

др Гордана Павловић-Лажетић

QUO VADIMUS — РАЧУНАРСТВО НА КРАЈУ 20. ВЕКА

Резиме. Рачунарство је научна дисциплина која систематски проучава алгоритамске процесе који описују и трансформишу информацију: њихову теорију, анализу, пројектовање, ефикасност, имплементацију и примене. Корени рачунарства препознају се у математици, природним наукама и техници, а основне парадигме које карактеришу рачунарство и одређују га према овим дисциплинама су теорија (апстракција), експеримент и пројекат.

Мада је експанзивни развој рачунарства обележио другу половину овога века, у структури саме дисциплине, основним поставкама и методологији истраживања и рада уочавају се инваријанте које дају стабилност том развоју. У излагању ће бити приказане савремене тенденције и правци даљег развоја области рачунарства. Само пуно препознавање суштинских тих праваца и значаја фундаменталних поставки дисциплине, омогућују производњу, кроз наставни процес, професионалаца за време и технологију која долази.

1. Увод

Наслов овог рада — Куда идемо — Рачунарство на крају 20. века, делимично је позајмљен из чланка [12] објављеног у водећем часопису из области рачунарства — Communications of the ACM, који се односио на стање дисциплине рачунарства осамдесетих година овога века. Питања која су тада разматрана добијају на актуелности на измаку овога века, као и поводом педесет година развоја ове изузетно динамичне дисциплине, када се утврђује стање, сумирају резултати и препознају и фундирају правци њеног даљег развоја. Нове, али не и потпуно нове одговоре на та питања покушаћемо да пронађемо у најновијим извештајима, саопштењима, радовима, у актуелној пракси и увек живој дискусији коју ова питања отварају, као и у појединачним искуствима свакога од нас.

2. Природа рачунарства

Мада изузетно динамична научна дисциплина, за коју се верује да знања од данас већ постају застарела сутра, дубља анализа фундаменталних поставки и основних принципа који воде развој дисциплине, као и методологија истраживања и рада садрже инваријанте које јој дају стабилност. Зато и одговори на питања о томе шта је рачунарство, шта је објекат, предмет и које су методе изучавања, које су његове најважније гране, на којим се правцима могу очекивати најснажнији пробоји и које врсте, који су домети, домени примене и правци развоја, који се захтеви постављају пред наставне планове и програме у овој области на целој

Предавање одржано на Републичком семинару „Настава математике и рачунарства“, Београд, јануара 1997.

образовној вертикали, представљају само делом новост у односу на одговоре на слична питања постављена пре деценију или две.

На нашим просторима термин „рачунарство“ користи се често као део синтагме „рачунарство и информатика“. Такав је случај и са називом предмета који се предају у средњим школама. Мада се овим терминима могу придружити у извесној мери различита значења (нпр. информатика проучава информационе системе а рачунарство изучава алгоритамске процесе), основна разлика у терминима је та што долазе са различитих говорних подручја. Док се на енглеском говорном подручју усталио термин везан за рачунање (computing, computer science, computer engineering), на подручју западне Европе исту улогу имају термини који у корену имају реч информатика — *informatique*, *informatik*). Зато ћемо у даљем излагању користити само термин „рачунарство“ који се односи на рачунарску науку и рачунарску технику.

При покушају да се нађе општеприхваћена дефиниција рачунарства долази се до открића да ни језгро такве дефиниције не постоји. Наставља се, на разним форумима, стара дебата о томе да ли је рачунарска наука — наука, или техничка дисциплина, или просто технологија? Шта је интелектуална суштина дисциплине? Да ли је то трајна дисциплина или ће јој важност и живот престати у наредној генерацији? Какав је однос рачунарства и програмирања?

Ма колико различите биле, све дефиниције рачунарства се мање-више слажу у констатацији да рачунарство дели заједничке научне методе са математиком, природним наукама и техником. Парадигма која га везује за математику је теорија, са природним наукама везује га експериментални научни метод, док је пројектовање заједничко за рачунарство и технику.

