



International JOURNAL of SOCIAL and HUMANITIES SCIENCES RESEARCH (JSHSR)

Uluslararası Sosyal ve Beşeri Bilimler Araştırma Dergisi

Received/Makale Gelis

05.01.2023

<http://dx.doi.org/10.26450/jshsr.3506>

Published /Yayınlanma

28.02.2023

Research Article

Volume/Issue (Sayı/Cilt)-ss/pp

10(92), 434-439

ISSN: 2459-1149



Arş. Gör. Emre Musa TURHAN



<https://orcid.org/0000-0003-3139-5052>



Niğde Ömer Halisdemir Üniversitesi, Eğitim Fakültesi, Eğitim Bilimleri Bölümü, Niğde / TÜRKİYE

BİLGİ İŞLEMSEL DÜŞÜNME COMPUTATIONAL THINKING

ÖZET

Bilgi işlemsel düşünme becerilerinin geliştirilmesi, teknoloji odaklı bir dönüşüm yaşayan 21. yüzyıl toplumunda başarı elde etmek adına giderek daha da hayati bir önem kazanmaya başlamıştır. Bu çalışmada bilgi işlemsel düşünme; gerekliliği, dayandığı temeller, alt boyutları ve açısından ele alınmıştır. Bilgi işlemsel düşünme, bilgisayar ve bilgisayar bilimi kavramlarının ortaya çıkmasından bu yana varlığını sürdüren bir kavramdır. Bilgisayar programcılığına benzer bir şekilde problemleri bilişsel olarak parçalara ayırmaya ve analiz etmeye olanak tanıyan “bilgi işlemsel düşünme, bireylerin problem çözme becerilerinin gelişmesini kolaylaştıran bir süreç olarak düşünülmektedir. Bilgi işlemsel düşünme, herkesin öğrenmeye ve uygulamaya istekli olacağı bir beceri bütünü olarak kabul edilmektedir ancak bilişsel becerileri geliştirdiğine dair az sayıda çalışma ve kanıt bulunmaktadır.

Anahtar Kelimeler: Bilgi işlemsel düşünme, soyutlama, örüntü, algoritma.

ABSTRACT

The development of computational thinking skills has become more and more vital in order to gain success in the 21st-century society, which is undergoing a technology-oriented transformation. In this study, computational thinking; necessity, the foundations on which it is based, its sub-dimensions, and its aspects are discussed. Computational thinking is a concept that has existed since the emergence of the concepts of computer and computer science. Computational thinking, which allows one to break down and analyze problems cognitively similar to computer programming, is considered a process that facilitates the development of problem-solving skills in individuals. Computational thinking is considered a skill set that everyone will be willing to learn and practice, but there are few studies and evidence that it improves cognitive skills.

Keywords: computational thinking, abstraction, pattern, algorithm.

1. GİRİŞ

Sürekli olarak gelişen küresel ekonomi, problem çözme ve eleştirel düşünme becerileri ile donatılmış bir iş gücüne ihtiyaç duymaktadır. Günümüz dijital dünyasında, teknolojinin çocukların yaşamları ve gelişimleri üzerindeki etkisi her geçen gün artmaktadır. Birçok ülkede çocuklar iki yaşından önce akıllı teknolojileri kullanmaya başlamaktadırlar (AVG Technologies, 2018; Gerson, Morey ve van Schaik, 2022; OECD, 2010). Ailelerin teknoloji algıları, çocukların teknolojiye erişimleri ve kullanımı üzerinde önemli bir etkiye sahiptir (Hartle, 2019). Eğitimi, yaşamlarımızı ve eğitim anlayışımızı büyük ölçüde etkileyen sosyo-dijital devrimle uyumlu hale getirmek için gelişen bir ihtiyaç doğmuştur (Ezeamuzie & Leung, 2022). Bu nedenle, teknolojinin çocukların gelişimlerine pozitif yönde katkılarını sağlamaya yönelik eğitim programları ve politikalarının geliştirilmesi çok önemlidir. Friedman’a (2005) göre dünya, 21. yüzyılın önemli engellerinin üstesinden gelmek için belirli bir alanda ve teknolojiye uzmanlığa sahip, spesifik problemler çözebilmek adına bilgi işlem becerilerini alana özgü becerileriyle birleştirebilen, değişen durumlara hızlı adapte olabilen ve çok yönlü bireylere ihtiyaç duymaktadır (s. 289).

