Semina □ Nr 17
Scientiarum 2018
s. 84–106
DOI: http://dx.doi.org/10.15633/ss.3516

Robert Janusz

Paradygmat obiektowy w oprogramowaniu astronomicznym

Astronomia od wieków uchodzi za naukę *par excellence* matematyczną. Starożytnym (i nie tylko) filozofom wydawało się, że obroty sfer niebieskich pozwalają abstrahować matematykę w najczystszej formie. Nowożytna fizyka Newtonowska, poszerzająca doświadczenia Galileusza (1564–1642), wyznaczyła nowy kierunek w filozofii przyrody, mianowicie dzięki zasadom matematycznym stało się możliwe lepsze poznawanie Kosmosu, co prawda tylko w okolicy Słońca. Szybko okazało się, że moce obliczeniowe znane ludzkości nie wystarczają do rozwiązywania nawet niezbyt złożonych układów ciał niebieskich.

Wraz z nowoczesnymi teoriami fizyki także problemy numeryczne astronomii stały się jeszcze bardziej skomplikowane i złożone. Sukcesy obliczeń za pomocą tablic logarytmicznych i innych prostych jeszcze liczydeł sprawiły, że wiara w obliczenia astronomiczne nie tylko nie gasła, ale rosła – opłacało się inwestować w obliczenia, gdyż odkrywały one rzeczywistość niedostępną innym metodom poznawczym, pozwalały zrozumieć, np. to co Galileusz i jego następcy obserwują przez swoje lunety i teleskopy. Ta wiara[1] nie ustaje do dziś,

[1] Chodzi o to, że obliczalność, będąca kluczowym, choć jeszcze prostym przejawem matematyczności świata badanym w astronomii, nie jest problemem astronomicznym, ale istotnie filozoficznym. Bez tego typu wiary, jak również bez wiary m.in. w istnienie obiektów mierzonych, nie jest możliwe sensowne uprawianie jakiejkolwiek nauki realnej. Podobnie w informatyce – wiara w istnienie np. komputerów nie jest zagadnieniem informatycznym, ale filozoficznym.

gdyż podpierają ją bardzo precyzyjne przewidywania obserwacyjne. Dzięki superkomputerom można rozwiązać w bardzo krótkim czasie problemy tylko opisane przez dawniejszych uczonych – problemy, które stały się jakby kamieniami milowymi w odkrywaniu matematycznych praw rządzących Kosmosem. Już dzięki Newtonowi zrozumieliśmy, że równania różniczkowe sterują Kosmosem, a dzięki Einsteinowi, że nieco bardziej złożone równania rządzą ewolucją Kosmosu jako całości.

Astrofizyka rozwinęła się jednak dopiero w XIX wieku. Niektórzy matematycy nie byli przygotowani na to, że fizykę (ziemską) można ekstrapolować w przestrzeń zarezerwowaną dotąd wyłącznie dla matematyki czystej. Zastosowanie fizyki w astronomii uważano za niegodne prawdziwego uczonego, za chwilową rozrywkę, nienaukową modę[2]. Wraz z rozwojem nowej dyscypliny gruntowała się – także dzięki obliczeniom – nie tylko nowa wiedza, ale i filozofia przyrody. Trzeba było inaczej spojrzeć na nasze miejsce w Kosmosie, inaczej na pochodzenie materii ziemskiej, inaczej na rolę, jaką pełnią prawa matematyczne, którymi opisany jest nie tylko lokalny, ale i kosmiczny ład.

Dotychczasowy patos towarzyszący wąskim, czysto matematycznym sukcesom ustąpił prozie analizowania astrofizycznych hipotez, które dziś na co dzień stawiają uczeni próbujący opisać i zrozumieć odległe gwiazdy i galaktyki, o których nie mieli wyobrażenia zamknięci w swojej filozofii uczeni z maleńkiej planety Ziemi. Okazuje się, że ilość danych, jakie wypełniają dzisiejsze bazy wiedzy jest tak ogromna, że stawianie naukowych hipotez wydaje się nie mieć końca. Nowa astronomia (astrofizyka) osądziła i nadal osądza nie zawsze do końca uświadamiane poglądy raczej filozoficzne astronomów i związane z nimi bardzo lokalne i chwilowe programy naukowe.

Aby postawić hipotezę, bardzo często znajomość matematyki jest, paradoksalnie, początkowo mało istotna – ważniejsza jest filozofia. Dostrzeżenie realnych i istotnych związków między obiektami jest bowiem zasadniczo innej natury niż jedynie matematyczno-obliczeniowa

---

[2] Por. I. Chinnici, *Astronomia i wizje świata*, „Rocznik Filozoficzny Ignatianum" 19/2 (2013), s. 96n.

spekulacja. Dopiero gdy filozoficzny kierunek wydaje się matematycznie właściwy, powstaje kolejna trudność powiązania danych z ich abstrakcyjnym modelem, co niejednokrotnie wiąże się z korektą dotychczasowego lub stworzeniem nowego ilościowego modelu.

Wydawać się może, że na tym etapie twórczej działalności uczonych komputery nie są przydatne, gdyż nie komunikują się z uczonymi na wystarczająco wysokim poziomie języka zdolnego do tworzenia i testowania hipotez. Nie zawsze obecnie tak jest, gdyż w wielu dziedzinach, gdzie stosuje się komputery, nie są one już sparaliżowane brakiem programów, lepiej – brakiem programistów zdolnych zaangażować *hardware* do współpracy z uczonymi. Dzięki paradygmatowi obiektowemu udało się bowiem zdefiniować takie języki programowania komputerów, które „pracują językowo" bezpośrednio w dziedzinie naukowego problemu, co sprawia, że cyfrowa (zero-jedynkowa) moc obliczeniowa, która jest niedostępna astronomowi, dzięki tym językom może być wykorzystana do osiągnięcia ilościowych, matematycznych wyników, na podstawie których można odrzucać stawiane hipotezy albo uprawdopodabniać wybrany, testowany model.

Ta własność języków programowania obiektowego sprawia, że można popatrzeć na samo zagadnienie obliczalności nie z Turingowskiego, ale właśnie z obiektowego, systemowego punktu widzenia[3]. Postaramy się uzasadnić tezę, że informatyczna moc procesorów nie

[3] Recenzent słusznie zauważył, że paradygmat obiektowy nie przeciwstawia się Turingowskiemu modelowi obliczeń na niskim poziomie cyfrowej Maszyny Turinga i że obliczenia na wyższym poziomie abstrakcji/implementacji można zorganizować obiektowo. Tego niskiego poziomu oczywiście nikt nie zamierza kwestionować. Problem polega na tym, że binaria wykonywane przez obliczającą Maszynę Turinga są zupełnie pozbawione warstwy znaczeniowej – nie pracują w dziedzinie problemu (nauki). W programowaniu obiektowym zaś dąży się do tego, aby warstwa informatyczna była jak najbliżej pojęć (nauki). W praktyce nieobiektowej wygląda to tak, że np. astronom stawia problem, który programiści kodują w języku nie zrozumiałym dla astronomów (ani dla maszyn); po kompilacji (do kodu maszynowego) takiego *software'u* sami programiści nie rozumieją tego, co otrzymali (nie pojmują tego), zaś komputer umie to „coś" przetworzyć na „coś", co zrozumieją informatycy, którzy następnie wytłumaczą astronomom, co ci powinni zrozumieć z ich obliczeń. Zaś w obliczeniach obiektowych pojęcie z dziedziny problemu (np. „masa") jest tym, które ma być interpretowane przez system obiektowy prawie natychmiast (bez kompilatorów, informatyków itd.).