Када је у питању рачунарство, ова три аспекта су међусобно испреплетена и ниједан нема апсолутну предност. За разлику од природних наука, нпр. физике, које изучавају природни свет какав постоји, и објашњавају постојеће феномене, рачунарство се бави оним што може да постоји и како објаснити и анализирати могуће у обради информација. Мада заједничке, фундаменталне парадигме математике и физике, на пример, толико су различите од оних у рачунарству, да се истраживања не могу мерити истим аршином. На пример, „лаки“ доказ теореме у математици није занимљив, док је верификација сваког програма веома значајна у поузданости софтвера. Улога експеримента у физици и рачунарству је сасвим различита. У физици се преко експеримента може доћи до значајне теорије и резултата. У рачунарству теорија, интелектуални оквир, диктира шта се може опазити и шта је релевантно да се опазе па дакле и чиме да се експериментира.

Кратко речено, рачунарство је систематично изучавање алгоритамских процеса који описују и трансформишу информацију: њихове теорије, анализе, пројектовања, ефикасности, имплементације и примене. Основно питање које се поставља пред цело рачунарство је: Шта се може ефикасно аутоматизовати [2].

Мало историје: Рачунарство је рођено раних четрдесетих година, као спој теорије алгоритама, математичке логике и проналаска електронског рачунара са памћењем програма. Корени рачунарства су дубоко у математици — у њеној аналитичности, и техници — у пројектовању. Специфичност дисциплине је

у томе што укључује сопствену теорију, експерименталну методу и технику — инжењерство, за разлику од већине природних наука које су раздвојене од техничких дисциплина које примењују њихове резултате (нпр. хемија и хемијско инжењерство).

Један производ дугогодишње интеракције математике и технике биле су механичке направе за помоћ у рачунању. Постоје подаци о навигационим инструментима старим хиљаду година. Паскал и Лајбниц су изградили аритметичке калкулаторе средином 17. века, а Чарлс Бебиџ је тридесетих година прошлог века конструисао аналитичку машину која је могла механички и без грешке да рачуна логаритме, тригонометријске и друге аритметичке функције. Двадесетих година овога века Буш је конструисао аналогни рачунар за решавање општег система диференцијалних једначина. Истовремено су се појавиле електромеханичке машине за израчунавање збира, разлике, производа, количника, квадратног корена. Нагласимо да су све ово справе за помоћ у рачунању или за само рачунање, а не рачунари (са памћењем програма).

С друге стране, ту је математичка логика. То је грана математике која се бави критеријумима ваљаности извођења и формалних принципа расуђивања. У 19. веку почело је тражење универзалног система логике који не би садржао некомплетности уочене у познатим дедуктивним системима. У комплетном систему било би могуће механички одговорити на питање да ли је произвољни исказ тачан или не. Године 1931. Гедел је објавио резултат да такав комплетни систем не постоји. Крајем 30-тих година Алан Тјуринг је изнео идеју универзалног рачунара који би симулирао произвољну корак-по-корак процедуру било које машине за рачунање. Његови резултати слични су Геделовим: постоје добро дефинисани проблеми који се не могу решити ниједном механичком процедуром. Значај логике је не само у томе што је омогућила дубоки увид у границе аутоматског рачунања, већ и због тога што је открила да ниске симбола, можда кодирани бројевима, могу бити интерпретиране и као подаци и као програми. Ово је кључно запажање које одваја рачунаре са памћењем програма од машина за рачунање. Кораци алгорита кодирају се машинском репрезентацијом и смештају у меморију за касније декодирање и извршавање, при чему се машински код изводи аутоматски из симболичког облика вишег нивоа, програмског језика.

Оно што је створило дисциплину рачунарства јесте експлицитно преплитање израчунавања и логичке манипулације симболима, са модерним достигнућима електронике и електронске репрезентације информација.

Научни прогрес рачунарске науке је значајно компликован растом рачунарске индустрије. Експлозија технологије принудила је пројектанте машина на *ad hoc* решења да би искористили брзо смањење односа цена/квалитет, често без прилике да уђу дубоко у истраживање најадекватније архитектуре. Данас је развој рачунарске науке засењен, и више него пре деценију или две (мада се и онда то запажало), технолошким развојем у индустрији, узастопним револуцијама компоненти рачунара. Слично, људи у рачунарској науци су тако близу апликацијама и непрестаним захтевима за решавањем текућих техничких проблема, да занемарују базични научни развој. Рачунарство се разликује и од других

техника по томе што се технолошка база мења сувише брзо. Све што се учи ad hoc застарева врло брзо. Готово да би се могла исказати јеретичка мисао да би се рачунарство солидније и стабилније развијало без рачунара.