Çalışma ve iletişim kurma biçimimizin merkezinde yer alan bilgi işleme, bilim, mühendislik, ticaret ve tıp alanlarında, çevremizde olanları dramatik bir şekilde etkileyen bir değişim yaratmıştır. Bilgi işleme teknolojileri sayesinde, neredeyse her alanda ilerlemeler gerçekleştirilmiştir. Fakat bunu gerçekleştirenlerin çok azı resmi olarak bilgisayar bilimcisi olarak eğitilmiştir. Bu eğilimin devam etmesi için, öğrencilerin yalnızca içerik tüketiminde değil, aynı zamanda içerik oluşturmada da bilgiyi işleme becerisini kazanmaları gerekmektedir (Stokke, 2019). Bugünün öğrencilerine, hangi alanda ilerlemeyi seçerlerse seçsinler hazır olabilmeleri için bilgisayarları bir problem çözme aracı olarak nasıl kullanacakları öğretilmelidir (Resnick, Bruckman, & Martin, 1996; Wing, 2006). Bu bağlamda öğrencilere bilgisayarlar ve diğer teknolojilerle problem çözmeyi kolaylaştıran düşünce süreçlerinin gelişimini desteklemeye yönelik eğitim verilmesi daha verimli sonuçlar alınmasını sağlayacaktır. Bilgi işleme teknolojisi ile içerik tüketmekten ziyade, içerik

oluşturabilen bireyler, kendi fikirlerini ifade edebilme becerisini de aynı zamanda geliştirmiş olacaktır (Stokke, 2019). Bilgisayarla içerik oluşturmanın, teknoloji kullanımının çok ötesine uzanan birçok avantajı vardır. Öğrencilerin bilgisayar programcılığı ile ele alınabilecek sorunları mantıklı bir şekilde parçalara ayırmasına ve analiz etmesine olanak tanıyan “bilgi işlemsel düşünmeyi” öğrenmeleri bu avantajlardan birisi olarak düşünülebilir (Papert, 1996; Wing, 2006).

2. BİLGİ İŞLEMSEL DÜŞÜNMENİN TEMELLERİ

Bilgi işlemeyi ya da diğer adıyla hesaplamayı 'zihinsel bir araç olarak' olarak kullanma fikri, bilgisayar ve bilgisayar bilimi kavramlarının ortaya çıkmasından bu yana var olmuştur (Forsythe, 1959). Örneğin, 1960lı yıllarda Alan Perlis, ana branşlarda öğrenim gören üniversite öğrencilerinin, matematik ve ekonomi gibi farklı alanlardaki anlayışlarını yeniden şekillendirebilmek adına "hesaplama teorisini" kullanabilmeleri için programlamayı öğrenmeleri gerektiğine inanıyordu (Guzdial, 2008, s. 25). Denning (2009), bilgi işlemsel düşünmenin birtakım problemleri veya sorunları çözmek için "algoritmik düşünme" olarak geçmişte zaten var olduğunu ileri sürmüştür (s. 28). Günümüzde algoritmalar ve algoritmik düşünme, bilgi işlemsel düşünmenin sadece küçük bir bölümünü oluşturmaktadır.

“Bilgi işlemsel düşünme” kavramını ilk olarak kullanan kişi, 1980 yılında çocuklarla birlikte "Logo" adındaki programlama diliyle ilgili bir araştırma yürüten Seymour Papert isimli bilim insanıdır. Papert (1980), "Mindstorms: Children, Computers, and Powerful Ideas" adlı kitabında bilgi işlemsel düşünmenin günlük yaşamın çeşitli yönleriyle olan ilgisine değinmiştir. Papert, daha sonra bu kavramı daha da derinleştirmiş ve bilgi işlemsel düşünmenin amacının matematiksel fikirleri daha erişilebilir ve daha güçlü hale getirmek olduğunu öne sürmüştür. Bilgi işlemsel düşünmenin bireye bilgi oluşturma gücü verdiğini ve düşünme becerilerini artırmaya yardımcı olduğunu açıklamıştır (Papert, 1996). 2006'da Jeanette Wing, bilgi işlemsel düşünme kavramının daha popüler hale gelmesine sebep olan bir çalışma yayınlamıştır. Wing, bilgi işlemsel düşünmeyi, bilgisayar biliminde yer alan kavramlardan yararlanıp, bilgisayar bilimciler veya bilgisayar programcıları gibi düşünerek problem çözme, sistem tasarlama ve insan davranışlarını anlamlandırma olarak tanımlamıştır. Wing'e (2008) göre, evrensel olan ve hayati bir önem taşıyan bilgi işlemsel düşünme, erken çocukluk döneminde her bireye öğretilmesi gereken bir beceridir. Ayrıca Wing (2008), bilgi işlemsel düşünmenin, beşeri bilimler ve sanatla ilgili mesleklerde başarılı olma fırsatı vererek, araştırmacılara veri madenciliği aracılığıyla insan yaşamındaki yeni eğilimleri keşfetme olanağı verdiğini belirtmiştir. Sussman, bilgi işlemsel düşünmeyi, karmaşık olan şeyleri yönetebilme ve bunun otomatik hale gelebilmesini için belirli bir dizi komut belirleme aracı olarak tanımlamıştır (Lee vd, 2011; National Research Council, 2010). Bogost ise bilgi işlemsel düşünmenin, bilgisayar programlamanın aksine, problem çözmek için kullanılan süreç ve prosedürlerin anlaşılmasıyla ilgili olduğunu belirtmiştir (Vee, 2013). 2009'da Bilgisayar Bilimi Öğretmenleri Derneği (CSTA) ve Uluslararası Eğitimde Teknoloji Derneği (ISTE), Ulusal Bilim Vakfı'nın finansmanı ile birlikte, bilgi işlemsel düşünmenin işlevsel bir tanımını oluşturmak için bir proje başlatarak, bilgi işlemsel düşünmenin aşağıdaki adımları içeren bir problem çözme süreci olduğu sonucuna vardılar:

- Problemleri makinelerin çözebileceği şekilde formüle etme
- Verileri mantıksal olarak organize etme ve analiz etme
- Soyutlamalar kullanarak bir süreçle ilgili verileri tanımlama
- Bir sorunu çözmek için kullanılacak sıralı bir dizi adım belirleme
- Çözümler için verimlilik elde etme
- Benzer problem çözme yolunu diğer durumlara uyarlamak için genelleme yapma (Computer Science Teachers Association and the International Society for Technology in Education, 2011).

Barr ve Stephenson (2011), bilgi işlemsel düşünmeyi bilgisayar aracılığıyla uygulanabilecek bir problem çözme süreci olarak tanımlamış, ayrıca mantıkla ilgili alanlarda da kullanılabilirliğini söylemişlerdir. Bahsettikleri tanımdan yola çıkarak, bilgi işlemsel düşünmenin adımlarını şu şekilde sıralamışlardır:

- Veri toplama
- Veri analizi
- Soyutlama
- Otomatik hale getirme

- Test etme ve doğrulama
- Algoritma geliştirme
- Problem çözümlenme
- Kontrol yapıları oluşturma
- Simülasyon

Bu adımları oluşturduktan sonra bilgi işlemsel düşünmenin;

- Karmaşık problemlerle uğraşırken güven verdiğini,
- Belirsizliğin üstesinden gelme becerisi kazandığını,
- Açık uçlu problemlerle başa çıkabilme yetisini geliştirdiğini,
- Ortak bir amaç veya çözüm için birlikte çalışırken, farklılıkları bir kenara bırakmayı öğrettiğini,
- Grup çalışması içinde, bireyin kendi güçlü ve zayıf yönlerini tanımasına olanak sağladığını dile getirmişlerdir.

