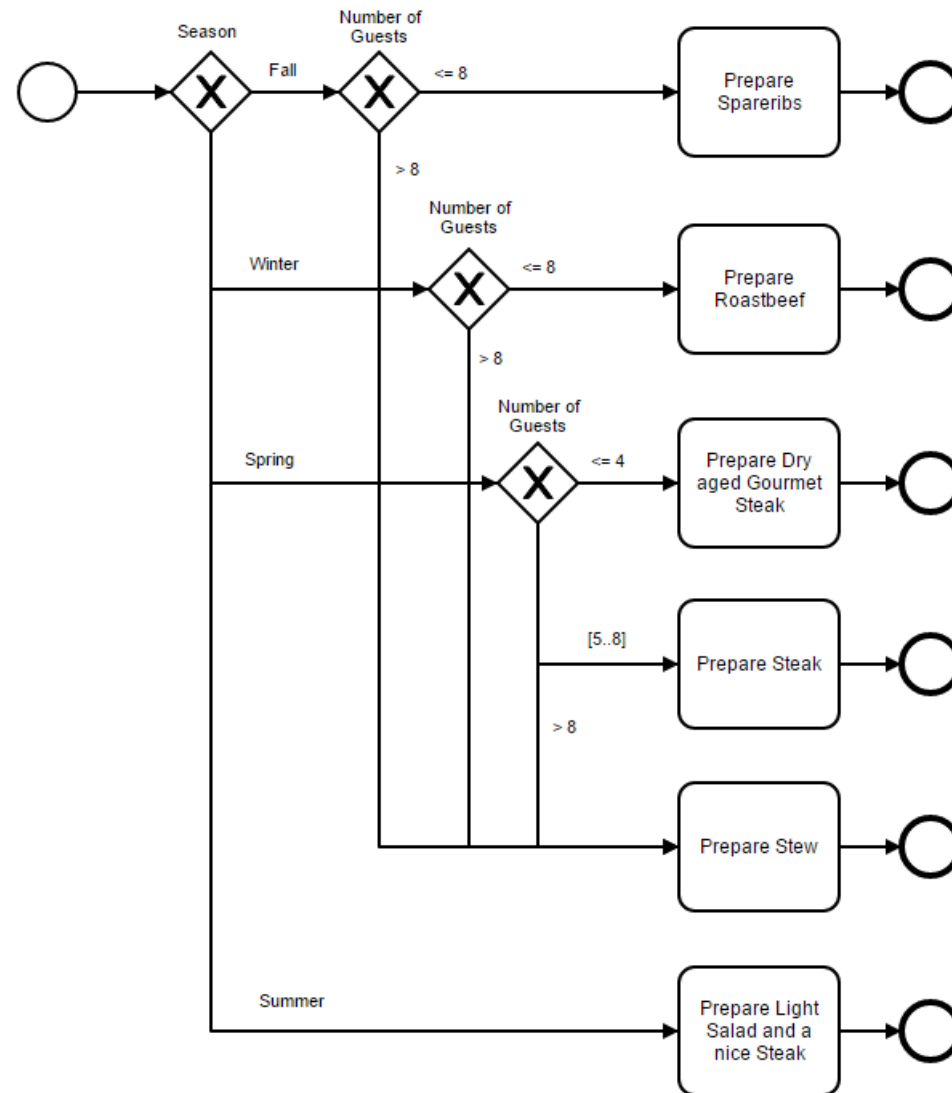
| U | Input + | | Output + | Annotation |
|---|----------------------------|-----------------|--------------------------------|----------------|
| | Season | How many guests | Dish | |
| | season | guestCount | desiredDish | |
| | string | integer | string | |
| 1 | "Fall" | <= 8 | "Spareribs" | - |
| 2 | "Winter" | <= 8 | "Roastbeef" | - |
| 3 | "Spring" | <= 4 | "Dry Aged Gourmet Steak" | - |
| 4 | "Spring" | [5..8] | "Steak" | Save money |
| 5 | "Fall", "Winter", "Spring" | > 8 | "Stew" | Less effort |
| 6 | "Summer" | - | "Light Salad and a nice Steak" | Hey, why not!? |
| + | - | - | - | - |

DMN and BPMN Processes

Perhaps you're thinking:

Hey, why should I use DMN anyway, I can express those rules with [BPMN gateways](#)!

If we express the example above in BPMN, it looks like this:



<https://thrillfighter.tistory.com/538>

Python win32api Window Programming

However Robin aims to go a step further. Stavropoulos' ambition is for Robin to be implemented as standard across the market, so customers can easily migrate bot definitions from one vendor's platform to another.

DMN and CMMN Cases

<https://camunda.com/dmn/>

Simulator

<https://camunda.com/dmn/simulator/>

<https://www.visual-paradigm.com/guide/cmmn/what-is-cmmn/>

BPMN vs CMMN

In recent decades, there has been a focus on modeling and automating well-structured and routine processes. BPMN is best used for well-structure and highly predictable work where knowledge workers mainly execute tasks, while CMMN covers the section of less predictable processes with the active involvement of knowledge workers making decisions and planning during run-time

Case management (CM) was introduced as a tool for knowledge workers by van der Aalst in 2005. In May 2014, the OMG published a [standard for case management called Case Management Model and Notation \(CMMN\)](#). Its focus is on supporting unpredictable, knowledge-intensive and weakly-structured processes. Case management is a type of business process technology that does not use control flow to describe the process.

Case management is about empowering knowledge workers by providing them with access to all the information concerning the case and giving them discretion and control on how a case evolves. Case management it is not about the process, it is about the workers. In contrast to classic processes, a certain goal and providing possibilities to choose from is more important than the way to achieve the goal itself.

Here listed the differences between BPMN and CMMN:

| Most of BPMN Notations | CMMN Notation |
|---------------------------------|-------------------------------------|
| Imperative | Declarative |
| Process centric | Data centric |
| Arcs describe the sequence | No predefined sequence |
| Guided work (head down workers) | Enables workers (knowledge workers) |
| Everything is modeled | Not everything is modeled |

Declarative Notation does not attempt to model the flow of a problem; they establish desired results i.e. specifying what they want to happen but not how it should happen. SQL is an example of declarative programming because it does not attempt to control the flow of a program; it simply states what it would like to appear but not how it is done.

Imperative Notation, on the other hand, do attempt to model the flow of a problem; for example, imperative programming languages such as Java or C++, they establish commands that will tell the compiler how they wish the code to run but not explicitly what they want to happen.