2.1. Подобласти рачунарства

Класификациона шема Рачунарског прегледа (Computing Reviews) [11], који приказује све релевантне публикације и резултате у целокупној области рачунарства у свету, препознаје 10 подобласти од којих свака има хијерархијску структуру. Део те хијерархијске структуре области рачунарства представљају следеће подобласти:

1. ХАРДВЕР
2. ОРГАНИЗАЦИЈА РАЧУНАРСКИХ СИСТЕМА

...

- 2.2. Рачунарско-комуникационе мреже
- 2.3. Системи посебне намене и системи засновани на апликацији

3. СОФТВЕР

- 3.1. Програмске технике
- 3.2. Софтверско инжењерство

...

- 3.2.4. Верификација програма

...

- 3.3. Програмски језици
- 3.4. Оперативни системи

...

4. ПОДАЦИ

- 4.1. Структуре података

...

5. ТЕОРИЈА РАЧУНАРСТВА

...

- 5.2. Анализа алгоритама и сложеност проблема

...

- 5.4. Математичка логика и формални језици

...

6. МАТЕМАТИКА ИЗРАЧУНАВАЊА

- 6.1. Математичка анализа
- 6.2. Дискретна математика
- 6.3. Вероватноћа и статистика

...

-
- 7. ИНФОРМАЦИОНИ СИСТЕМИ
 - 7.1. Модели и принципи
 - 7.2. Управљање базама података
 - 7.3. Чување и претраживање информација
 - 7.4. Апликације информационих система
 - ...
 - 8. МЕТОДОЛОГИЈЕ ИЗРАЧУНАВАЊА
 - ...
 - 8.2. Вештачка интелигенција (обрада природног језика)
 - 8.3. Рачунарска графика
 - 8.4. Обрада слика
 - 8.5. Препознавање облика
 - 8.6. Симулација и моделирање
 - 8.7. Обрада текста
 - ...
 - 9. ПРИМЕНЕ РАЧУНАРСТВА
 - 9.1. Обрада административних података
 - ...
 - 9.5. Уметност и друштвене науке
 - 9.6. САД (пројектовање уз помоћ рачунара)
 - ...
 - 10. РАЧУНАРСКЕ СРЕДИНЕ
 - 10.1. Рачунарска индустрија (и рачунарски стандарди)
 - 10.2. Рачунари и образовање
 - ...
 - 10.5. Правни аспекти рачунарства
 - ...
 - 10.7. Рачунарска професија (и кодекси добре праксе и етика)
 - ...

На нашим просторима активна су истраживања у готово свим поменутиим областима.

3. Правци развоја

Предвиђати у тако динамичној дисциплини каква је рачунарство веома је незахвалан посао. Ипак, њиме се из крајње прагматичних разлога баве високи форуми и институције. Тако, занимљиво је да је 80-тих година угледни форум најодговорнијих људи САД (министар одбране, Председников саветник за науку и други), заједно са тимом најкомпетентнијих експерата, процењивао технологије последње деценије овога века, из свих области људске делатности. Као три

најперспективније означио је микрорачунаре, дистрибуирано израчунавање и вештачку интелигенцију [3].

Данас, при крају 90-тих, може се констатовати повећани интерес за низ области рачунарства; за неке области интерес је смањен или пренет на сродне области. Наведимо укратко неке од тих области.

Однос рачунарских комуникација и рачунарства добија на значају највише због улоге мреже над мрежама — Интернета. Он мотивише развој фундаменталних области као што је валидација протокола. Моделирање топологије мреже које је представљало задатак будућности седамдесетих, нема више такав значај. Аутоматизација пројектовања представља и даље изазов, како за теоретичаре и комбинаторичаре, тако и за оне који се баве графичким улазом, језицима врло високог нивоа и симулацијом система. Препознавање говора је још увек, као и седамдесетих, практично на почетку. У препознавању рукописа постоје значајни резултати. Препознавање облика и вештачка интелигенција, које су представљале правац развоја седамдесетих и осамдесетих, замењене су (и допуњене) замућеним (fuzzy) системима, интелигентним системима, неурорачунарима. Уместо на канцеларијске системе и персоналне рачунаре, нагласак се данас може ставити на Интернет машине и разноврсност сумеђа (interfaces). Очекује се појава оперативних система прилагођених новим могућностима оперативне меморије од више стотина мегабајта. Уз ову изузетну густину (extra density) на силицијуму мора да иде и развој софтверског знања, да чипови не би завршавали као компоненте фрижидера и штедњака. Размишља се о другим архитектурама, са богатијим скуповима базичних инструкција, као што су „сорт“, налажење медијане, итд. Пред пројектанте експерименталних језичких система постављају се изазови изградње програмских језика високог нивоа, непроцедуралних, довољно општих — не оријентисаних ка апликацији, којима би се могли изразити захтеви типа „анализирај податке“, „решит једначину“, итд. Језици као што су ALGOL, COBOL, PL/I, LISP, не могу то да ураде.

Задржимо се нешто детаљније на неколиким рачунарским темама, у овом тренутку веома актуелним, а којима свакако припада и будућност. То су: формалне методе, објектно-оријентисана методологија, нова генерација система за управљање базама података (SUBP), електронски речник, препознавање говора.

Прво, формалне методе. О формалним методама најубедљивије говори један од твораца рачунарства као научне дисциплине, холандски научник, по образовању математичар и физичар, Е. Дејкстра (Edsger Dijkstra) (аутор раних алгоритама у теорији графова, први имплементатор језика ALGOL/60, првог оперативног система компонованог од експлицитно синхронизованих секвенцијалних процеса, аутор предикацких трансформатора за дефинисање семантике, и програмерске методологије у најширем смислу; ради на формалном извођењу програма из спецификације и на свеопштој математизацији рачунарства). Формалне методе су нови облик у коме идеја верификације шездесетих и седамдесетих добија на актуелности данас. То су методе засноване на предикатском рачуну које омогућују извођење формалног доказа да развијени програм задовољава такође формално задату функционалну спецификацију (да ради оно што се очекује).

Програм и доказ се развијају паралелно. Исказ да програм задовољава формалну спецификацију односи се на сва могућа израчунавања која се обављају под контролом тог програма (сва његова могућа извршења). То даље значи да је потребно радити са програмским текстом занемарујући његову интерпретацију као извршног кода, тј. без његовог тестирања. Тиме се у значајној мери штеди мисаона активност потребна при разматрању понашања програма тј. при његовом тестирању. Оно што се добије оваквим поступком је еквивалентно раду са елементима скупа на начин који игнорише саме елементе а оперише само дефиницијом скупа. То је позната метода у математици. Колико је она супериорна, Дејкстра илуструје и следећим примером [7].

Размотримо равну фигуру облика квадрата 8 пута 8, из које су изузета два квадрата 1 пута 1 на супротним угловима. Површина фигуре је 62. Тврђење је да се ова фигура не може покрити 31 домином димензија 1 пута 2. Могући начин за доказивање овог тврђења је да се генеришу сви могући размештаји 31 домине и да се утврди да ниједан не генерише дату фигуру. Једноставни доказ пак је да се квадрати димензија 1 пута 1 обоје наизменично као на шаховској табли. Свака домина тада треба да заузме два суседна квадратића — један бео и један црни, тј. свако покривање фигуре доминама 1 пута 2 покрива једнак број белих и црних квадратића. Наша фигура пак има два квадратића мање у једној него у другој боји, и ниједан распоред 31 домине не производи ту фигуру. Овакав доказ не само да је елегантнији и штеди мисаоне напоре, већ је и општији и важи за произвољни квадрат парне дужине странице, што се на први начин не би ни могло доказати.

Доста духовито, Дејкстра предлаже да се рачунарској науци додели акроним, с обзиром да се ниједан рачунарски производ не сматра вредним поштовања без акронима. Како је логика у рачунарству, према Дејкстри, од пресудног значаја, он предлаже, као одговарајући, акроним VLSAL (Very Large Scale Application of Logic — примена логике врло високог степена), по угледу на акроним VLSI (Very Large Scale Integration) [7].

Друго, објектна методологија. Објектна (или објектно-оријентисана) методологија се развија као одговор на софтверску кризу, а са циљем да се поновним коришћењем написаних програма (reusability) повећа продуктивност програмера. Објектно оријентисана (ОО) технологија, насупротив управљачки оријентисаној (УО) технологији, јавља се некако истовремено на подручјима програмских језика, вештачке интелигенције (система за репрезентацију знања) и база података. Основне карактеристике ОО технологије су [9]:

- апстракција података
- наслеђивање својстава
- учауреност података и операција
- трајност података.