Grover ve Pea (2013), Wing'in 2006 tarihli çalışmasında yer alan bilgilere dayanarak soyutlama basamağının, bilgi işlemsel düşünmeyi diğer düşünme biçimlerinden ya da becerilerinden ayıran şey olduğunu belirtirler ve yaygın olarak bilgi işlemsel düşünmeyi oluşturduğu kabul edilen unsurları şu şekilde ifade etmişlerdir (s. 39):

- Soyutlama ve örüntü genelleme
- Bilgiyi sistematik olarak işleme süreci
- Sembollerle temsil etme
- Algoritma oluşturma
- Problemi yapılandırarak çözümlenme
- Etkililik
- Hata ayıklama ve tespit etme

Brennan ve Resnick (2012) ise bilgi işlemsel düşünmenin tasarıma dayalı öğrenme etkinlikleri ile nasıl desteklenebileceği yönünde çalışmalar yürüterek bilgi işlemsel düşünmeyi kavram uygulama ve perspektif boyutlarından oluşan bir çerçeve altında toplamışlardır. Seiter ve Foreman'a (2013) göre bilgi işlemsel düşünmenin bileşenleri şu şekilde belirtilmiştir:

- Prosedürler ve algoritmalar
- Problem ayrıştırma
- Paralleleştirme ve senkronizasyon
- Soyutlama
- Verinin temsil edilmesi

Kalelioğlu, Gülbahar ve Kukul (2016), bilgi işlemsel düşünme tarihinin henüz bir olgunluk aşamasında olduğunu belirtmiş ve bu becerinin veya düşünme biçiminin henüz tam olarak tanımlanamayacağını belirtmişlerdir. Ayrıca bilgi işlemsel düşünmenin nasıl öğretilmesi ve bu düşünme becerisinin nasıl ölçüleceği hakkında yeteri kadar bilgi birikiminin oluşmadığını öne sürmüşlerdir. Yaptıkları incelemelerden yola çıkarak bilgi işlemsel düşünmeyi "bilgisayarlı veya bilgisayarın olmadığı" durumlarda da kullanılabilen bir problem çözme süreci olarak tanımlamışlardır (s. 583-592). Bu doğrultuda bilgi işlemsel düşünmeyi oluşturan unsurları şu şekilde sıralamışlardır:

- Problemi tanımlama-Soyutlama ve çözümlenme
- Veri toplama ve analiz-Veriyi kavramsallaştırma ve temsil etme
- Çözümleri seçme ve planlama-Matematiksel mantık yürütme, algoritma oluşturma ve paraleleştirme

- Çözümleri Uygulama-Otomatik hale getirme ve modelleme
- Çözümleri değerlendirme-Test etme, hata ayıklama ve genelleme

Bilgi işlemsel düşünmenin farklı şekillerde ele alınan tanımları, bileşenleri ya da basamakları göz önünde bulundurulduğunda, parçalara ayırma, soyutlama, örüntü, algoritma ve değerlendirme-hata ayıklama basamakları çoğu çalışmada değinilen ortak boyutlar olarak karşımıza çıkmaktadır.

3. BİLGİ İŞLEMSEL DÜŞÜNMENİN BOYUTLARI

3.1. Parçalara Ayırma

Parçalara ayırma basamağı, karmaşık bir problemin alt boyutlara ayrılarak kolay bir şekilde ele alınmasını sağlayan bir kavramdır (Grover ve Pea, 2013). Eğer bir problem çözüme kavuşturulamıyorsa, yeteri kadar parçalara ayrılma yapılmadığı belirtilmektedir (bbc.com, 2017). Google Education tarafından yayınlanan bir çalışmada, ele alınan bir şiirin analiz edilebilmesi için, ahenk unsurlarına, tonlamaya ve kafiyelerine göre parçalara ayrılması öne sürülmüştür. Lego Education (2018) ise ana teması tatil olan bir problemin, kalınacak yerler, bilet fiyatları, tatile çıkarken bulundurulması gereken eşyalardan örnek vererek parçalara ayırma boyutunu daha da somutlaştırmıştır.