Business Process Management (BPM)

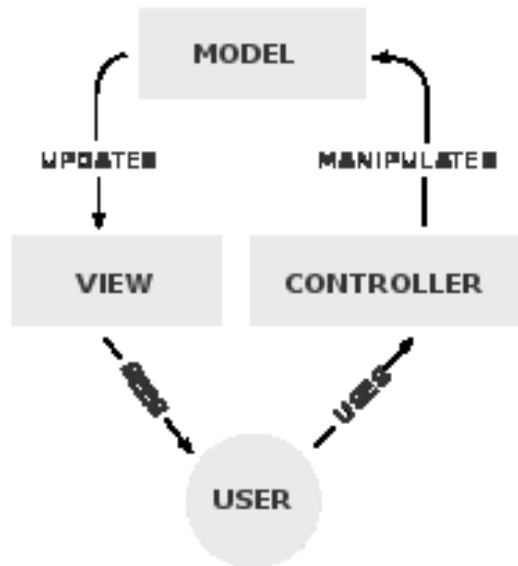
Business process management (BPM) is a systematic approach to making an organization's workflow more effective, more efficient and more capable of adapting to an ever-changing environment. A [business process](#) is an activity or set of activities that will accomplish a specific organizational goal.

The goal of BPM is to reduce human error and miscommunication and focus stakeholders on the requirements of their roles. BPM is a subset of [infrastructure management](#), an administrative area concerned with maintaining and optimizing an organization's equipment and core operations.

BPM is often a point of connection within a company between the line-of-business ([LOB](#)) and the IT department. Business Process Execution Language ([BPEL](#)) and Business Process Management Notation ([BPMN](#)) were both created to facilitate communication between IT and the LOB. Both languages are easy to read and learn, so that business people can quickly learn to use them and design processes. Both BPEL and BPMN adhere to the basic rules of programming, so that processes designed in either language are easy for developers to translate into hard code.

MVC (Model-view-controller) frameworks

Model-view-controller (MVC) is a [software design pattern](#) for implementing [user interfaces](#) on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.^{[1][2]} Traditionally used for desktop [graphical user interfaces](#) (GUIs), this architecture has become popular for designing [web applications](#) and even mobile, desktop and other clients.^[3]



Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) has several core ideas that should be addressed in your organization's SOA journey:

A set of services that a business wants to provide to their customers, partners, or other areas of an organization

An architectural style that requires a service provider, mediation, and service requestor with a service description

A set of architectural principles, patterns and criteria that address characteristics such as modularity, encapsulation, loose coupling, separation of concerns, reuse and composability

A programming model complete with standards, tools and technologies that supports web services, REST services or other kinds of services

A middleware solution optimized for service assembly, orchestration, monitoring, and management

With the convergence of mobile, social, cloud, and big data analytics, SOA is more important than ever before for offering insight and integrating systems from end to end.

By applying Service Oriented Architecture principles, an enterprise can manage and govern business and IT transformation, setting them apart from their competitors.

The benefits range from seamless integration, cloud enabled solutions, holistic business insight and agility to externalized APIs. SOA integrates the front office, back office and the Internet of Things.

Middleware, best practices and patterns speed the Service Oriented Architecture journey and amplify the value it creates.

BPMN

Just like companies like Alfresco, Flowable and Camunda, which all offer open-source implementations of the BPMN standard for describing business processes, Softomotive will pursue an "open core" business model.

<https://camunda.com/> Process Automation Reinvented for the Digital Enterprise

<https://blog.camunda.com/post/2016/08/dmn-performance-improvements/>

Why should I care about DMN?

Decision Model and Notation (DMN) is an industry standard for modeling and executing decisions that are determined by business rules.

<https://github.com/camunda/camunda-bpm-examples/tree/master/dmn-engine/dmn-engine-java-main-method>

<https://www.linkedin.com/pulse/softmotives-open-source-rpa-robin-fly-neil-ward-dutton>

Robin aims to go a step further. Stavropoulos' ambition is for Robin to be implemented as standard across the market, so customers can easily migrate bot definitions from one vendor's platform to another.

There are two levels of value that Robin (or an open-source initiative like it) can deliver to the vendor making the move.

The first level of value is simply about seeding the market. An open-source implementation means a free implementation, and in most cases a free implementation without restrictions. That means that users will be able to download the technology for free, use it wherever they like, to whatever extent that they like, and build on or embed the code into other products or services. There's clearly a market development benefit for Softomotive here that goes beyond a simple free product version, which would typically come with deployment restrictions or usage time limits. I can see systems integrators and BPO providers potentially taking advantage of this.

the second, bigger potential gain – the adoption of Robin as a true technology standard in the market - will be much more difficult to achieve. This benefit that would only come if Softomotive were able to convince a range of RPA technology vendors to actively adopt and contribute to the standard – that means UiPath, Automation Anywhere, Blue Prism, Workfusion, Kofax, Pegasystems and others. Going back to the BPMN standard I mentioned above: that effort was instigated by vendors that collectively held well over 50% of the market for BPM platforms – and was ultimately shepherded by the neutral OMG standards body.

01 장 환경구축과 기본 사용법

Chapter 1. 환경구축

Chapter 2. 기본문법

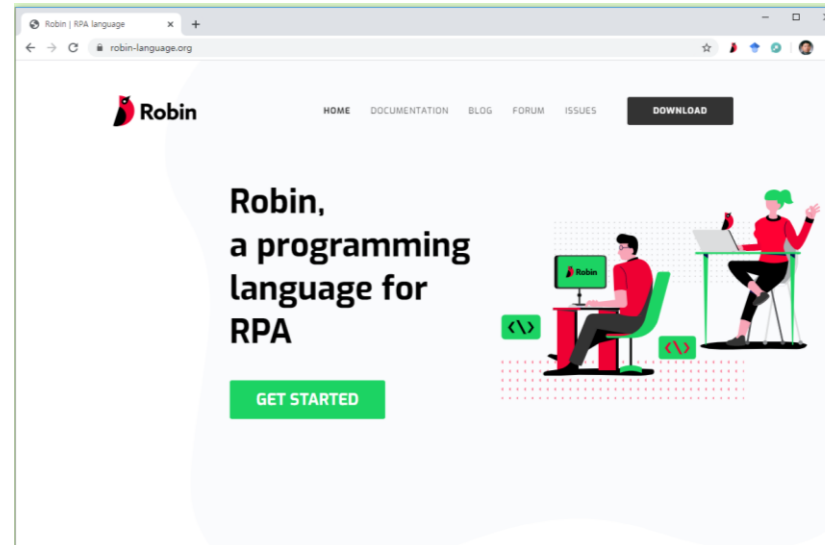
Chapter 3. 에디터, UI Spy 및 CLI 사용법

Robin 설치

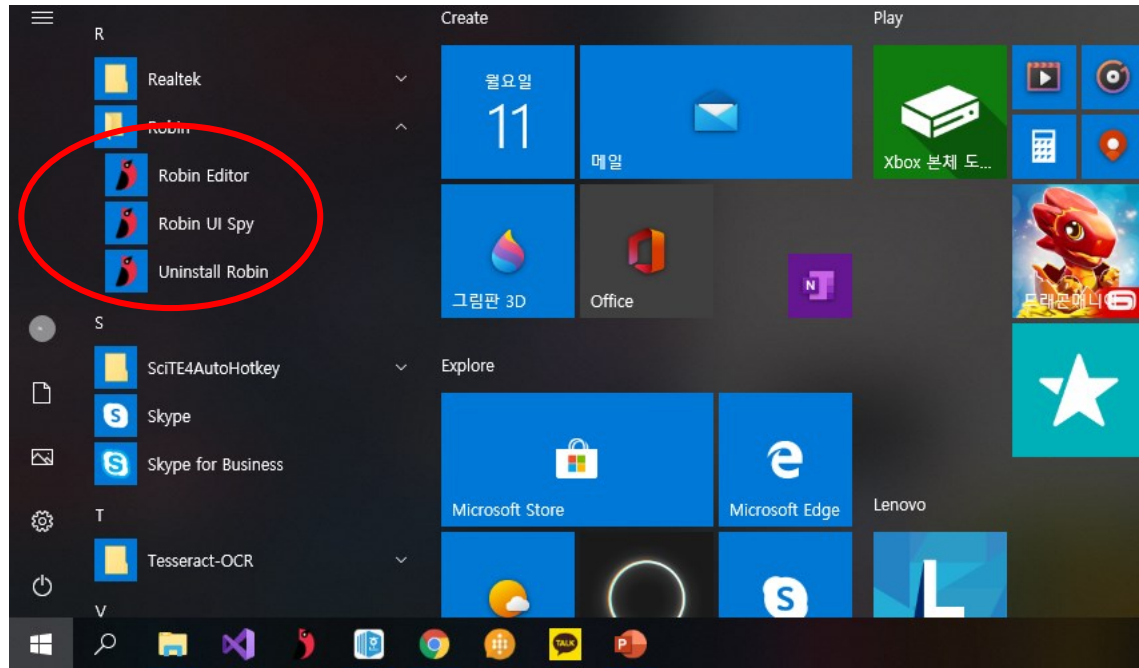


URL <https://robin-language.org/>

1. Robin Editor
2. Robin UISpy
3. Robin CLI



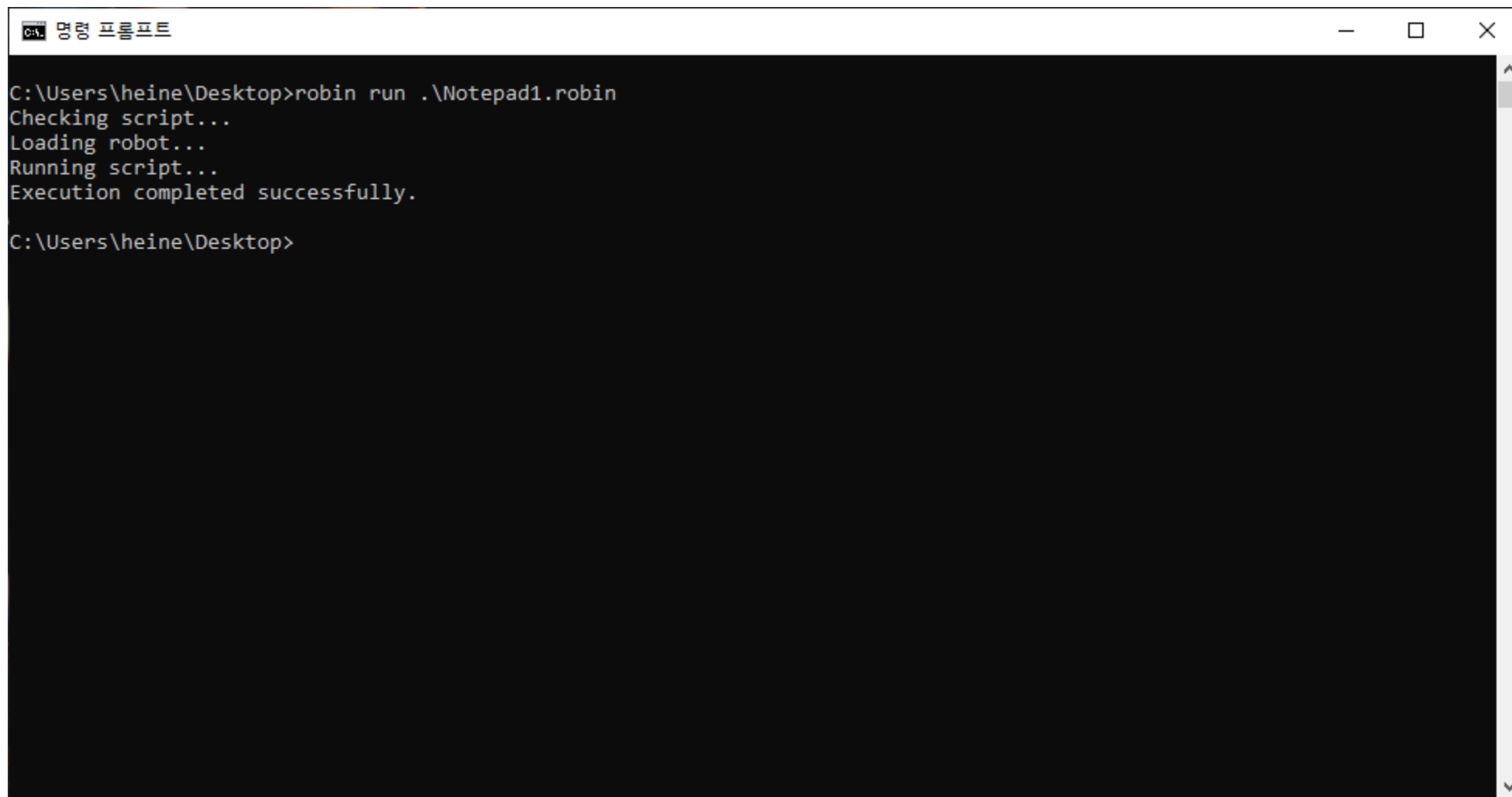
Robin은 아래 링크에서 다운로드 할 수 있습니다. 로빈 설치 중에 Robin Language는 Robin Editor, UISpy, Robin CLI 및 SDK와 함께 시스템에 설치됩니다.