На поменутих подручјима не постоји стандард ОО модела података. Сви постојећи ОО модели, ипак, деле заједничко ОО-језгро, које се састоји од:

- објеката (произвољне сложености)

- инстанцих променљивих (својстава објеката)
- порука и метода (којима објекти одговарају на поруке)
- класа (као скупова објеката са истим својствима)
- хијерархије класа (које су у релацији подкласа/надкласа)
- наслеђивања својстава и метода (међу класама које су у релацији подкласа/надкласа).

Треће, системи база података нове генерације. Од 60-тих година, технологија управљања подацима еволуирала је од датотечких система до хијерархијских и мрежних система (тзв. I генерација система база података) и релационих система (тзв. II генерација система база података). Осамдесетих година природа апликација база података нагло се променила од једноставнијих ка комплекснијим апликацијама, па почиње период технологије нове генерације. Циљ ове технологије је:

– подршка ОО моделу података – архитектура која омогућује једноставно укључење функционалних проширења – подршка новим типовима података, нпр. временским (историја верзија), просторним, текстуалним, мултимедијалним, дугим трансакцијама, непрецизним подацима и правилима.

Деведесете године постављају нове задатке:

– прелаз од технологије релационих база података на пострелациону технологију – подршка истовременом постојању и међуповезаности старих и нових технологија база података.

Трећу генерацију модела и система база података чине пострелациони, објектни и дистрибуирани хетерогени системи. Развој ових система има специфичну мотивацију, а сваки од проблема система друге генерације добија нову димензију и захтева нова решења [10]. Нове могућности медија у домену капацитета, брзине и квалитета чувања података, као и уређаја за презентацију, постављају нове изазове пред истраживаче и индустрију. Основне идеје о управљању подацима у системима нове генерације, као и сугестије везане за развој мултимедијалних база података, изложене су у документима познатим као Манифести система база података нове генерације [1].

Посебни правац у реализацији система база података нове генерације представљају мултимедијални и дистрибуирани системи, и то хетерогени дистрибуирани системи. Мултимедијални системи база података су они који обезбеђују различите презентације података коришћењем различитих уређаја — магнетних, оптичких, електронских, CD ROM, видеодискова итд. Дистрибуираност се односи и на податке и на систем за управљање базама података. При томе систем за управљање базама података може бити истог типа на свакој од локација на којима је дистрибуирана база података, али се та претпоставка (тзв. претпоставка о хомогености ДСУБП) показује као прејакo ограничење у данашњим условима, када значајне количине података и апликација постоје на разним рачунарима, под различитим оперативним системима и под контролом различитих СУБП. Потреба за истовременим приступом оваквим подацима унутар једне апликације,

или чак и једне трансакције, поставља захтев пред произвођаче ДСУБП да обезбеде подршку хетерогеним ДСУБП. То значи да, поред независности приступа подацима и обраде података од локације, фрагментације, понављања података, машине, оперативног система и мрежног протокола, хетерогени дистрибуирани систем мора да обезбеди и независност од СУБП на појединим локацијама.

Постоје два битно различита приступа решавању овог проблема. Један је изградња тзв. система мултибаза података, СМБП (енгл. multidatabase system) [10]. То је програмски систем који се састоји од низа компоненти. Једна од тих компоненти је јединствени језик за креирање података и манипулисање подацима који су под контролом хетерогених СУБП. СМБП је, пре свега, јединствена сумеђа, кроз коју корисници и апликације могу да комуницирају са разнородним системима. Друга компонента СМБП је глобални управљач трансакција. СМБП обезбеђује, поред локалних трансакција (над једним СУБП), и управљање глобалним трансакцијама. Основни циљ ових система је да кроз управљање глобалним трансакцијама одржавају конзистентност мултибазе.

Други приступ хетерогеним ДСУБП је мање амбициозан, али и комерцијално заступљенији. Његова суштина је у изградњи апликативних програма, тзв. пролаза (енгл. gateway) једног СУБП према другом СУБП, који омогућују кориснику првог СУБП да комуницира са другим СУБП, истом сумеђом којом комуницира са првим системом (други систем се кориснику лажно представља као да је први).