jest pierwszoplanowa (istotna) w najbardziej nawet konkretnych obliczeniach, które prowadzą do np. wstępnego (w sensie epistemologicznym) uzyskania wiedzy, gdyż to właśnie językowe, domenowe ujęcie problemu pozwala osiągnąć uczonym wiedzę nie dzięki tasowaniu bitów, ale porządkowaniu teoretycznych i obliczalnych pojęć. To język tworzy naukę i to język określa, co w ogóle warto obliczać (na bitach). Dobry język ujmujący program określa bowiem także filozoficzny sens i cel wiedzy i działań (m.in. właśnie czystych obliczeń) w określonej dziedzinie. Język naukowy i jego użycie w poszukiwaniu prawdy są właściwe jedynie uczonym-osobom.

Nie chcemy przez to powiedzieć, że Turingowski model obliczalności „nie pracuje" w astronomii (raczej: w filozofii astronomii). Bliskie temu paradygmatowi algorytmiczne podejście do zagadnień, jak zobaczymy, jest skuteczne[4] jedynie *ad hoc* – np. wtedy, gdy problem nie ewoluuje. Gdy jednakże podejście do zagadnień się zmienia – mamy np. nowe przyrządy pomiarowe – programy należy pisać praktycznie od nowa. Złożoność technologii cyfrowej sprawia, że uczony (astronom) nie jest w stanie tego uczynić sam, gdyż może to wykonać wyspecjalizowany zespół informatyków, ale ci zwykle nie znają się wystarczająco na astronomii, zwłaszcza jeśli są to specjaliści od sprzętowych (pseudo-Turingowskich[5]) obliczeń.

Obliczalność w sensie Turingowskim miałaby może znaczenie wtedy, gdyby nauka (astronomia) w ogóle się nie rozwijała, a to jest niemożliwe, gdyż każdej nauce fizycznej daleko jest do tego poziomu ogólności, jakim cechuje się racjonalna metafizyka, która też się rozwija. Z drugiej strony można poszerzyć ograniczone „stopem"[6]

---

[4] Jest często nawet dużo lepsze niż obiektowe, podobnie jak zmysły zwierząt niejednokrotnie przewyższają wrażliwość człowieka.

[5] Żaden sprzęt nie jest w gruncie rzeczy Maszyną Turinga – jest jedynie automatem, choć bardzo złożonym.

[6] Recenzent sugerował pominąć ten akapit, ma on jednak ważne znaczenie filozoficzne, niewyrażalne w odniesieniu do realnych komputerów, ale także tych teoretycznych, modelowanych Maszyną Turinga, która osiąga sukces obliczeniowy, gdy się zatrzyma. Rozwiązanie obliczeniowe problemu naukowego nie rozwiązuje problemu nauki, która poszerza swoją dziedzinę, jeśli jest, jak astronomia, wystarczająco bogata i nie zadowala się samymi obliczeniowymi rozwiązaniami pojedynczych problemów. Na temat obliczalności i roli „stopu" obliczeń (zatrzymania się) maszyn

Turingowskie obliczanie i zregularyzować je funkcją ζ -Riemanna, co przenosi problem (z filozoficznego zresztą poziomu, jakim cechuje się właściwa, nierealizowalna fizycznie, Turingowska Maszyna) w zakres platonizmu informatycznego, tzn. że wynik obliczeń „znany jest" uprzednio przed samym obliczeniem[7].

Te wstępnie nakreślone, zazębiające się zagadnienia rozwiniemy najpierw na płaszczyźnie bardziej astronomicznej, później przeniesiemy się w obszar bardziej informatyczny. Swoista perspektywa siłą rzeczy jest tu bardzo ograniczona do wybranych przykładów tego zakresu, gdzie informatyka i astronomia mogą się twórczo spotkać, a co wystarcza do wyrażenia i uzasadnienia naszych tez.

1. Astronomiczne obliczenia

Trudności związane z wprowadzeniem informatyki do astronomii można ukazać (z ogromnym przybliżeniem, ale nie bez utraty istoty problemu) już na bardzo podstawowym, starożytnym problemie, jakim jest liczenie czasu.

1.1. Proste problemy o wielkim znaczeniu filozoficznym

Frege uważał, iż „odkrycie, że co ranka wschodzi nie nowe Słońce, lecz zawsze to samo, było chyba jednym z najdonioślejszych

Turingowskich istnieje bogata literatura – tu wystarczy powołać się na R. Penrose'a, np. *Nowy umysł cesarza*.

[7] Maszyna Turinga jest klasycznym przykładem problemu akademickiego: inżynierowie nigdy jej nie zbudują, bo nie mają nieskończenie długiej taśmy, a matematycy są zadowoleni, bo ją „mają" w sensie obiektu Platońskiego. Ponieważ Recenzent postawił ten problem już tu (we Wprowadzeniu) sygnalizuję, że jest on omawiany niżej. Traktując Turingowskie obliczenia niematerialnie, ale matematycznie (tzn. mamy dostępną nieskończoną taśmę, czas jednego kroku obliczeniowego jest „zerowy"), możemy przejść do problemów analitycznych, znanych w rachunku różniczkowym i funkcjach analitycznych. Przedłużenie analityczne funkcji ζ pozwala inaczej patrzeć na obliczalność wyrażeń typu $^{1+1+\cdots}$, obliczanie całek tradycyjnie „nieobliczalnych" itd.

w astronomii"[8]. Mamy tu więc podstawowy problem natury czysto epistemologicznej. Tego nie potrafią żadne super-komputery, my zaś wiemy, że tego typu odkrycia – obserwacje astronomiczne, których opisy są nam znane – są dokonywane przez ludzi od ponad 30 tysięcy lat. Regularność ruchów Słońca i Księżyca, a także problemy z ruchami gwiazd błądzących (planet), przyczyniły się do powstania pierwszych kalendarzy – komputerów czasu, pozwalających na jego rachubę. To one pozwalały obliczać najlepszy czas na zasiew i zbiory roślin, hodowlę zwierząt, przewidywać regularne zmiany pogodowe. Już około 2000 lat a.Ch.n. obliczenia astronomiczne były imponujące. Do tych najbardziej znanych (wyznaczenie okresów Słońca, Księżyca i planet) dołączyło wyznaczenie kąta nachylenia ekliptyki do równika etc.[9].