Robin Editor
Robin UI Spy

정상적 프로그램 설치 확인

Robin CLI 실행 방법



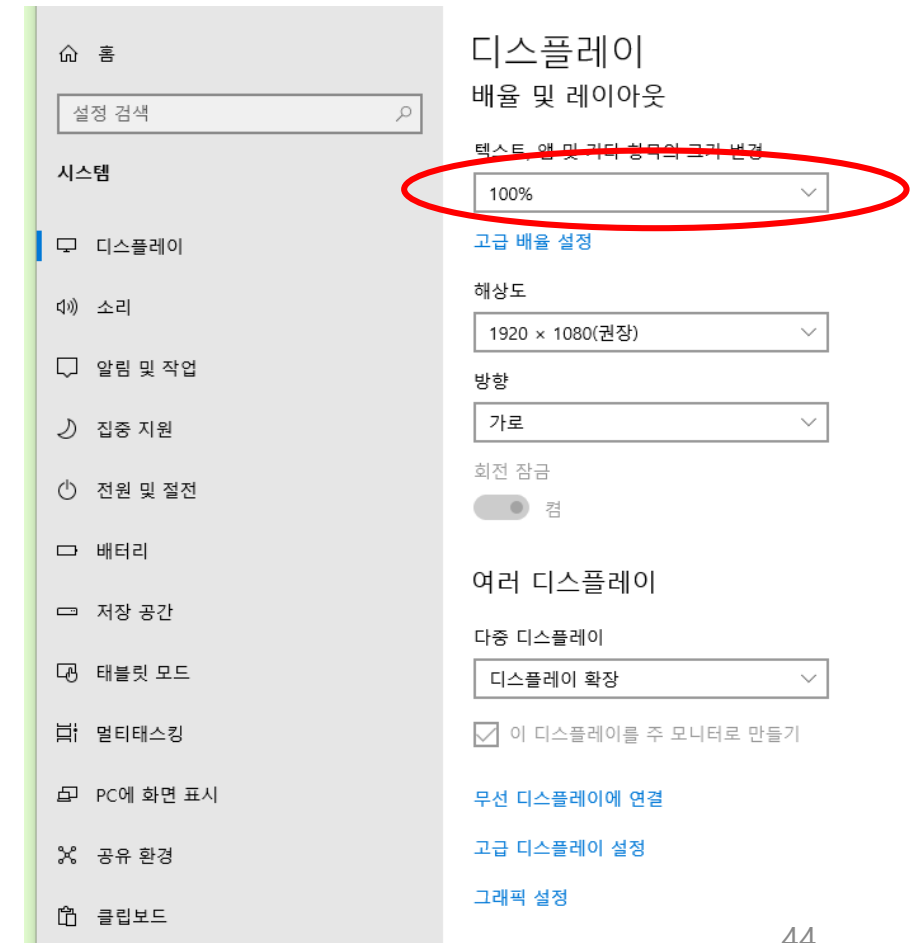
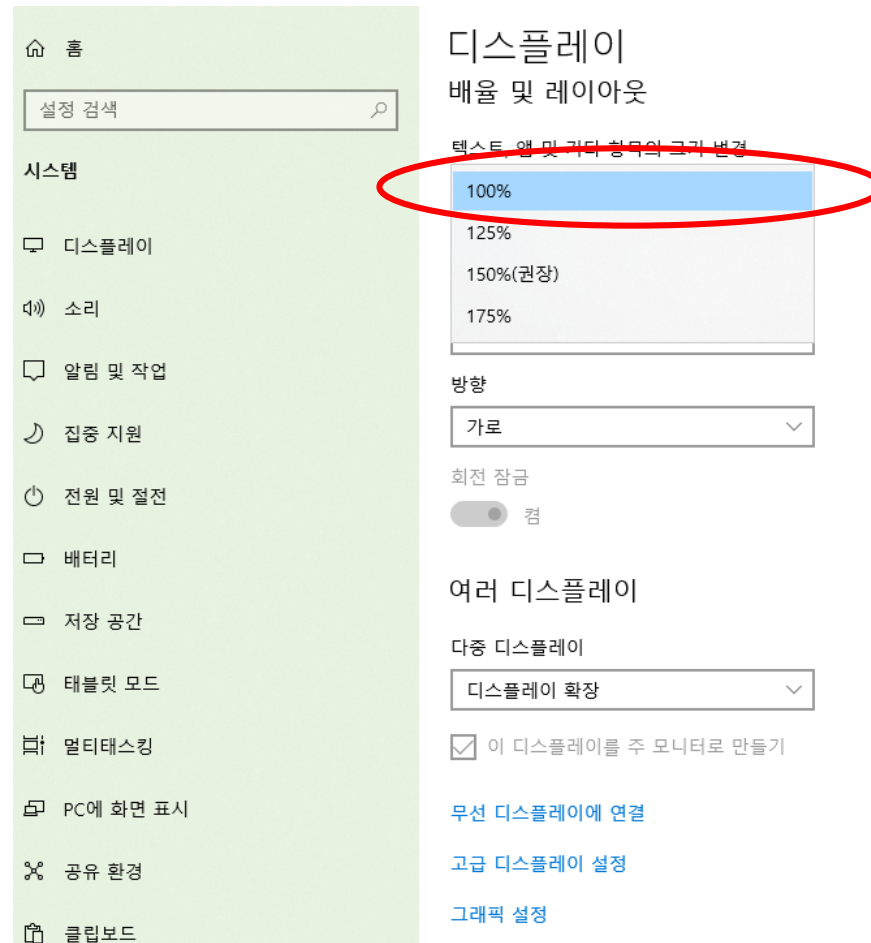
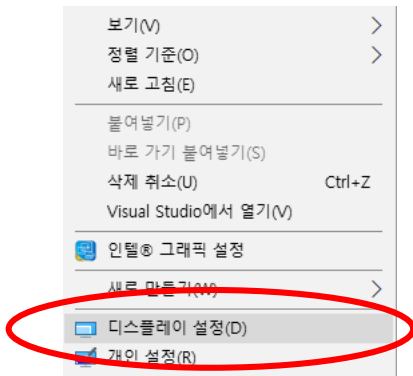
A screenshot of a Windows Command Prompt window titled "명령 프롬프트" (Command Prompt). The window shows the execution of the command `robin run .\Notepad1.robin`. The output displays the following steps: "Checking script...", "Loading robot...", "Running script...", and "Execution completed successfully." The prompt then returns to the default directory `C:\Users\heine\Desktop>`.