3.2. Soyutlama

Soyutlama basamağı, problemin odak noktasında ne gibi detayların bulunması ile ilgilidir ve problemin içeriğine konsantre olmayı daha da kolay bir hale getirir. Soyutlama basamağında, problemin asıl olarak ele alınması gereken yönleri daha belirgin bir hale getirilerek, gereksiz detaylar daha geri plana atılır (Wing, 2008). Grover ve Pea (2013), problemin anlaşılır hale gelmesi ve çözüme ulaşabilmesi için soyutlama basamağının önemli bir role sahip olduğunu dile getirmişlerdir.

3.3. Örüntü

Örüntü, belirli bir sıraya göre nesnelerin ya da durumların birbirini izleyerek ilerlemesi anlamına gelmektedir ve daha çok matematik ile ilgili alanlarda karşımıza çıkar. Bilgi işlemsel düşünmede örüntü, örüntü tanıma, örüntü/model çıkarma gibi terimler kullanılmaktadır. Google Education (2017) tarafından sunulan tanıtımda, verilerdeki örüntüler, eğilimler ve düzenleri gözlemek olarak ele alınmıştır. Kimyasal reaksiyonlar esnasında oluşan sıra ve ülke ekonomisi ile ilgili yükseliş-düşüş trendleri, örüntüye verilen örnekler arasında yer almaktadır. Diğer bir deyişle olaylar arasındaki döngüler örüntüleri oluşturmaktadır.

3.4. Algoritma Oluşturma

Bilgi işlemsel düşünmenin en önemli boyutlarından olan algoritma oluşturma, 9. Yüzyılda yaşayan ünlü Türk matematikçi Harizmi tarafından ilk defa temel anlamda ele alınan bir kavram olarak karşımıza çıkmıştır (Knuth, 1985). Algoritma, problemler için girdi verilerini almak ve istenilen çıktıları oluşturmak için işlemlerin adım adım tasarlanmasını içeren soyut bir süreçtir (Wing, 2008). Futschek'e (2006) göre algoritma birtakım talimatlardan oluşan bir süreç iken; Selby ve Woolard (2013), algoritmayı bir görevin belirli bir sıraya göre tamamlanması olarak tanımlamıştır. Bu bağlamda algoritma kavramı sadece bilgisayar bilimiyle ilişkili değil aynı zamanda bilgi işlemsel düşünmenin de bir parçası olarak düşünülmelidir.

3.5. Değerlendirme ve Hata Ayıklama

Değerlendirme boyutunda, probleme yönelik oluşturulan çözüm yolu optimum düzeyde verimli olabilmesi açısından ele alınıp incelenmektedir (Csizmadia vd., 2015). Weiser'e (1982) göre bu basamakta, adım adım gidilen yollar tekrar değerlendirilir ve eğer istenmeyen veya hatalı bir yol varsa bu hata ayıklama yapılarak bu durumlar ortadan kaldırılır. Bu kavram genellikle bilgisayar biliminde algoritma oluşturduktan sonra, kodların daha verimli bir hale getirilmesinde kullanılan bir kavramdır.

4. SONUÇ

Bilgi işlemsel düşünme, bilgisayar ve bilgisayar bilimi kavramlarının ortaya çıkmasından bu yana varlığını sürdürmektedir. Araştırmacılar, 1960'lı yıllarda matematik ve ekonomi gibi ana branşlarda kullanılması gerektiğini iddia etmişlerdir. Alt boyutları veya basamakları farklı biçimlerde ele alınmış olsa da parçalara ayırma, soyutlama, örüntü, algoritma ve değerlendirme-hata ayıklama basamakları çoğu çalışmada değinilen ortak boyutlar olarak karşımıza çıkmaktadır. Bilgi işlemsel düşünme, evrensel bağlamda ele alındığında yalnızca bilgisayar bilimcilerinin değil herkesin öğrenmeye ve uygulamaya istekli olacağı bir tutum ve beceri bütünü olarak karşımıza çıkmaktadır. Buna rağmen, alanyazında bilgi işlemsel düşünmenin bilişsel becerileri geliştirdiğine dair az sayıda çalışma ve kanıt bulunmaktadır.