Z obliczeniami odnoszonymi do realnego świata Egipcjanie, wynalazcy najlepszego kalendarza sprzed 5000 lat, mieli niemały problem, gdyż obserwacja nie była współmierna z cyklicznym systemem obliczeniowym. Praktyczni Egipcjanie przewyższali jednak Greków, gdyż mieli lepszy algorytm „wyrównywania czasów" przez dodanie do dwunastu równych miesięcy pięciu dodatkowych dni ($12 \cdot 30+5=365$). Używanie kalendarza księżycowego (Mezopotamia), który liczył 354 dni, wymagało – jak u Greków – wprowadzania miesiąca przestępnego. Dodatkowo cały problem obliczeń czasu komplikuje zjawisko precesji Ziemi (również znane w starożytności), które sprawia, że algorytmy, już na tym poziomie dostępnych obserwacji, stają się złożone. Zauważmy, że opisane tu problemy kalkulacji czasu (właściwego ujęcia kalendarza, jak również precesji) podlegały ewolucji w tak podstawowej dziedzinie, jaką jest fizyka. Do dziś przejawiają się one m.in. w problemie sekundy przestępnej oraz filozoficzno-naukowych sporach nt. eksperymentu i obserwacji jako takich[10].

---

[8] G. Frege, *Sens i znaczenie*, w: *Pisma semantyczne*, tłum. B. Wolniwicz, Warszawa 1977, s. 60.

[9] Por. A. K. Wróblewski, *Historia fizyki od czasów najdawniejszych do współczesnych*, Warszawa 2007, s. 6n.

[10] Por. Herodot, *Dzieje*, II (Euterpe) 4, tłum. A. Bronikowski, Poznań 1862, s. 98n; A. K. Wróblewski *Historia fizyki od czasów najdawniejszych do współczesnych*, dz. cyt.,

Przytoczmy jeszcze jeden bardzo wymowny przykład, jak obliczalność dotycząca faktów historycznych może być zaskakująca. Nawet dawne, błędnie ustalone społecznie sposoby liczenia czasu, poprzez odniesienie ich do astronomii, mogą dzisiaj być dokładnie wyznaczone przez co nieokreślone historycznie fakty stają się uporządkowane w sposób niedostępny dla żadnej metody humanistycznej. I tak, ojciec filozofii Tales z Miletu (VII–VI w. a.Ch.n.) miał przewidzieć zaćmienie Słońca. Dzięki dzisiejszym obliczeniom można stwierdzić, że chodzi o 28 V 585 r. a.Ch.n.[11].

Bardziej bliskim przykładem naszego dzisiejszego wyobrażenia obliczalności jest współpraca wybitnego astronoma Johannesa Keplera (1571–1630) z Wilhelmem Schickardem (1592–1635), twórcą bodajże pierwszego mechanicznego, cyfrowego komputera. Jego maszyna (projekt ok. 1620 r.) miała pomóc Keplerowi w pokonywaniu rachunkowych trudności. Schickard pisał do swego przyjaciela: „[…] mechanicznie spróbowałem zrobić to, co ty wykonujesz ręcznie, i zbudowałem maszynę, która natychmiast, automatycznie przelicza zadane liczby, dodaje, odejmuje, mnoży, dzieli […]. Skakać będziesz pewnie z radości, gdy zobaczysz, jak przenosi ona liczbę dziesiątek i setek lub też ujmuje ją przy odejmowaniu”[12]. Maszyna ta, jak widać, umiała przetwarzać liczby dziesiętne, a jej oprogramowanie było wkomponowane w mechanikę. Logiką były tu po prostu mechaniczne elementy i, jak się domyślamy, była to logika całkowicie

s. 10n. Zob. Roskal, Z. E. Roskal, *Obserwacyjne versus eksperymentalne testy ogólnej teorii względności*, w: *Albert Einstein i rewolucja relatywistyczna*, red. Z. Pietrzak, Wrocław 2016 (Lectiones & Acroases Philosophicae, IX, 1), s. 59n; Autor mówi tu o tzw. eksperymencie komputerowym; można zauważyć, jak problematyka realnego eksperymentu naukowego i jakościowej obserwacji są dodatkowo komplikowane przez wykorzystanie informatyki w nauce i filozofii; Autor odwołuje się tu do przykładów ściśle astronomicznych. Zob. Platon, *Timaios*, 39d; Rok Platoński miał być *resetem* czasu, czyli momentem, gdy czas astronomiczny zacznie się powtarzać; precesję odkrył Hipparchos z Nikei (ok. 190–120) (być może nie był pierwszym, który był tego świadom), jej okres wynosi ok. 26 tys. lat (0:50.29 rocznie). Zob., D. H. Kelley, E. F. Milone, *Exploring Ancient Skies. A Survey of Ancient and Cultural Astronomy*, New York 2011, s. 246.

[11] Por. A. K. Wróblewski, *Historia fizyki od czasów najdawniejszych do współczesnych*, dz. cyt., s. 18.

[12] Cyt. za: R. Ligonnière, *Prehistoria i historia komputerów*, Wrocław 1992, s. 25.

sztywna, nieprogramowalna. Zwróćmy jednak uwagę na to, że cyfrowe obliczenia potrzebne astronomowi nie są sterowane liczydłem, ale w istocie – matematycznymi prawami Keplera.

Wymienione wyżej przykłady wyraźnie wskazują, jak szeroki jest wachlarz problemów, z którymi ma się zmierzyć informatyka astronomiczna. Potocznie rozumiana, kojarzy się ona z wykorzystaniem jakichś komputerów, automatycznych systemów liczących w nauce. Tymczasem informatyka jest nie do pomyślenia bez oprogramowania. To ono jest „duszą" (*software*) informatyki porządkującą „materię" bitów (*hardware*), organizując je w informatyczne struktury. Może też być bardzo pomocne w tworzeniu nauki jako takiej.

1.2. Oprogramowanie astronomiczne

Nie sposób ująć w krótkim opisie całości oprogramowania astronomicznego, więc to co przedstawiamy poniżej, jest jedynie szkicem, daleką perspektywą, która jednak ujmuje, w przybliżeniu właściwym dla ujęć filozoficznych, złożoną problematykę obliczeń astronomicznych.

Oprogramowanie astronomiczne możemy podzielić na amatorskie w tym sensie, że korzystający z niego uczony nie musi być informatykiem, specjalistą od komputerów. Na tym poziomie oprogramowanie pozwala wizualizować naukowe zagadnienia[13]; pozwala gromadzić i selekcjonować naukowe dane, a także przeprowadzać nawet symulacje nieosiągalne dotąd w inny sposób. Tak zasymulowano i przewidziano coś, co starożytni określiliby (opisanym wyżej) końcem świata, mianowicie ustawienie się planet na linii w dn. 10 III 1982 r. Fatalizm grecki musiał ustąpić nowej filozofii w nauce.