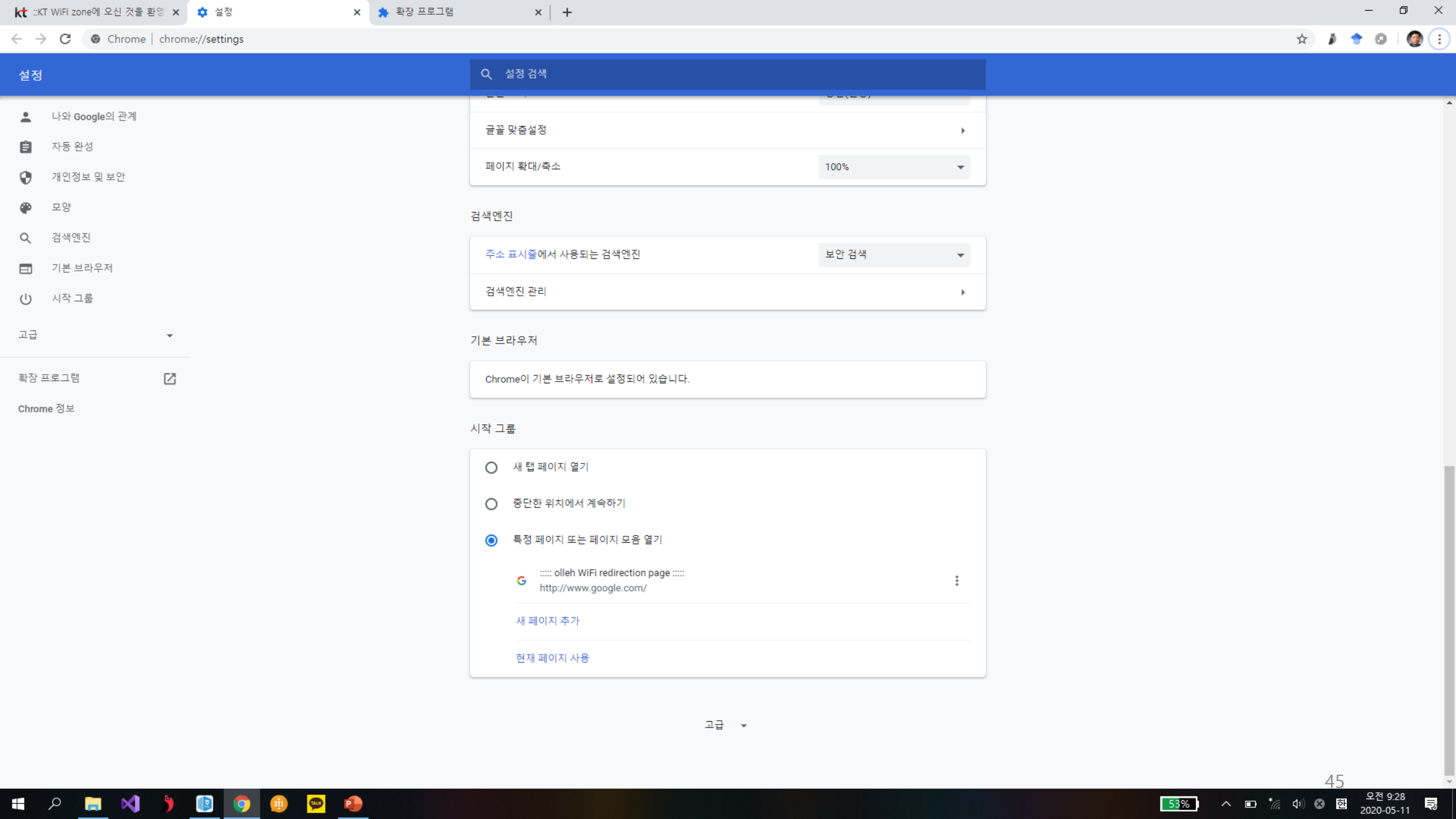
```
C:\Users\heine\Desktop>robin run .\Notepad1.robin
Checking script...
Loading robot...
Running script...
Execution completed successfully.

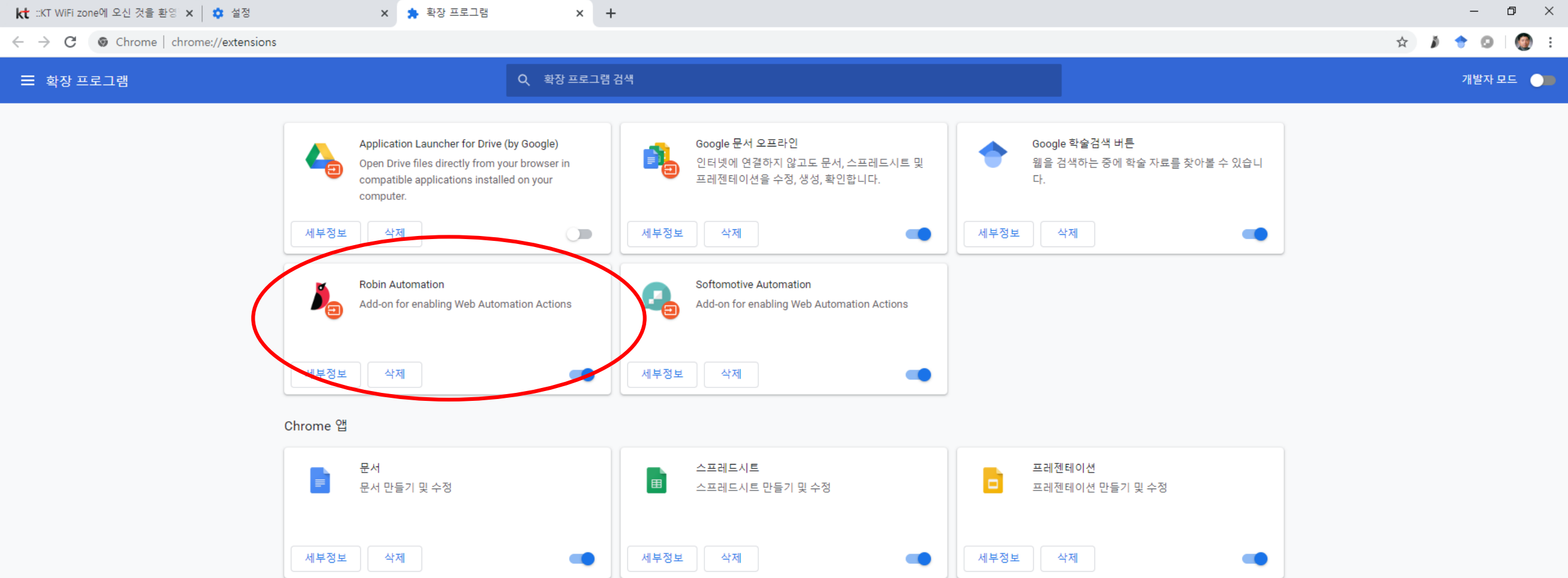
C:\Users\heine\Desktop>
```

Robin 실행 환경 점검

디스플레이 설정 -> 텍스트, 앱 크기 배율 설정











You need to add Softomotive add-on
Got to chrome Settings-> Extensions- Search for Softomotive add-on.

