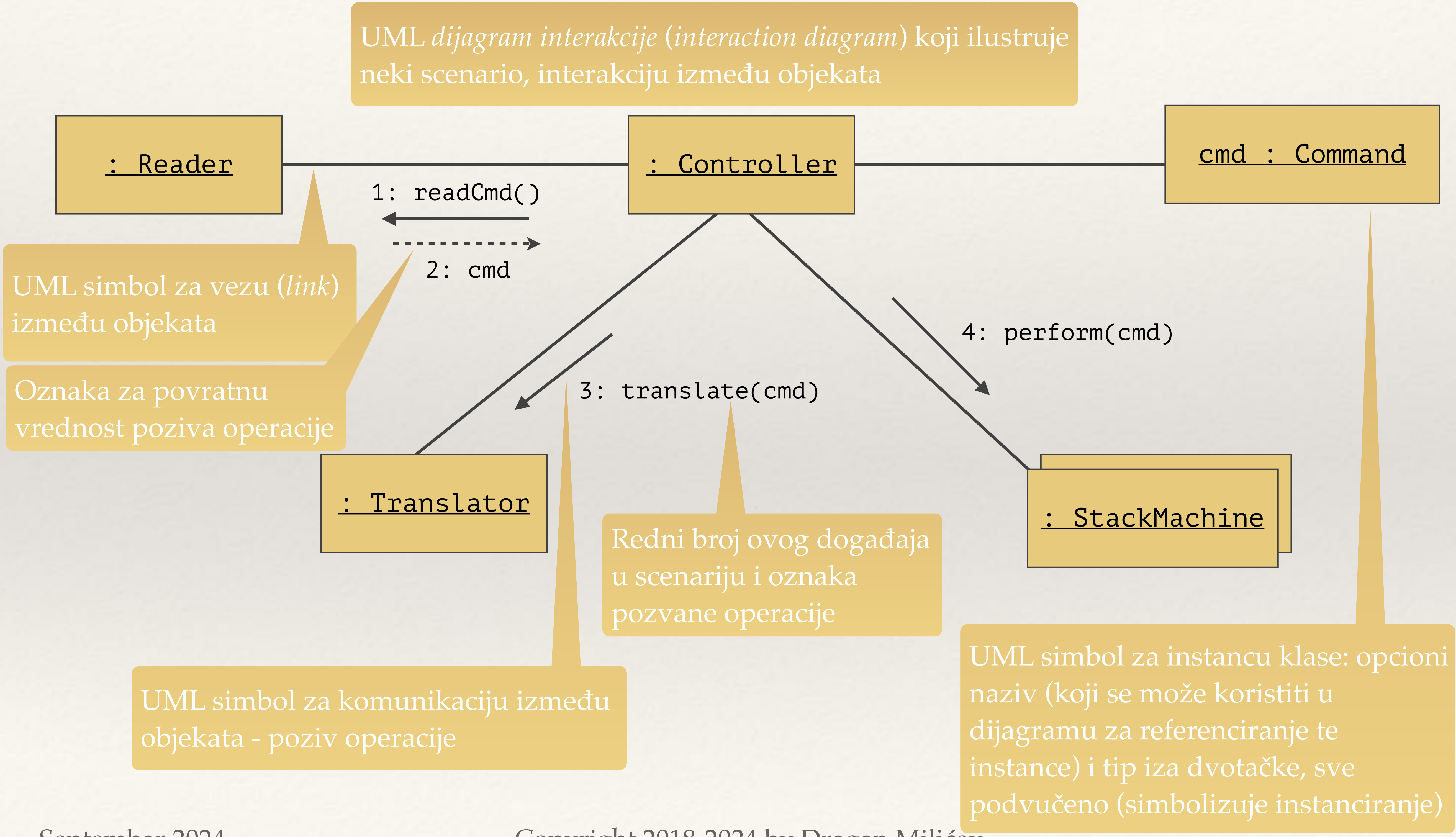
Algoritamska dekompozicija



UML *dijagram interakcije* (*interaction diagram*) koji ilustruje neki scenario, interakciju između objekata

UML simbol za vezu (*link*) između objekata

Oznaka za povratnu vrednost poziva operacije

Redni broj ovog događaja u scenariju i oznaka pozvane operacije

UML simbol za komunikaciju između objekata - poziv operacije

UML simbol za instancu klase: opcioni naziv (koji se može koristiti u dijagramu za referenciranje te instance) i tip iza dvotačke, sve podvučeno (simbolizuje instanciranje)

Algoritamska dekompozicija

```
enum OpCode { add, sub, … };

class Command {
public:
 Command (int out, string cmd);

 OpCode getOpCode ();
 int    getOut ();
 …

private:
  string in;
  OpCode code;
  int out;
};


class Controller {
public:
  void main ();
  …

private:
  Reader* myReader;
  Translator* myReader;
  StackMachine* mySMs[…];
};

void Controller::main () {
  Command* cmd = myReader->read();
  while (cmd!=nullptr) {
    myTranslator->translate(cmd);
    int out = cmd->getOut();
    mySMs[out]->perform(cmd);
  }
}
```

```
class Reader {
public:
  Command* read ();
  …
};

class Translator {
public:
  void translate (Command*);
  …
};

class StackMachine {
public:
  void perform (Command*);
  …

protected:
  void push (int);
  int  pop ();
  …

private:
  int stack[MaxStackSize];
  unsigned sp;
  …
};

void StackMachine::perform (Command* cmd) {
  switch (cmd->getOpCode()) {

    case add: {
      int op1 = this->pop();
      int op2 = this->pop();
      this->push(op1+op2);
      break;
    }

    case sub: …
    …
  }
}
```