Drugim typem oprogramowania astronomicznego są użytkowe systemy *On Line*, systemy czasu rzeczywistego, które mają na celu bezpośrednie sterowanie np. teleskopem podążającym za wybraną gwiazdą (*Tracking*; zob. wyżej: problem Fregego), oprogramowanie

[13] Mamy tu do czynienia jakby z *quasi Virtual Reality*.

współpracujące z detektorami danych (np. CCD) itd. Można tu zaliczyć także komputerowe urządzenia przeznaczone do lotów kosmicznych, stosowane w sondach, planetaria itp. Przy takim oprogramowaniu nauka stawałaby się jakby systemem wolnym od uczonych, ale tylko pozornie. Urządzenia komputerowe uczestniczące w obserwacji zachowują się bowiem, jako części fizycznego świata, stosownie do tego, z czym mamy do czynienia w m.in. mechanice kwantowej – z nieoznaczonością. Nawet jeśli „coś liczą", to – jako system bez astronoma – nie wychodzą poza liczby, które ktoś po obliczeniach (uczony) musi interpretować. Innymi słowy, komputer – jakkolwiek by nie liczył – jest częścią materialnego świata, zaś uczony dzięki rozumowi poza niego wykracza.

Trzecim rodzajem oprogramowania jest oprogramowanie typowo naukowe, matematyczne, które musi umieć integrować dane historyczne i współczesne, musi wspomagać nie tylko kalkulacje, ale też tworzenie wiedzy. Cechą tego oprogramowania jest m.in. to, że choć znane są dokładnie przybliżone metody rozwiązania, np. równań nieliniowych, analizy Fourierowskiej itp., to jednakże problemem staje się ich bezpośrednie zakodowanie w danym układzie – sprzęcie komputerowym oraz odniesienie ich dokładności i efektywności do realnego problemu naukowego.

Dzięki oprogramowaniu naukowemu możliwy jest postęp w astronomii. Systemy składające się na takie oprogramowanie, można traktować jak wirtualne mikro-obserwatoria astronomiczne i testery teoretyczne, które pozwalają modelować „znane" problemy lub wykorzystywać „znane" prawa natury. Można tu zaliczyć m.in. modelowanie zderzenia galaktyk, ewolucji gwiazd, eksplozji supernowych, powstawania Układu Słonecznego oraz metodyczne testowanie teorii nieobliczalnych analitycznie, do których należy np. rozwiązania ogólne układu ruchu trzech ciał w grawitacyjnym polu Newtonowskim. Oprogramowanie naukowe ma także swego bardzo ważnego reprezentanta w systemach analizy danych. Mamy tu na myśli ich interpretację i reprezentację, np. graficzną, tworzenie i analizę synchroniczną i diachroniczną zjawisk na podstawie informacji z baz danych itd. Zaliczyć tu można także odkrywcze aspekty metodologiczne – np. dzięki analizie danych

można pośrednio odkryć czarną dziurę, której nie można obserwować bezpośrednio[14].

Wspomnijmy w końcu, niestety, także oprogramowanie astrologiczne, tworzone przez niepoważnych informatyków.

1.3. Zalety i wady oprogramowania naukowego

Do oprogramowania typowo naukowego można zaliczyć olbrzymie pakiety naukowe, które mają wspomagać astronoma w jego odkrywczej pracy. Skupimy się tu na systemie IRAF, który należy do jednych z najlepszych w swej dziedzinie, a jednocześnie wskazuje na problemy rozwojowe, typowe dla paradygmatu proceduralnego w programowaniu.

IRAF jest wyjątkowo dobrym i stabilnym systemem, jeśli idzie o jego logiczną budowę. W jego prawie 30-letnim rozwoju zaimplementowano olbrzymią wiedzę astronomiczną, matematyczną oraz informatyczną. System napisany został w paradygmacie proceduralnym, bardzo zbliżonym do FORTRAN-u, wyposażonym w prymitywne skrypty (bez pod-programów), które pełnią rolę jakby interfejsu między informatyczno-obliczeniową, a astronomiczno-teoretyczną warstwą systemu. Interpreter IRAF-u jest uruchamiany co 15 sekund, co potwierdza, że IRAF jest rzeczywiście liderem oprogramowania astronomicznego. W oparciu o niego powstało wiele specjalistycznych pakietów obliczeniowych, m.in. STSDAS, którego nieszczęśliwą historię przytaczamy niżej. IRAF potrafi wyśmienicie analizować obrazy astronomiczne, obliczać fotometrię, analizować widma i ma wiele, wiele innych naukowych zastosowań, jednakże w ogóle nie wspiera paradygmatu obiektowego, ma kłopoty z obliczeniami w klastrach itd.[15]. Słabości tego typu oprogramowania wyłaniały się stopniowo. Opiszemy najważniejsze z nich.

---

[14] Wiele odkryć naukowych dokonało się najpierw „na papierze", tzn. „faktów" domagała się matematyczna teoria, dzięki której sformułowano hipotezę potwierdzoną empirycznie (odkrycia Neptuna, fal grawitacyjnych).

[15] Zob. «iraf.net», «www.stsci.edu».

Wiele systemów, stając naprzeciw nowych analiz naukowych i metod stosowanych w astronomii, uczyniło z systemu IRAF motor swego oprogramowania, ujmując go w swoich nowoczesnych pakietach. Tak np. postąpiły Space Telescope Science Institute (STSI), STScl and Gemini Observatory, AURA. Niestety, zostały one zamknięte 26 IV 2016 r. na rzecz projektu „AstroConda". Problemem stały się konflikty z „resztą" oprogramowania, a także kłopoty z samym rdzeniem IRAF-u. Próbą powiązania zalet IRAF-u z nowoczesnym podejściem obiektowym stał się system PyRAF – swoista hybryda, jeśli idzie o metodologię informatyczną. Właśnie tutaj pojawił się podstawowy brak, z którego nie wszyscy sympatycy Turingowskich obliczeń zdawali sobie sprawę. Przytoczmy ten istotny kłopot dosłownie:

> Wiele procedur IRAF-u, które włączamy do AstroConda, jest tak przestarzałych [*sic!*], że nie mogą być skompilowane do 64-bitowych programów bez znacznych zmian w kodzie źródłowym. Z powodu tego ograniczenia, zawsze dostarczamy IRAF jako 32-bitowy program, nawet dla naszych 64-bitowych [procesorów][16].

[16] „Many of the IRAF tasks that we include with AstroConda are so old that they cannot be compiled as 64-bit executables without significant changes to the source code. Because of this restriction, we always build IRAF as a 32-bit program, even for our 64-bit distributions". Zob. «ssb.stsci.edu/ureka/», «astroconda.readthedocs.io/en/latest/», «astroconda.readthedocs.io/en/latest/faq.html#why-isn-t-iraf- installed-by-default».

[P]odobny był los STSDAS w 64-bitowym IRAF: «http://www.stsci.edu/institute/software_hardware/stsdas/iraf64» – oto jak Space Telescope Science Institute (STSDAS) tłumaczy swoją decyzję: „STSDAS will remain 32-bit. Details: IRAF 2.16 includes support for 64-bit systems. The changes necessary to support 64-bits generally require applications to be modified. For many packages the changes are relatively minor. Nevertheless, for others they can be more substantial, and involve quite a bit of testing and risk of undiscovered problems. In the case of TABLES and STSDAS, there are packages that fall into both these categories. STScI has decided that it is not worth the effort to port these packages in their entirety to 64-bits. The following will describe our plans for handling support of 64-bit IRAF with regard to TABLES and STSDAS:

[N]OAO will produce a 64-bit version of the core tables libraries so that 64-bit IRAF application can access tables. There will continue to be a 32-bit version of tables in the TABLES release to support 32-bit applications. STScI will release 32-bit binaries of the TABLES and STSDAS packages to use with IRAF 2.16. Support for 32-binaries will continue as long as there are sufficient platforms that such binaries

Decyzja o utrzymywaniu przez STSDAS pakietu TABLES jest o tyle kuriozalna, że pakiet ten nie może aspirować do nowoczesnego systemu bazodanowego, którego zresztą w systemie IRAF nigdy skutecznie nie zaimplementowano.

Obiektowe podejście programistyczne wchodziło do astronomii powoli. Było ono szeroko dyskutowane na ostatnich posiedzeniach IAU, począwszy od programów aż po obiektowe bazy danych[17]. Od tego czasu pojawiło się wiele tego typu publikacji i projektów, z których – naszym zdaniem – największe nadzieje budzi projekt nazywany „astropy".

2. Dlaczego język, a nie maszyna?

W informatyce Turingowskiej, której nb. nie można zrealizować na realnych komputerach, bo nie mają one nieskończonej taśmy, chodzi o to, że program jest idealny oraz sama maszyna jest idealna, niezmienna bezawaryjna (kto z nas nie spotyka ludzi przekonanych o takich idealnych komputerach?). Wszystko to sprawia, że zwolennicy Turinga przeżywają niejedną frustrację, gdy przychodzi im liczyć na realnym sprzęcie oraz współpracować z realnymi informatykami i uczonymi, nie zaś z jakąś Popperowską epistemologią bez podmiotu poznającego.

Nie należy tego odczytywać pejoratywnie (socjalnie), ale raczej obiektywnie. Tak jak przyrządy fizyczne obiektywizują nam jakościowe doświadczenia zmysłowe, a także dostarczają danych w obszarach, w których nasze zmysły po prostu nie działają, podobnie komputery funkcjonują w obszarze tego, co można nazwać obiektywnie matematyką dyskretną[18], która może nie jest całą matematyką,

---

will run on. When the 32-bit versions of TABLES and STSDAS are no longer viable, support for those packages will end".

[17] Więcej na ten temat, np. w wieloautorskiej: *Highlights of Astronomy*, t. 11A, red. J. Andersen, (As Presented at the XXIIIrd General Assembly of the IAU, 1997), Dordrecht, Boston, London 1998, s. 192n, 457n; G. Klare, *Reviews in Modern Astronomy*, t. 2, Berlin 2012, s. 236.

[18] Zob. K. A. Ross, C. R. B. Wright, *Matematyka dyskretna*, Warszawa 1999.

ale taką, która może być obliczana na komputerach i jednocześnie dawać wgląd w całą matematykę. Tak jak przyrządy dają nam wgląd w fizykę świata, podobnie informatyka, choćby ta rozumiana jako matematyka dyskretna, daje nam wgląd w matematyczność świata.

Tak właśnie rozumiemy rolę paradygmatu obiektowego, który dziś przeżywa swój rozwój. Nie zawsze interpretuje się go w podany przez nas filozoficzny sposób. Na nasze potrzeby, należy więc opisać jego jedynie podstawowe właściwości.

W podejściu obiektowym, wszystkie komputerowe nazwy, metody itp. odnoszą się do dziedziny naukowego rozwiązywania problemu (nie do *hardware'u*). Mamy więc tu sytuację analogiczną do tego, jak jest w matematyce i fizyce – gdy piszemy równanie $F=ma$, wtedy F łączy matematyczną formę z fizycznym pojęciem *siły*, podobnie m – masy oraz a – przyspieszenia, które jest swoistą matematyczną, obiektową metodą, mianowicie drugą pochodną po czasie położenia r. Jak wiemy, jest bardzo niewiele równań matematycznych (w dziedzinie czystej matematyki), które mają takie filozoficzne (fizyczne) znaczenie, które jest podstawą fizyki matematycznej. Podobnie jest wiele kodów binarnych, ale bardzo niewiele takich, które są naukowo sensowne.

Po drugie, program obiektowy „staje się" w czasie rzeczywistym, tzn. metody działające na obiektach nie są znane podczas kompilacji (działania informatyka); mogą być one „wtyczkami" do systemu obiektowego, co więcej – użytkownik takiego systemu też może być swoistą wirtualną wtyczką, a skoro osoba nie jest Turingowsko obliczalna, zatem system obiektowy z osobowym użytkownikiem stanowi autentyczne *novum* relacyjne w systemie informatycznym. Zauważmy także, iż w realnych systemach operacyjnych chcielibyśmy, aby system nam ciągle towarzyszył bez „stopu", czyli nie chcemy, by był Turingowską Maszyną[19].

[19] Czasem określa się system operacyjny jako „kopertę" na programy rozumiane jako Maszyny Turinga, które mają zgłosić systemowi swoje zakończenie. Jednak system operacyjny nie ma takich wymagań, bo nie ma się zakończyć jak Turingowska Maszyna – ma się zapętlić. Recenzent chciałby usunąć owo „zapętlenie się", ponieważ – jego zdaniem – system operacyjny nie powinien się zapętlać i mieć określone warunki zakończenia swej pracy (np. wciśnięcie przycisku „zamknij sys-

Po trzecie, obiekty (a ściślej ich klasy) uszczegóławiają ogólne typy, tworząc hierarchie polimorficznie usystematyzowane. Możliwe jest dziedziczenie metod wirtualnych pomiędzy właściwymi poziomami uszczegółowienia w odniesieniu i wyłączności przypisanej do tegoż poziomu w rozległej nieraz polimorficzności pojęcia (naukowego). W filozofii odniesienia tego typu określane są jako różne typy analogii.

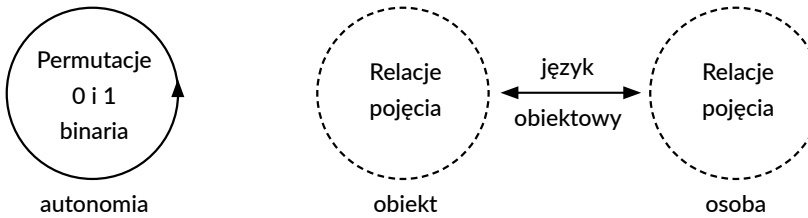
W końcu – system obiektowy jest przygotowany na to, że oddziałują w nim na siebie pojęcia-obiekty/metody, a nie fizyczne manipulatory bitowe, jak tego chciał Turing. Często Turingowskie obliczanie w ogóle nie jest istotne dla naukowego problemu, gdyż jest nim ustalenie się właściwych relacji pojęciowych. Jeśli system obliczeniowy satysfakcjonuje „włączonego" doń użytkownika, wszystko wydaje się po Turingowsku „zaprogramowane" w sensie odczytu stanu *hardware'u*; jeśli jednak użytkownik Turignowski chciałby zmodyfikować system, musi zejść na język bitów, zaś użytkownik obiektowy jest w naturalnej relacji językowej z systemem i modyfikuje jego metodę wirtualną, wzbogacając sam system, nie usuwając niższego poziomu.