Robin Editor

Robin을 설치하면 Robin Editor도 설치됩니다. 자동화 개발 및 테스트를 위한 주요 도구입니다. 구문 강조, 오류 표시 및 자동 완성 기능을 제공하는 코드 편집기가 포함되어 있습니다. Robin Editor를 사용하면 자동화를 개발하고 구문 오류가 있는지 확인하고 실행할 수 있습니다. 상단 섹션에는 자동화 실행, 중지 및 확인 옵션이 있습니다. 하단 섹션에서 모든 오류와 콘솔 출력이 나타나는 출력 창을 찾을 수 있습니다. 화면 상단에는 표준 파일 및 편집 작업이 포함 된 메뉴와 도구 모음에서도 사용할 수 있는 자동화 실행, 중지 및 확인 옵션이 있습니다.

 File Edit View Run Help

hello.robin - Robin Editor

 Run  Stop  Check

```
1
2 set message to "Hello RPA world!"
3
4 Console.Write Message: message
```

OUTPUT

Checking script...

Loading robot...

Running script...

Hello RPA world!

Execution completed successfully.

Robin UI Spy

Robin Editor 외에도 UISpy도 설치됩니다.

UISpy 도구는 AppMasks 용 편집기입니다. UISpy를 사용하여 Appmask를 편집하고 생성 할 수 있습니다. UISpy의 맨 위 섹션에는 새 화면 및 컨트롤을 추가하는 옵션이 있습니다. AppMask 버전과 함께 응용 프로그램 이름을 설정하는 필드도 있습니다.

각 화면 또는 컨트롤 위로 마우스를 가져 가면 세 가지 옵션을 사용할 수 있습니다. 화면과 컨트롤을 각각 편집, 이름 바꾸기 또는 삭제합니다. UISpy의 오른쪽에 캡처 시점의 스크린 또는 컨트롤 스크린 샷이 나타납니다.

Robin CLI

Robin의 명령 줄 인터페이스도 설치에 포함되어 있습니다.

단순히 "robin"을 입력하여 명령 프롬프트에서 Robin을 호출 할 수 있습니다.

명령 줄을 통해 자동화를 실행하고 확인할 수 있도록 제공됩니다.

Robin의 명령 : 실행 및 확인을 사용하여 자동화 파일을 실행하고 확인하는 명령을 제공합니다

(CLI) 사용 예시

```
robin check "C:\Users\user\Desktop\Automation.robin"
```

```
robin run "C:\Users\user\Desktop\Automation.robin"
```

두 명령의 인수는 실행하거나 확인해야하는 로빈 자동화의 파일 경로입니다.

또한 -help 또는 -h 스위치를 통해 사용 가능한 도움말을 제공합니다.

Robin CLI는 오류 좌표를 보여주는 목록 형식으로 오류를 가져 오는 옵션도 제공합니다.

Robin Syntax 예시

몇 가지 주목할만한 특징

Robin은 매우 간단한 구문으로 사용하기가 매우 쉽습니다.
로빈은 대소 문자를 구분하지 않습니다
누구나 처음부터 모듈을 개발할 수 있으므로 Robin은 무한합니다.

구문

Robin은 모든 프로세스를 자동화하고 자동화로 변환 할 수있는 RPA 언어입니다.
이 섹션에서는 Robin에서 사용되는 구문에 대한 기본 사항을 볼 수 있습니다.

변수

Robin은 변수를 사용하여 정보를 저장합니다. 변수는 set assignment 문을 사용하여 직접 저장하거나 전용 작업을 사용하여 저장할 수 있습니다.
아래에서 "Set Variable"대 입문을 사용하여 NumVar 변수에 값 5가 할당 된 것을 볼 수 있습니다.

Set NumVar to 5

Robin Syntax 예시

두 번째 방법은 액션을 사용하는 것입니다. 아래에는 DateTime 모듈의 "현재"작업이 사용됩니다. 조치의 출력은 실행시 날짜 시간의 값입니다. 값은 나중에 참조 할 수 있는 "CurrentDateTime"변수에 저장됩니다.

```
DateTime.Current DateTimeFormat:DateTimeFormat.DateAndTime Offset:'0' CurrentDateTime=> CurrentDateTime
```

Input and Output of Actions

앞의 예에서 DateTime은 " CurrentDateTime " 변수에 저장되었습니다. 이것을 출력 액션이라고도 합니다. 일반적으로 조치에는 입력 및 출력이 필요합니다. 그러나 입력을 필요로 하지 않는 반면 다른 사람들이 출력을 생성하지 않을 수 있는 액션이 있습니다.

아래 명령에서 작업 표시 메시지는 메시지 상자를 통해 날짜 시간을 표시합니다. "CurrentDateTime" 변수를 통해 값이 전달됩니다.

이 액션에는 여러 입력 매개 변수가 포함됩니다. 이 중 두 개의 입력 매개 변수가 사용 중입니다. 제목과 메시지. 제목은 따옴표를 사용하여 채우고 메시지는 변수를 사용하여 채웁니다.

Actions Structure

액션은 모듈에 속합니다. 아래 구조 예시와 같습니다.

이 경우 사용되는 작업은 제목과 대화 상자 메시지가 필요한 "Display Input Dialog"입니다. 모듈에서 조치를 선택하면 다음 명령이 자동 생성됩니다.

