

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

BMJ PUBLISHING GROUP LIMITED,

Petitioner,

v.

EVIDIUM, INC.,

Patent Owner.

Case No. PGR2027-00001

U.S. Patent No. 12,518,882

**DECLARATION OF MARK A. MUSEN, M.D., Ph.D. IN SUPPORT OF
PETITION FOR POST GRANT REVIEW OF U.S. PATENT NO. 12,518,882**

TABLE OF CONTENTS

I. INTRODUCTION.....	1
II. QUALIFICATIONS AND EXPERIENCE	3
A. Overview of Qualifications and Experience.....	3
B. Research and Development	4
C. Teaching Experience	8
D. Service to the Scientific Community.....	9
E. Honors and Awards	10
III. PERSON OF ORDINARY SKILL IN THE ART	10
IV. APPLIED LEGAL PRINCIPLES	12
A. Prior Art	12
B. Anticipation	13
C. Obviousness	14
D. Claim Construction Standard.....	15
E. Patent-Eligible Subject Matter.....	16
F. Written Description	17
G. Burden of Proof	18
V. MATERIALS AND PRIOR ART CONSIDERED.....	18
VI. TECHNOLOGY BACKGROUND.....	22
A. Medical Ontologies and Terminologies	23
1. SNOMED-CT.....	23
2. LOINC.....	26
3. Other Ontologies and Terminologies.....	27
B. Medical Journals.....	27
C. Knowledge Graphs in Healthcare.....	28
D. Clinical Decision Support Systems	32
E. APIs for Accessing Medical Knowledge Graphs.....	34

VII. OVERVIEW OF THE ‘882 PATENT	35
A. Summary of the ‘882 patent	35
B. Prosecution History of the ‘882 patent	40
VIII. CLAIM CONSTRUCTION	45
IX. OVERVIEW OF PRIOR ART REFERENCES.....	45
A. Datla (Ex. 1005).....	45
B. Goldberg (Ex. 1006).....	46
C. Gnanasambandam (Ex. 1007).....	47
D. Dang (Ex. 1008).....	47
E. Zhong (Ex. 1009).....	48
F. Van Assel (Ex. 1011).....	49
G. Kim (Ex. 1012).....	50
X. INVALIDITY ANALYSIS OF THE CHALLENGED CLAIMS	51
A. Ground 1: Claims 1-3, 8-9, and 11-14 are Obvious Over Datla, Goldberg, and Gnanasambandam.....	51
1. Motivation to Combine	51
2. Independent Claim 1	54
3. Dependent Claim 2.....	80
4. Dependent Claim 3.....	83
5. Dependent Claim 8.....	84
6. Dependent Claim 9.....	85
7. Dependent Claim 11.....	88
8. Dependent Claim 12.....	93
9. Dependent Claim 13.....	96
10. Dependent Claim 14	97
B. Ground 2: Claims 4-5 are Obvious Over Datla, Goldberg, Gnanasambandam, and Kim.....	99
1. Motivation to Combine	99

2. Dependent Claim 4.....	101
3. Dependent Claim 5.....	105
C. Ground 3: Claim 6 is Obvious Over Datla, Goldberg, Gnanasambandam, and Van Assel.....	108
1. Dependent Claim 6.....	108
D. Ground 4: Claims 7 and 15-20 are Obvious Over Datla, Goldberg, Dang, and Gnanasambandam.....	110
1. Motivation to Combine	110
2. Independent Claim 15	112
3. Dependent Claim 16.....	136
4. Dependent Claim 17.....	141
5. Dependent Claim 18.....	144
6. Dependent Claim 19.....	145
7. Dependent Claim 20.....	146
8. Dependent Claim 7.....	147
E. Ground 5: Claims 21-41 Lack Adequate Written Description.....	150
F. Ground 6: Claims 21-22 and 26-31 are Obvious Over Datla, Dang, Gnanasambandam, and MacRae	153
1. Motivation to Combine	153
2. Independent Claim 21	154
3. Dependent Claim 22.....	166
4. Dependent Claim 26.....	167
5. Dependent Claim 27.....	170
6. Dependent Claim 28.....	173
7. Dependent Claim 29.....	175
8. Dependent Claim 30.....	177
9. Dependent Claim 31	177

G. Ground 7: Claims 23 and 24 are Obvious Over Datla, Dang, Gnanasambandam, MacRae, and Kim	178
1. Dependent Claim 23.....	179
2. Dependent Claim 24.....	181
H. Ground 8: Claim 25 is Obvious Over Datla, Dang, Gnanasambandam, MacRae, and Van Assel	183
1. Dependent Claim 25.....	183
I. Ground 9: Claims 32-33 and 38-41 are Obvious Over Zhong, Datla, Gnanasambandam, and MacRae	185
1. