

과제 1

20기 예비 김도현

목차

1. Topics
2. Services
3. Parameters
4. Actions

Topics

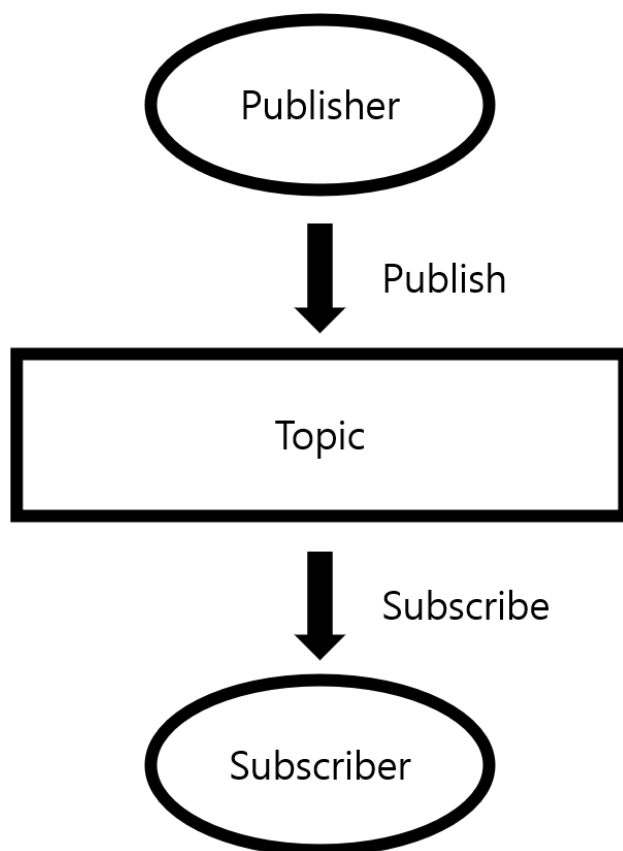
Topic은 ROS2 시스템에서 분할된 modular node간 메시지 교환에서 데이터의 전달 경로인 데이터 버스이다.

각 node는 topic에 데이터를 publish하거나 특정 topic을 subscribe하고 해당 토픽에 게시되는 데이터를 수신 받을 수 있다.

Topic 통신은 node와 node간의 일대일 통신(one-to-one communication)뿐만 아니라 하나의 topic 여러 node에서 수신 받거나 하나의 node에서 여러 topic의 데이터를 수신 받는 다대일 통신(one-to-many / many-to-one communication), 다대다 통신(many-to-many communication)이 가능하다.

Topic통신을 위해 publisher와 subscriber는 동일한 data structure를 사용해 message를 주고받아야 한다.

지속적인 업데이트가 필요한 연속적인 통신에서 사용된다.



Command

- `$ ros2 run rqt_graph rqt_graph`
node와 topic의 통신을 시각화 한다.
- `$ ros2 topic list`
현재 활성화된 topic의 목록을 출력한다.
- `$ ros2 topic list -t`
활성화된 topic의 목록의 topic type을 표시한다.
- `$ ros2 topic echo <topic_name>`
해당 topic에 게시되는 데이터를 출력한다.
- `$ ros2 topic info <topic_name>`
해당 topic의 type, publisher count, subscription count를 출력한다.
- `$ ros2 interface show <msg_type>`
message의 data structure를 출력한다.
- `$ ros2 topic pub <topic_name> <msg_type> '<args>'`
지정된 topic이름과 data structure를 사용하는 메시지를 publish한다.
Args로는 YAML syntax가 사용된다.
- `$ ros2 topic pub --once -w <n> --rate <n> <topic_name> <msg_type> '<args>'`

--once는 메시지를 한 번 게시하고 종료한다.

-w는 지정한 만큼의 subscriber가 subscribe할 때까지 대기한 후 게시를

시작한다.

다른 인수가 없을 경우 기본 1hz의 속도로 게시를 반복한다.

- `$ ros2 topic hz <topic_name>`

해당 topic의 게시 속도를 출력한다.

- `$ ros2 topic bw <topic_name>`

bandwidth 사용률, topic에 게시되는 message들의 수 등 해당 topic의 bandwidth정보를 출력한다.