(Robin Editor)

```
Display.ShowDialog Title:" Message:" DefaultValue:" InputType:InputType.SingleLine IsTopMost:False  
UserInput=> UserInput ButtonPressed=> ButtonPressed
```

위의 액션을 분석하면,

| Command | Definition |
|--------------------------------|--|
| Display | The action belongs in the Display Module. |
| InputDialog | The action is called Input Dialog. |
| Title:" | The first input parameter is the Title and should be populated in the quotes. |
| Message:" | The second input parameter is the Message that appears to the user and should be populated in the quotes. |
| DefaultValue:" | The third input parameter is the default value appearing in the input field. |
| InputType:InputType.SingleLine | As a fourth parameter we need to identify the field to be populated. This can be single line, single line with password or multi line. |
| IsTopMost:False | An option to keep the message box always on top is the final Input parameter. |
| UserInput=> UserInput | The first output variable this action produces is the input that the user populated and it's saved in the UserInput variable. |
| ButtonPressed=> ButtonPressed | The second output parameter is the button that the user pressed. |

Reserved Words

아래에서 변수 함수 이름 및 레이블로 허용되지 않는 예약어 목록을 찾을 수 있습니다.

| erm | Usage |
|----------|---|
| action | The term action is used in the exception handling of an action when it needs to be repeated. |
| and | The term and is used in conditional operators. |
| block | The term block is used to be able to disable a block (i.e. function, conditional, loop). |
| call | The term call is used to control the flow of the automation by calling a function. |
| case | The term case is used to form conditions. |
| default | The term default is used to form conditions. |
| disable | The term disable is used to disable a particular command so that it's not performed during the execution. |
| else | The term else is used to form conditions. |
| end | The term end is used to mark the end of a block. A block can be a conditional, a loop or a function. |
| error | The term error is used in the exception handling of an action. It's used to throw an error. |
| exit | The term exit is used to terminate the automation, a function or a loop. |
| false | The term false is used as a boolean operator. |
| for | The term for is used in conjunction with the wait term to set the duration the automation should wait until moving to the next action. |
| foreach | The term foreach is used to control repetitions. |
| from | The term from is used to control repetitions. |
| function | The term function is used to interact with functions in an automation. |
| goto | The term goto is used to control the flow of the automation. |
| if | The term if is used to form conditions. |
| import | The term import is used to enable addition of external resources to the automation. |
| in | The term in is used in conditional operations related to lists and strings. |
| label | The term label is used to control the flow of the automation. |
| loop | The term loop is used to create repetitions. |
| main | The term main is used by the main function. |
| mod | The term mod is used in mathematical operations. |
| next | The term next is used to control repetitions. |
| no | The term no is used as a boolean operator. |
| not | The term not is used in conditionals operators. |
| on | The term on is used in the exception handling of an action. It's used in conjunction with the word error when it's used. |
| or | The term or is used in conditional operators. |
| output | The term output is used to make variable values of a function available to the rest of the process. |
| repeat | The term repeat is used in the exception handling of an action. It's used when an action should be repeated. |
| set | The term set is used to assign a value to a variable. |
| step | The term step is used to control repetitions. |
| switch | The term switch is used to form conditions. |
| then | The term then is used to form conditions. |
| throw | The term throw is used in the exception handling of an action. It's used to throw an error. |
| times | The term times is used in the exception handling of an action. Its used to define the number of times the action should attempt to run. |
| to | The term to is used while a value is assigned to a variable as well as when to control the number of repetitions. |

Default Data Types

<https://docs.robin-language.org/robin-the-rpa-language/default-data-types/>

Variables Manipulation

<https://docs.robin-language.org/robin-the-rpa-language/variables-manipulation/>

Looping

Functions

In Robin, automations can be split in functions. That way, sections that are repeated can be developed once, but they are set to run multiple times. A function is marked with the "function" term followed by the function name.

In the below example function "today" contains the "Get Current Date and Time" action.

(Robin Editor)

function today

**DateTime.Current DateTimeFormat:DateTimeFormat.DateAndTime Offset:'0' CurrentDateTime=> CurrentDateTime
End**

A function can then be called using the term "call" followed by the function name.

(Robin Editor)

call today

Similarly, as in the above example, value1 and value2 are two variables which are not used in the automation at all, the value total is used within the function, but it cannot be used in the automation in general. The output value calculated within the function, needs to be exported as a function output. This is a prerequisite to be able to utilise the result in the general automation, or on another function.

To set the output variables, use the name of the variable in the function, arrow (=>) and then the variable used in the automation. Apart from there, all the output variables need to be stated in the function definition line as well.

(Robin Editor)

```
set value1 to 10
```

```
set value2 to 5
```

```
function addition num1, num2, output total, output difference
```

```
    set total to num1 + num2
```

```
    set difference to num1 - num2
```

```
end
```

```
call addition num1:value1 num2:value2 total=>output1 difference=>output2
```

Global Variables

Alternatively, using the "g:" prior to a variable in a function, marks the variable as global. That way the variable is not required to be set as output. The above example would look like this.

(Robin Editor)

```
set value1 to 10  
set value2 to 5
```

```
function addition num1, num2  
    set g:total to num1 + num2  
end  
call addition num1:value1 num2:value2
```

Variables and Functions

<https://www.laserfiche.com/ecmblog/what-is-the-difference-between-robotic-process-automation-rpa-bpm/>

BPM vs RPA

Definition of RPA

Robotic Process Automation is a software technology that enables employees to better focus on high priority tasks by pushing routine, monotonous tasks to software “robots” to complete. These robots work directly across application user interfaces, automatically inputting data and triggering actions across multiple systems acting on the behalf of an employee. As a business user-friendly tool that does not involve any programming, robotic process automation technology enables non-technical professionals to self-serve and configure robots for themselves to solve their own automation challenges.

Definition of BPM

BPM is a strategic approach that concentrates on reshaping an organization’s existing business processes to achieve optimal efficiency and productivity. BPM software is the foundational backbone to facilitate completion of an organization’s projects, providing a variety of tools to help improve and streamline how business processes are performed. BPM software components can include business analytics, workflow engines, business rules, [web forms](#), and collaboration tools. Often large-scale projects, BPM process improvement initiatives can impact an organization’s technology, employees, and customers, bringing about significant transformational returns.

BPM and RPA as a Combined Approach

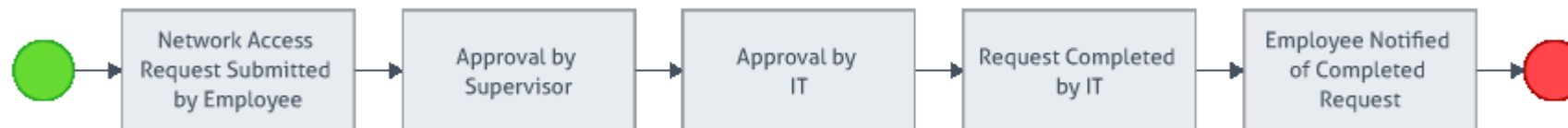
Given their similar goals of increasing productivity and efficiency, **BPM and RPA are not competing approaches and work together in harmony.** RPA can be a valuable tool in boosting gains achieved with a traditional BPM system.

In many cases, an organization may not always have the resources and time to fully implement a completely integrated and automated process or have a process that does not require any human involvement. Implementing BPM and RPA together can bring about much greater process automation and value that one technology approach alone may not be as positioned to achieve.