С обзиром на значај који има реализација квалитетних програма пролаза, постоји развијена активност стандардизације одговарајућих протокола. У имплементацији пуне функционалности пролаза јављају се значајни проблеми, па зато комерцијални производи овог типа не подржавају све потребне функције. Оне су још увек у истраживачкој сфери.

Четврто, електронски речник. Један од методолошких пробоја у области рачунарске лингвистике припада концепту електронског речника (е-речник), који поставља темељ новом прилазу аутоматској обради текста [13]. Е-речник, за разлику од машински читљивог речника, садржи широку и исцрпну класификацију и категоризацију речи, довољну за генерисање свих облика, што је посебно значајно за флективни језик као што је наш. Проблем морфолошке анализе, присутан у рачунарској лингвистици као саставни део проблема обраде језичког улаза, захваљујући електронском речнику замењује се општијим и ефикаснијим поступком лексичког препознавања. Ефикасност и богатство примена ове методе може се илустровати на примерима са подручја обраде текста, рачунарске лексикографије и рачунарске лингвистике.

Пето, препознавање говора. Главни проблем у аутоматском препознавању говора представља замућеност података и експлозивни раст простора претраживања — базе образаца говорних сигнала. Решења овог проблема траже се у два смера. Један је технички оријентисан и садржан је у дисциплини неурорачунарства тј. неуроизрачунавања. Он укључује напоре усмерене ка савлађивању величине простора претраживања — високи паралелизам рачунара и програми-

рања, уз могућност обучавања, који треба да симулира масовност и функцију неуронских ћелија мозга, уз специфична хардверска решења — оријентисана према препознавању говора. Овај приступ поставља истраживачке задатке пред неурорачунарство као дисциплину.

Други приступ је оријентисан не ка савладавању простора претраживања већ ка његовом значајном редуковању. Он полази од евидентне чињенице да у препознавању говора, осим препознавања гласова, учествује и један виши степен препознавања (разумевања) поруке која се говором шаље. Овај приступ је интердисциплинарног карактера. Он се састоји у прикупљању што ширег и што дубљег знања из дисциплина које се баве процесима сазнавања и дисциплина чији апарат омогућује формализацију и прецизно описивање појава које проблематика препознавања говора проучава; то су дисциплине које помажу у описивању и разоткривању тајни природног језика — лингвистика (општа, теоријска, психо, когнитивна, невро), математика и посебно математичка логика. Управо знања која се апстрахују из резултата истраживања у овим областима треба да послуже да се простор претраживања значајно редукује. Посебан значај у овом другом, мултидисциплинарном приступу проблему препознавања говора има информатика и посебно информатичка обрада природног језика — рачунарска лингвистика.

4. Образовање

Рачунарство се разликује од свих познатих интелектуалних дисциплина. То је наука која концептуализује и креира интелектуална средства и теорије које помажу разумевању, анализи, и изградњи могућег. Парадигме које професионалци у рачунарству носе из својих матичних дисциплина (математика, физика, техника), иако различите од оних у рачунарству, помажу развоју рачунарства истраживачким методологијама тих дисциплина. Насупрот њима, професионалци који се школују на одсецима за рачунарство не долазе из добро заснованих наука, и према неким ауторима, неизвесно је да ли имају праву методологију, јер парадигме рачунарске науке нису још сасвим схваћене [3]. То би се могло схватити као један, објективни разлог због којег се рачунарство не испушта из наручја математике (или електротехнике) у нашој универзитетској средини.

Значајна компонента рачунарства, присутна у готово свим гранама рачунарства, јесте програмирање. Ипак, изједначавање рачунарске науке са програмирањем, које је веома присутно у образовним програмима, занемарује друге битне аспекте рачунарства, који нису програмирање, као што су пројектовање хардвера, архитектура система, пројектовање нивоа оперативних система, структурирање базе података за специфичне апликације, валидација модела, итд.

У учењу уопште, па и у учењу програмирања, два су приступа веома изражена — грубо речено: *занатски и научни* [6]. Први се заснива на усвајању вештине од „мајстора“, посматрањем и имитирањем, без експлицитне формулације знања, и уз обавезно еснафско скривање стеченог знања унутар еснафа. Други подразумева покушаје од стране учитеља да формулише знање и да објасни вештину што је могуће експлицитније, излажући тако и знање и вештину као опште добро јавном коришћењу. Ова два приступа присутна су и у другим областима.