Innymi słowy, informatycy, tworząc programy binarne, tworzą osobne „byty", zaś system obiektowy jest jak rosnąca roślina w relacji z otoczeniem – staje się większy, adaptując się do danych okoliczności zewnętrznych. Informatyk nie jest tu herosem, mającym ogarnąć programowo wszystko, ale specjalnym użytkownikiem, który wchodzi w lokalną relację z systemem obiektowym, dostarczając metod zrozumiałych przez sprzęt, podobnie uczony jest tym lokalnym użytkownikiem, który pośrednio daje dane, np. są nimi teorie matematyczne, fizyczne, itd. Obaj spotykają się w obszarze pojęć. Podobnie jak matematyk nie potrzebuje obliczać np. nieskończonego rozwinięcia liczby Π, aby ją poprawnie stosować i rozumieć, podobnie uczony, nie musi wszystkiego obliczać materialnie na komputerze, gdyż wystarcza taki poziom komunikacji, który daje wystarczający wgląd w niematerialne rozumienie międzyludzkie. Dzieje się tak dlatego, że pośrednikiem między osobami (programistami, uczonymi) jest język, który dyscyplinuje z jednej strony *hardware*, by spełniał

---

tem"). Jednakże zatrzymany system operacyjny nie jest już systemem operacyjnym, a wciśnięcie przycisku pochodzi od użytkownika, nie Maszyny.

funkcje wg. swej binarnej natury matematyki dyskretnej, oraz osobę uczonego, dla której język obiektowy jest pewną formą myśli prawdziwej, bo odniesionej do natury badanego świata i do matematyki ciągłej (szerszej niż dyskretna).



Sytuację tę przedstawia powyższy rysunek: autonomiczny system obliczeniowy (część materialnego świata) niejako z założenia w czasie działania nie jest podatny na życzenia informatyka, zaś dla systemów obiektowych sam język dyscyplinuje obiekty oraz osoby, które z tego systemu korzystają (są w niego włączone, wchodzą z nim w interakcję określoną relacjami językowymi). Można zatem powiedzieć, że system obiektowy należy do kultury, nie może być rozpatrywany jedynie materialnie, co go zasadniczo odróżnia od systemów Turingowskich.

Na konkretne poparcie naszych obiektowo-językowych tez przytoczymy informatyczne argumenty Andrew Williams'a, który nie wchodzi, co prawda, w analizy o charakterze filozoficznym, niemniej jednak pośrednio na ich potrzebę wyraźnie wskazuje[20]. Sięgając do historii, przedstawia on m.in. elementarny problem zgodności pomiędzy różnymi sposobami kodowania znaków. Może to wydawać się dziś dziwne, ale był to poważny problem w wymianie danych w latach siedemdziesiątych.

Wybitny informatyk bardzo trafnie wskazuje istotę problemu, który nęka omawiany już IRAF, mianowicie jest nim jego własny

[20] Zob. A. Williams, *PLUG: Python in Astronomy*, https://www.youtube.com/watch?v=XVTV_VG3rok, 2012. Wykładowca pracuje w zespole radio-astronomicznym: Murchison Widefield Array (MWA).

preprocesor SPP. W klasycznym (Turingowskim, binarnym) podejściu, zdaniem Williamsa, utrwaliła się koncepcja oprogramowania działającego według reguły: dane dają wyniki (liczby). W tym celu skonstruowano *języki*: C, FORTRAN, które były powszechnie wykorzystywane w oprogramowaniu astronomicznym: AIPS, MIDAS, VISTA, DAOPhot, DoPhot, itd. Analiza DoPhot pokazuje całą problematykę takiej metodologii, mianowicie oprogramowanie to, stworzone przez Polaków, zostało przekazane Francuzom, a następnie Anglikom, co sprawiło, że zawiera ono polskie identyfikatory oraz trochę ich francuskich objaśnień, więc możemy sobie wyobrazić, jak na ten stan rzeczy reagują informatycy angielscy.

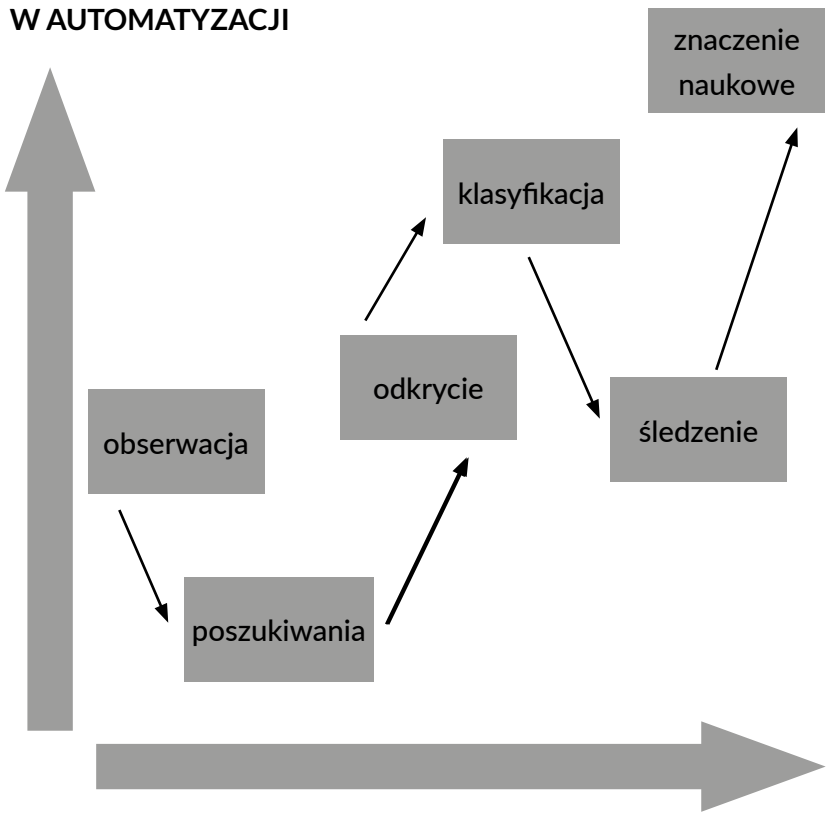
W przeciwieństwie do klasycznego, Turingowskiego modelu obliczalności Williams wskazuje, że obiektowy język Python przedstawia inne podejście: kod jest wykonywany na bieżąco, bez kompilacji, wyposażony w algebrę numeryczną i pakiety dające dostęp do wszystkiego, co jest dostępne systemowi informatycznemu. W takim obiektowym podejściu możliwe jest opracowanie np. wzorca, umieszczenie go w bazie danych, dzięki czemu reszta (obserwacja oraz redukcja danych) mogą być automatyczne[21]. Tak powstaje wspomniany projekt «astropy.org», który wykorzystuje moc programowania obiektowego w astronomii[22].