Example of BPM & RPA in Action: Network Access Request

Members of the IT department are involved with a number of highly important responsibilities for the organization but may be hindered by a number of repetitive, manual tasks that do not require complex decision making.

Below is a simplified version of an automated network access request process where an employee submits a request to be granted access to or be removed from a certain software application. After being approved by a supervisor, IT can then review the request and complete the request as appropriate.



Key Differences between RPA & BPM [Text Version]

| | Robotic Process Automation | Business Process Management |
|--------------------------|--|--|
| Technology | Software Technology Software “bots” are configured to complete routine , monotonous tasks that an employee would normally have to do | Holistic Technology Encompasses a wide range of software technology components including business analytics, workflow engine, and more to optimize business processes for maximum value and efficiency |
| Automation Focus | Manual Tasks Minimizing manual, repetitive, and rule-based tasks that do not require complex decision making | End-to-End Process Automation Re-engineering process flows to eliminate bottlenecks, connect systems, and increase productivity enterprise-wide |
| Deployment Effort | Low Non-disruptively works across an organization’s existing processes and applications without requiring coding or extensive training | High A longer-term effort which could require dedicated technical resources, depending on process complexity and depth of integrations |
| Business Impact | Quick & Immediate Returns can be realized quickly and cost-effectively, but implementations may not always address underlying process inefficiencies | Significant & Transformational Wide gains can be achieved in overall productivity, agility, cost reduction, efficiency, and compliance |

Examples and Codes

<https://github.com/JSJeong-me/RPA>

삼성 평택2라인 낸드 강화..의미심장 키워드 셋

2021년 하반기 양산 돌입



[이코노믹리뷰=최진홍 기자] 삼성전자가 평택캠퍼스 2라인에 8조원을 투입, 낸드플래시 생산라인을 구축하는 투자를 단행한다고 1일 밝혔다. 메모리 반도체 시장의 업황 악화가 예상되는 가운데 나온 강력한 투자 본능이다. 나아가 삼성전자의 반도체 로드맵 전반의 의미있는 변화에도 시선이 집중된다.

키워드 둘, 중국과의 연결고리

삼성전자의 평택캠퍼스 2라인 낸드플래시 생산라인 구축을 두고 업계에서는 중국 거점과의 유기적인 연결고리에도 주목하고 있다.

삼성전자는 국내에는 화성과 평택, 해외에는 중국 시안에 낸드플래시 생산라인을 운영하고 있다. 이런 가운데 이재용 부회장은 지난달 18일 삼성전자 메모리사업부장 사장, 박학규 DS부문 경영지원실장 사장, 황득규 중국삼성 사장 등과 함께 중국 시안 반도체 공장을 전격 방문했다.

1. 최종 선발결과

(단체)

- 선발결과 : 5개 단체 합격

| 연번 | 단체명 | 담당자 | 연번 | 기관명 | 담당자 |
|----|---------|-----|----|---------|-----|
| 1 | 고려대학교 | 서O철 | 4 | 서울디자인재단 | 조O찬 |
| 2 | 배화여자대학교 | 박O열 | 5 | 강동구청 | 홍O길 |
| 3 | 건국대학교 | 김O경 | - | - | - |

※ 순서는 신청서접수순, 합격자 대상 2차 현장평가 실시

(개인)

- 선발결과 : 7명 합격(예비합격 2명)
- 합격자

| 연번 | 이름 | 휴대폰뒷자리 | 연번 | 이름 | 휴대폰뒷자리 |
|----|------|--------|----|------|--------|
| 1 | 송O리* | 1983 | 5 | 김O지 | 8910 |
| 2 | 이O연* | 7755 | 6 | 김O석 | 7202 |
| 3 | 정O수 | 3644 | 7 | 조O우* | 7132 |
| 4 | 이O석 | 0410 | - | - | - |

※ 가점 관련 증빙 대상자는 이름 옆에 ‘*’표시함(증빙서류 제출 대상)

RPA를 활용한 OFFICE업무 자동화

교육 개요

- 교육기간 : 2020.06.13(토). 06.20(토), 06.27(토) 토요 3일과정
- 교육시간 : 오전 9시 ~ 오후 6시 (1일 8시간, 총24시간)
- 교육정원 : 최대 15명 (10명이하인 경우 폐강될 수 있습니다)
- 교육장소 : @camp 대학로 6층
- 교 육 비 : 전액무료 (교재비, 중식 제공) 단, 중소기업재직자에 한함 (대기업재직자 10만원/학생 수강불가)

교육 특징

전액무료교육

Office업무 자동화를 위한 교육

IT신기술교육

대학로교육장

교육 목표

IT비전문가가 RPA전문가가 되어가는 과정

- 사무 환경에서 수행되는 프로젝트를 ① 소규모 프로세스로 분할하고 난이도와 RPA 적합성을 평가하는 방법 ② 자동화된 프로세스를 통해 구현되는 업무 편의성을 측정하는 방법 ③ 업무 요구 사항에 따라 RPA 솔루션을 개발할 수 있도록 포괄적인 프로세스를 관리하는 방법을 습득
- 전체 RPA 구현 과정에서 업무 전문가 및 관리자가 수행하는 역할과 RPA 솔루션 도입을 통해 비즈니스 요구 사항을 효과적으로 충족하는 방법을 구현
- RPA 구현의 여러 단계, 자동화에 적합한 Tool을 선정하는 방법, RPA 프로젝트를 진행하는 방법 및 조직 전반에서 RPA 적용 효율 극대화를 위한 방안 제시

정 준 수 / 컨설팅학박사 (연락처: 010 - 5359 - 3644)

□ 학 력

1987년 2월 고려대학교 전기공학과 학사

1989년 5월 New York Institute of Technology 전산학 석사 (Neural Networks 전공)

2011년 2월 서강대학교 경영전문대학원 MBA과정

2017년 2월 한성대학교 지식서비스&컨설팅대학원 박사

□ 경 력

○ 1989년 6월 ~ 1995년 1월 : 삼성전자 (삼성의료원-삼성생명과학연구소)

○ 1995년 2월 ~ 1999년 2월 : 삼성SDS (정보기술연구소)

○ 1999년 3월 ~ 2002년 2월 : NPION, Inc. (New York, USA)

○ 2002년 3월 ~ 2017년 7월 : (주)이미넷, (주)신기사

○ 2017년 3월 ~ 현재 : 한성대학교 대학원, 엡캠프 - AI, 머신러닝 및 인지과학(Cognitive Science) 강의

□ <https://github.com/JSJeong-me/RPA>