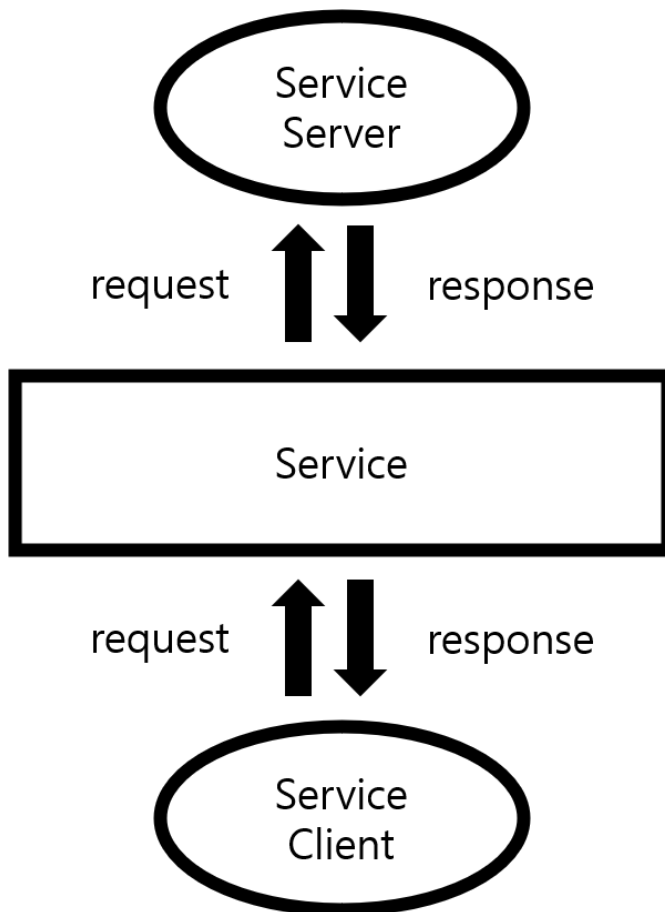
Subscriber의 기준에서의 기록하기 때문에 publisher에서 게시될 때의 bandwidth와는 차이가 있을 수 있다.

- `$ ros2 topic find <topic_type>`

해당 type의 data structure를 사용하는 topic 들의 목록을 출력한다.

Services

Service 통신에서 node는 service server와 service client로 구분되며 service client의 호출 시 service server에서 데이터를 게시한다.



Command

- `$ ros2 service list`

현재 활성화된 service 통신의 목록을 시각화한다.

- `$ ros2 service list -t`

활성화된 service의 목록의 service type을 표시한다.

- `$ ros2 service type <service_name>`

해당 service의 data structure를 출력한다.

Type Empty는 request와 response에 데이터가 포함되지 않음을 의미한다.

- `$ ros2 service find <type_name>`

해당 type의 data structure를 사용하는 service들의 목록을 출력한다.

- `$ ros2 interface show <type_name>`

해당 type의 data structure를 출력한다.

- `$ ros2 service call <service_name> <service_type> <arguments>`

해당 서비스를 호출한다.

Parameters

Parameter는 node를 구성하는 데이터이다.

key, value, descriptor로 구성된다.

key는 string value는 bool, int64, float64, string, byte[], bool[], int64[], float64[], string[]의 자료형을 가진다.

Descriptor는 비어 있거나 parameter 설명, 값 범위, type 정보 등을 저장한다.

Set parameter callback과 on parameter event callback을 설정할 수 있다.

Parameters

- /node_name/describe_parameters

Type: rcl_interfaces/srv/DescribeParameters

Descriptor 목록 반환

- /node_name/get_parameter_types

Type: rcl_interfaces/srv/GetParameterTypes

Parameter의 type 반환

- /node_name/get_parameters

Type: rcl_interfaces/srv/GetParameters

Parameter의 값 반환

- /node_name/list_parameters

Type: rcl_interfaces/srv/ListParameters

Parameter 목록 반환

- `/node_name/set_parameters`

Type: `rcl_interfaces/srv/SetParameters`

Parameter 설정

- `/node_name/set_parameters_atomically`

Type: `rcl_interfaces/srv/SetParametersAtomically`

Parameter 설정, 하나 실패 시 전체 실패 처리

Command

- `$ ros2 param list`

각 node의 parameters의 목록을 출력한다.