Ważną analizę problemów oprogramowania przedstawia Joshua Bloom, w wykładzie *Python as Super Glue for the Modern Scientific Workflow*. Nawiązuje on praktycznie do tego, czym cechuje się paradygmat obiektowy jako taki. W odniesieniu do astronomii widzianej obiektowo, cały Kosmos stanowi zmienną dziedzinę badań. Również dane astronomiczne cechują się tym, że trudno je mierzyć – pojęcia takie jak odległość, energia, skale czasowe (zob. wyżej) itp. w astronomii stają się niewspółmierne z tym, co rozumie się przez pomiar

[21] Dodajmy, *quasi*-automatyczne, tzn. powtarzalne w relacji z bezpośrednio je interpretującym specjalistą z dziedziny problemu, astronomem, nie informatykiem. Podobnie jak hardware uwolnił się od analogowego stechnicyzowania, dzięki wprowadzeniu logicznych stanów (0,1), tak uczeni uwalniają się od informatyków, dzięki językowi zdolnemu działać na komputerze i w ich dziedzinie.

[22] Zob. Wykłady, które prowadził Yogesh Wadadekar, w szczególności dziewiąty. Zob. Erik Tollerud (Yale University), *The Astropy Project in 2015*.

na Ziemi. Z drugiej strony, ilość tych danych staje się jakby nie do ogarnięcia, nawet przez specjalistyczny zespół. Zmieniające się modele i teorie astronomiczne sprawiają, że same sposoby ujmowania rzeczywistości astronomicznej są płynne[23]. I wreszcie to, co dla klasycznego informatyka jest chyba największym ciosem: automatyzacja w podejściu do istotnych danych jest bardzo problematyczna. Boom przedstawia tę sytuację na poniższym diagramie:



[23] Słyszy się, że ciemna energia/masa zmuszą nas do napisania podręczników na nowo.

Jak widać, poszukiwanie obiektów obserwowanych w masie danych informatycznych może być najlepiej zautomatyzowane, ale wymaga się tu inteligentnego uczonego, a nie obserwującego robota. W naukowym odkryciu składnik racjonalności odkrywcy napotyka na większe bariery, a jeszcze trudniej jest sklasyfikować to, co się odkryło. W końcu znaczenie naukowe, wymagające największego stopnia inteligencji twórczej, ma największe bariery automatyczne.

Możemy więc śmiało stwierdzić, że choć uczymy systemy coraz to lepszego rozpoznawania szczególnych przypadków, to naukowa istota problemu pozostaje wciąż niezmiennie aktualna: pojęcie rodzące się w uczonym nie jest wynikiem mechanicznych manipulacji liczbowych.

3. Nieobliczalność pojęć

„Γρηγορεῖτε οὖν, ὅτι οὐκ οἴδατε τὴν ἡμέραν οὐδὲ τὴν ὥραν”[24]. Starożytnym ludziom dzień kojarzył się z ruchem Słońca po niebie. Nie była to poprawna interpretacja. Do dziś nie wiemy, czym jest czas, choć potrafimy go mierzyć. Mamy więc filozoficzny problem podstawowej natury, nierozstrzygalny na drodze obliczeń.

Podobny problem z obliczalnością – znali go już starożytni Grecy – dotyczył niewymierności $\sqrt{2}$. Prosta logika zdemaskowała podstawową nieznajomość natury liczb, przyczyniając się do odkrycia liczb głębszych – niewymiernych, których nigdy nie poznamy. Tu jeszcze wyraźniej niż w astronomii (eksperyment/obserwacja) uwydatnił się problem odkrycia racjonalnego. Wartości dokładnej $\sqrt{2}$ nie obliczy żaden komputer, żadna Maszyna Turinga, a odkrycie (odważnie używamy tego słowa) jednego takiego $\sqrt{\cdot}$ uzmysłowiło matematykom nieprzeliczalność całego zbioru liczb rzeczywistych – wskazało na rzeczywistość o wiele głębszą niż ta poznawana za pomocą liczb naturalnych i wymiernych (a co powiedzieć o bogactwie liczb zespolonych…). W ten sposób dochodzimy do hierarchii pięter zależnych od metod ujmowania nieskończoności $\square^i$; odkryliśmy filozoficznie

[24] Mt 25, 13; „Czuwajcie zatem, gdyż nie znacie dnia ani godziny”.

obiektowy polimorfizm nieskończoności. Być może po to, aby zrozumieć wspomniany czas, potrzebne jest doświadczenie (filozoficzne) wieczności.

Pewien drobny wgląd w kłopoty z nieskończonością daje nam matematyka. Zauważmy, że zegar taktujący Turingowską Maszyną można by tak zregularyzować funkcją ζ -Riemanna, że rozbieżna suma nieskończona $1+1+\cdots$ działania maszynowego w sensie $\zeta(0)=\frac{-1}{2}$ jest skończona. Można by to tak zinterpretować, że wynik nieskończonych zliczeń koniecznych do nieturingowskich obliczeń znany jest tuż przed uruchomieniem procesu obliczeniowego, tzn. że wszystko z góry jest już obliczone, choć nie przez skończone komputery. Argument ten może być ujmowany na wiele sposobów i z pewnością zasługuje na szersze opracowanie wykraczające poza ramy pracy skoncentrowanej raczej na astronomii.

Zakończenie

Każdy uczony, używając do swej pracy systemów komputerowych, pragnie wykorzystywać je jak najlepiej. W wielu przypadkach chodzi o jak najbardziej efektywne wykonywanie matematycznych obliczeń, co zakłada, że w ogóle takie matematyczne podejście do problemu jest znane. Wyraża się to jednakże wiarą, że równania, które się informatyzuje, rzeczywiście stosują się do rozwiązywanego zagadnienia, oraz że same obliczenia komputerowe nie psują tych równań, tzn. konieczne przybliżenia są zbieżne do realnego, matematycznego wyniku.

W zagadnieniach związanych m.in. z kontekstem odkrycia naukowego stawia się wiele hipotez, podejmuje się wiele strategii badawczych, aby ogarnąć w jakiś logiczny sposób niepełne (jakże często) lub przybliżone dane obserwacyjne. Astronom nie może „ustawiać" eksperymentu tak, jak to robi np. fizyk. Oznacza to, że wiele dziedzin astronomii tworzy modele *ad hoc*. Pojawianie się nowych metod badawczych w tak błyskawicznym tempie sprawia, że oprogramowanie z wczoraj nie bardzo potrafi sprostać problemom, które mamy dzisiaj. Dziedzina tego samego zagadnienia zmienia się płynnie.

Programowanie binarne, nastawione na zobrazowanie realnego problemu w świecie algorytmu jako takiego, wymaga tego, że do jego ujęcia konieczny jest wyspecjalizowany super-informatyk, który umiałby przełożyć hipotezę naukową na język zrozumiały dla *hardware'u*. Informatycy (i niektórzy filozofowie) marzą o materialnych komputerach całkowicie niezależnych od racjonalności człowieka. Jak wykazaliśmy, są to złudzenia.