KAYNAKÇA

- AVG (2018). <https://now.avg.com/digital-abilities-overtake-key-development-milestones-for-todays-connected-children> (Erişim tarihi: 01.01.2023)
- Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48-54.
- Brennan, K., & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, 1–25.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). *Computational thinking: A guide for teachers*. Computing At School.
- CSTA & ISTE (2011). Computational Thinking in K–12 Education leadership toolkit. <http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>.
- Denning, P. J. (2009). Beyond Computational Thinking. *Communications of the ACM*, 52(6), 28-30
- Ezeamuzie, N. O., & Leung, J. S. (2022). Computational thinking through an empirical lens: A systematic review of literature. *Journal of Educational Computing Research*, 60(2), 481-511.
- Friedman, T. L. (2005). *The world is flat: A brief history of the twenty-first century*. New York: Farrar, Straus and Giroux.
- Forsythe, G. E. (1959). The Role of Numerical Analysis in an Undergraduate Program. *The American Mathematical Monthly*, 66(8), 651–662.
- Futschek, G. (2006, November). Algorithmic thinking: the key for understanding computer science. In *International conference on informatics in secondary schools-evolution and perspectives*, 159-168. Springer, Berlin, Heidelberg.
- Gerson, S. A., Morey, R. D., & van Schaik, J. E. (2022). Coding in the cot? Factors influencing 0–17s' experiences with technology and coding in the United Kingdom. *Computers & Education*, 178, 104400. <https://doi.org/10.1016/j.compedu.2021.104400>
- Google Education (2017). Computational Thinking For Educators.
- Grover, S. & Pea, R. D. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.
- Guzdial, M. (2008). Paving the Way for Computational Thinking. *Communications of the ACM*, 51(8), 25-27.
- Hartle, L. C. (2019). *Technology and Young Children: Processes, Context, Research, and Practice*. In L.E. Cohen & S. Waite-Stupiansky (Eds.), *STEM for Early Childhood Learners: How Science, Technology, Engineering and Mathematics Strengthen Learning*, New York, NY: Routledge.
- Kalelioğlu, F., Gülbahar, Y. & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4 (3), 583-596.
- Knuth, D. E. (1985). Algorithmic thinking and mathematical thinking, *The American Mathematical Monthly*, 92(3), 170-181.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.
- LEGO Education (2018) .WeDo 2.0 Computational thinking teachers guide.
- National Research Council (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.
- Organisation for Economic Co-operation and Development [OECD] (2010). What do we know about children and technology? Educational Research and Innovation, OECD Publishing, Paris. <https://www.oecd.org/education/ceri/Booklet-21st-century-children>
- Papert, S. (1980). *Mindstorms: Children, Computers, And Powerful Ideas*. New York: Basic Books.

- Papert, S. (1996). An exploration in the space of mathematics educations. *Int. J. Comput. Math. Learn.*, 1(1), 95-123.
- Resnick, M., Bruckman, A., & Martin, F. (1996). Planos not stereos: Creating computational construction kits. *Interactions*, 5(5), 40-50.
- Selby, C.C. & Woollard, J. (2013). Computational thinking: the developing definition. *University of Southampton* (E-prints), 1-6.
- Seiter, L., & Foreman, B. (2013). Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* 59–66. doi:10.1145/2493394.2493403
- Stokke, T. L. (2019). *An Exploratory Study to Identify Barriers to Implementation of Computational Thinking*. Doctoral dissertation, The University of North Dakota.
- Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, 1(2), 42-64.
- Weiser, M. (1982). Programmers use slices when debugging. *Communications of the ACM*, 25(7), 446-452.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.