모든 node는 parameter `use_sim_time`을 가진다.

- `$ ros2 param get <node_name> <parameter_name>`

해당 parameter에 저장된 데이터를 출력한다.

- `$ ros2 param set <node_name> <parameter_name> <value>`

해당 parameter에 저장된 값을 변경한다.

- `ros2 param dump <node_name>`

해당 node의 모든 parameter를 출력한다.

- `ros2 param dump <node_name> > <parameter_file>`

해당 node의 parameter 전체를 yaml 파일로 저장한다.

- `$ ros2 param load <node_name> <parameter_file>`

활성화되어있는 node의 parameter를 파일에서 불러온다.

- `$ ros2 run <package_name> <executable_name> --ros-args --params-file
<file_name>`

지정된 parameter로 node를 활성화한다.

Actions

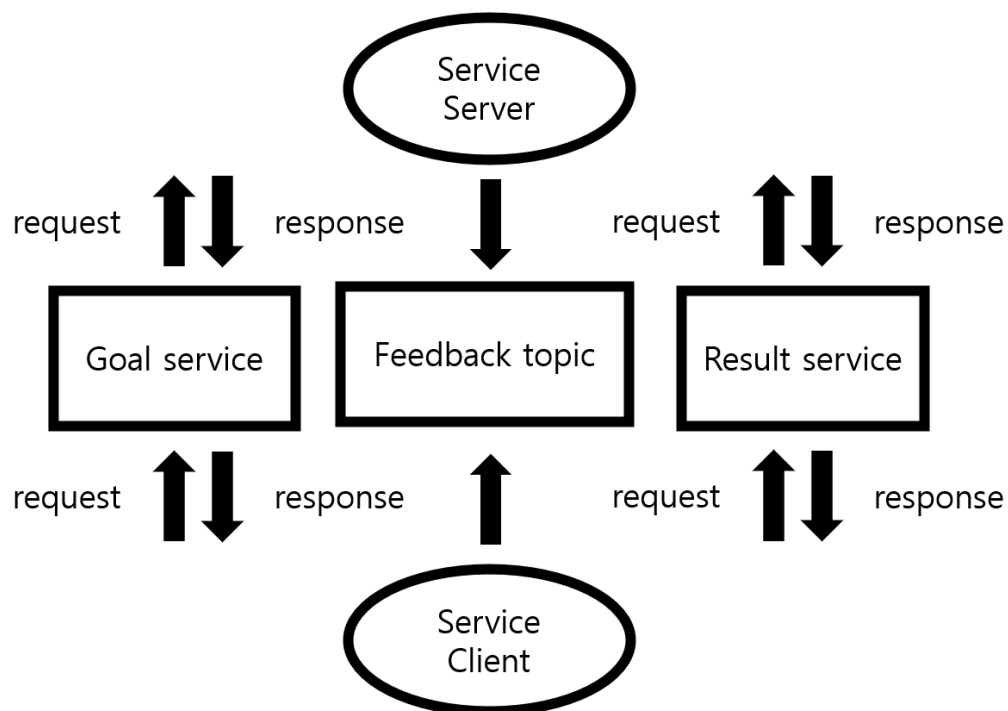
Action 통신은 goal, feedback, result로 구성된다.

Action server와 action client로 구성된다.

장기 작업에 적합하다.

Topic과 service에 기반하여 동작하며 작업을 취소할 수 있으며 지속적인 feedback을 제공한다.

Action client는 action server로 goal을 전송하고 action server는 feedback과 result를 반환한다.



Command

- `$ ros2 action list`

현재 활성화된 action의 목록을 출력한다.

- `$ ros2 action list -t`

활성화된 action의 목록의 action type을 표시한다.

- `$ ros2 action info`

해당 action의 Action, Action clients, Action servers를 출력한다.

- `$ ros2 action send_goal <action_name> <action_type> <values>`

yaml형식의 value를 goal로 전송한다.