Тако у учењу физике, на пример, преовлађује научни приступ, док у савладавању медицине провлађује занатски. Математика је негде између. Резултати се јавно публикују и уче, али се мало учи о томе како се ради математика, како се долази до резултата. Публиковање хеуристика које доводе до резултата сматра се најчешће недовољно научним. Експлицирање имплицитних знања и описивање вештине и искуства тако да се они могу пренети, представља зачеће нове науке. Да цитирамо Лајбница, „ништа није важније него видети изворе проналаска, који су, по мом мишљењу, занимљивији од самог проналаска“. Ниједна крајност у овим приступима у настави програмирања није добра. Ни произвољна „мешавина“ ових приступа такође не мора бити добра. Једна таква, најчешће заступљена поразна „мешавина“ у настави програмирања је научно експлицирање знања о „занатским“ средствима — алатима, језицима, системима. Прави приступ би требало да садржи [6] 10% чињеница о језицима, системима, алатима, и 90% експлицитног или имплицитног (имитирањем) преношења знања о вештини мишљења и решавања проблема. Тек такав приступ настави програмирања обезбеђује изградњу инвентивног програмера, способног да савлађује комплексност реалних проблема.

У вези са програмирањем, изузетно је важан избор програмског језика на коме се излаже и учи вештина решавања проблема и програмирања, с обзиром да средства која се у том процесу користе имају велики утицај на начин на који размишљамо, и на нашу способност размишљања. Занимљива је (мада дискутабилна) Дејкстрина оштра критика неких програмских језика [4], као што су PL/I, COBOL, APL, FORTRAN, BASIC. За компетентног програмера, према истом аутору, од виталног је значаја, поред математичких наклоности, савршено владање матерњим језиком.

Рачунарство у средњој школи представља подручје великог интересовања у образовним системима многих земаља, па, срећом, и наше. Мада је читав образовни систем сасвим различит од нашег, поменућу овде наставни план за средње школе који је сачинио АСМ образовни одбор за основне и средње школе у САД. Он се састоји од четири предмета који представљају део општег образовања ученика средњих школа:

- Увод у рачунарску науку I
- Увод у рачунарску науку II
- Увод у програмске језике високог нивоа
- Примене и последице рачунара

Прва два предмета су намењена ученицима са изражним афинитетима према рачунарству. Они представљају два нивоа дубине истог материјала и чине довољну основу за праћење рачунарских курсева са друге године универзитета.

Друга два предмета су општег карактера и намењена су свим ученицима средњих школа. Трећи предмет је предмет о програмирању тј. писању програма на вишем програмском језику, по избору школе. За овај предмет неопходна је значајна хардверска опрема. Четврти предмет нуди информације о начину на који се рачунари употребљавају и о последицама и утицајима које имају на живот

људи. За овакав предмет потребна је и значајна софтверска опрема, којом је неопходно да ученици овладавају сами. Уџбеници садрже принципе на којима се заснива апликативни софтвер којим експериментишу, као и мотивацију за његову примену, док је коришћење самог софтвера подржано приручницима. Избор софтвера је на школи тј. округу. У случају да округ нема могућности да адекватно — кадровски и опремом подржи целу линију рачунарских предмета, неопходно је да направи приоритете. При том је недопустиво да се држи више предмета него што то допуштају расположиви ресурси.

Анализа наставних планова и програма за рачунарство у нашој средњој школи није предмет овог разматрања. Ипак, истакла бих да они не посвећују довољно пажње конзистентним критеријумима професије и педагогије, а питања кодекса добре праксе и етике се и не постављају. Значајни делови програма за поједине предмете више би одговарали тематским курсевима које организују софтверске фирме за обуку својих корисника. Излажу се специфични софтверски производи уместо различитих принципа и приступа њиховој спецификацији и изградњи. Софтвер чије је изучавање садржај оваквих предмета и претпоставка њихове реализације, прописан је на нивоу Републике, а његова набавка (и цена његове набавке) као да није ни у чијој компетенцији. Неки од уџбеника су значајно кориговали недостатке програма. Неки су пак по садржају ефемерни и више личе на приручнике за одређени софтверски производ, а у делу који се бави концептима, постају неупотребљиво непрецизни, недоследни и некоректни. Зато је пред нашим наставницима и професорима тежак задатак да, понекад упркос наставном програму и уџбенику, науче ученике знањима која су у основи и оних производа које користе, али и многих других, знањима чија актуелност неће проћи пре него што ученици заврше школовање.