Dzięki temu, że ontologia dziedziny problemu może być opisana w paradygmacie programowania (podejścia) obiektowego, a metody obliczeniowe przedstawione jako wirtualne metody stosowane w danej dyscyplinie, możliwe jest programowanie łączące w sobie zalety systemu komputerowego oraz danej dyscypliny astronomicznej w spójny sposób. Tworzy się w ten sposób specyficzna algebra opisu problemu naukowego w symbolicznym języku programowania obiektowego, która jednocześnie steruje procesem obliczeniowym. Pośrednikiem jest tu język porządkujący relację między obiektami (m.in. komputerowym sprzętem) a osobami (uczonymi zdyscyplinowanymi wymaganiami obiektywnego języka), odnoszącymi swe pojęcia do rzeczywistej prawdy.

Wyewoluowanie jakiejś metody, np. przez zastosowanie lepszej procedury przybliżającej realną matematykę, jest w podejściu obiektowym naturalne, choć nie zawsze proste, dokonuje się bowiem jako „przeładowanie" nową wirtualną metodą obliczeniową w otwartym zbiorze metod nieznanych w czasie kompilacji, pierwotnej implementacji programu. Choć semantyka programowania w ogóle nie obrazuje problemu w świecie procedur binarnych, to język sprawia, że relacja znaczeniowa między uproszczonym światem binariów, a środowiskiem zewnętrznym, nie ginie, tzn. wynik binarny podlega (prawie) natychmiastowej interpretacji poprawnej w języku dziedziny problemu.

Metodologia obiektowa ponosi jednak pewną stratę – istnieje realne zwiększenie kosztu czasu obliczeniowego, co może być też znacznie kłopotliwe, gdy system ma stać się dedykowanym do praktycznych obliczeń bardziej niż do teoretycznych odkryć. Wtedy trzeba uciec się, niestety, nawet do *hardware'owych* rozwiązań specjalistycznych. Niemniej jednak w trakcie właściwego, wiedzotwórczego

rozwiązywania problemów, czas na ludzki namysł wielokrotnie przewyższa czas komputerowej weryfikacji.

Paradygmat obiektowy sprawia, że system komputerowy staje się partnerem dialogu człowieka ze światem idei matematycznych bez konieczności pośrednich pięter technicznych[25], których stopień złożoności jest tak wielki, że żaden uczony nie byłby jej w stanie choćby w części ogarnąć wtedy, gdyby uczony miał swoje problemy formułować w języku bitów.

Utwierdza to naszą tezę, że to właśnie dzięki językowi wchodzimy w relacje z zewnętrznymi obiektami (przedmiotami), z których najprostsze są bitami, a niebinarny kod określa semantycznie, co miałoby zostać w ogóle obliczone; język programowania dyscyplinuje także nasze indywidualistyczne, ludzkie myślenie.

Wreszcie, zabierając się do obliczeni wierzymy, że wynik istnieje bez względu na to, czy w ogóle te obliczenia przeprowadzimy. Tezę tę popieramy (trywialnym) faktem: to co obserwujemy w astronomii (nauce) po prostu jest, więc obliczenia powinny być zbieżne, gdyż sterować nimi powinna superweniencyjna metoda pochodząca z istnienia badanego obiektu ujmowana przez matematyczny język.

Summary

The Object Oriented Paradigm in Astronomical Software

To create an astronomical software, one should use the most effective method to produce the quickest hardware calculation. However, the not so effective object oriented paradigm seems to have bigger influence on the astronomical domain. We discuss the *status quo* of its methodology. The traditional, Turing computation model does not have so distinct influence on knowledge creation because it is hardware-oriented. An object oriented programming language is more influential on discovery, because it works directly in the scientific domain.

[25] Recenzent zwrócił uwagę, że skuteczne programy wymagają tych pięter. Właśnie o tym jest mowa wyżej, gdy chodzi np. o *hardware'owe* rozwiązania specjalistyczne. Oprogramowanie obiektowe, choć technicznie bardziej powolne, już teraz od góry (superwenientnie) określa niematerialne wymagania stawiane przez racjonalizm naukowy. To jest tak, jak z jakimś twierdzeniem o istnieniu rozwiązania: zawsze warto go poszukiwać, choć obecnie znane metody są nieskuteczne.

Keywords: *object oriented paradigm*, *scientific software*, Turing Machine, *knowledge*, *mathematical model*

Bibliografia

AstroConda, http://astroconda.readthedocs.io/en/latest/faq.html#why-isn-t--iraf-installed-by-default.

Bloom J., *Python as Super Glue for the Modern Scientific Workflow*, https://www.youtube.com/watch?v=mLuIB8aW2KA, 2012.

Chinnici I., *Astronomia i wizje świata*, „Rocznik Filozoficzny Ignatianum" 19/2 (2013), s. 82–106.

Frege G., *Sens i znaczenie*, w: *Pisma semantyczne*, tłum. B. Wolniwicz, Warszawa 1977.

Herodot, *Dzieje*, tłum. A. Bronikowski, Poznań 1862.

Highlights of Astronomy, t. 11A, red. J. Andersen,, Dordrecht, Boston, London 1998 (As Presented at the XXIIIrd General Assembly of the IAU, 1997).

Kelley D. H., Milone, E. F., *Exploring Ancient Skies. A Survey of Ancient and Cultural Astronomy*, New York 2011.

Klare G., *Reviews in Modern Astronomy*, t. 2, Berlin 2012.

Ligonnière R., *Prehistoria i historia komputerów*, Wrocław 1992.

Montenbruck O., Pfleger T., *Astronomy on the Personal Computer*, tłum. S. Dunlop, Berlin, Heidelberg 2013.

Mt [św. Mateusz Ewangelista], w: *The New Testament in the original Greek*, red. B. F. Westcott, F. J. A. Hort, New York 1885;

The Gospel according to Matthew, w: *The New Testament in the original Greek*, red. B. F. Westcott, F. J. A. Hort, New York 1885, s. 3–71.

Penrose R., *Nowy umysł cesarza. O komputerach, umyśle i prawach fizyki*, Warszawa 1996.

Platon, *Timaios*, tłum. W. Witwicki,.

Roskal Z. E., *Obserwacyjne versus eksperymentalne testy ogólnej teorii względności*, w: *Albert Einstein i rewolucja relatywistyczna*, red. Z. Pietrzak, Wrocław 2016, s. 51–70; (Lectiones & Acroases Philosophicae, IX, 1).

Ross K. A., Wright, C. R. B., *Matematyka dyskretna*, Warszawa 1999.

Tollerud E. *The Astropy Project in 2015*, https://www.youtube.com/watch?v=1b7lZdfKA6k, 2015.

Williams A., *PLUG: Python in Astronomy*, https://www.youtube.com/watch?v=XVTV_VG3rok, 2012.

Wadadekar Y., *Python for astronomical data analysis*, https://www.youtube.com/channel/UC-yInul-JV9-tbHLZuba4LQ, https://www.youtube.com/watch?v=6Sg9rCCXkvc, 2014.

Wróblewski A. K., *Historia fizyki od czasów najdawniejszych do współczesnych*, Warszawa 2007.