5. Закључак

Уместо закључка, пар речи о рачунарској терминологији. Размишљање је најинтимнија активност, а у значајној мери се открива начином на који се користи (или злоупотребљава) језик. Анализе језика пак почињу са речницима. Тако се у једној врсти речника енглеског језика, која описује савремену употребу језика — Webster's New Collegiate Dictionary, у једном од новијих издања појавило ново значење речи „интелигентан“: способан за обављање неких функција рачунара (интелигентан рачунарски терминал) [5]. Антропоморфизми у рачунарству (терминал – „интелигентан“, рачунар – „он“) нанели су велику штету самој области, јер подразумевају идентификацију програмера са програмима и машинама. Идентификовање програма са особом која живи у времену идентификује програм са његовом операционом семантиком, тј. са његовим понашањем при израчунавању, што је, како је већ описано у делу о формалним методама, разбацивање мисаоних напора. Употреба антропоморфизама карактерише недораслу, пре-научну фазу рачунарства. Да цитирамо опет Дејкстру [5], „онај који се бави рачунарском науком не заслужује тај назив пре него што се одважи да „баг“ (енгл. bug) – бубу назове правим именом — грешка, и да тиме кривицу за неисправан програм пребаци онеме коме и припада — програмеру“. Израз

„бубе у програму“ сугерише да су се оне некако „увукле“ у програм; ако их је мало, програм „није лош“. Али, ако се „бубе“ назову грешкама, и једна од њих чини програм, једноставно, погрешним, и то кривицом програмера. Духовит је предлог да се уведу казне за употребу антропоморфизама типа „рекао сам му (програму), одговорио ми је, програм жели нешто, зна нешто, очекује нешто, верује у нешто“, итд.

Превод рачунарске терминологије представља посебан проблем. Без развоја језика о новој технологији нема ни развоја сопствене технологије — преузимањем језика преузима се и технологија. Зато је потребно радити на стручном и адекватном преводу као и развијању сопствене терминологије. Ово не треба схватити механички; устаљени термини као што је хардвер (hardware) или софтвер (software) могу се и преузети, мада то одмах ствара проблеме са терминима истог корена (нпр. middleware). Један конзистентан приступ преводу рачунарске терминологије представља превод Оксфордског речника рачунарства [8].

ЛИТЕРАТУРА

- [1] Darwen, H., Date, C.J., *The third Manifesto*, ACM Sigmod Records, 24(1), 1995, 39–49.
- [2] Denning, P. (chairman) et al., *Computing as a discipline*, Communications of the ACM, 32(1), 1989, 9–23.
- [3] Denning, P. (editor-in-chief), *A Debate on Teaching Computing Science*, Communications of the ACM, 32(12), 1989, 1397–1414.
- [4] Dijkstra, E.W., *How do we tell truths that might hurt?*, *Selected writings on computing: A personal perspective*, 1982.
- [5] Dijkstra, E.W., *On Webster, Users, Bugs, and Aristotle*, (као [4]).
- [6] Dijkstra, E.W., *Craftsmen or scientists*, (као [4]).
- [7] Dijkstra, E.W., *On the cruelty of really teaching computer science*, Communications of the ACM, 32(12), 1989, 1398–1404.
- [8] Борђевић, С. (уредник): *Оксфордски речник рачунарства*, превод са енглеског, НОЛИТ, Београд, 1990.
- [9] Kim, W., *Introduction to Object-Oriented Databases*, MIT Press, 1990.
- [10] Kim, W.(editor), *Modern Database Systems*, Addison Wesley Publishing Comp, 1995.
- [11] Sammet, J.E., Ralston, A. (editors), *The new Computing Reviews classification system — final version*, Communications of the ACM 25(1), 1982, 13–25.
- [12] Traub, J.F., (editor), *Quo Vadimus: Computer Science in a Decade*, Communications of the ACM, 24(6), 1981, 351–369.
- [13] Vitas, D. (i ostali), *Electronic dictionary and text processing in Serbo-Croatian*, Sprache-Kommunikation-Informatik, Max Niemeyer Verlag, Tubingen, 1993, 225–231.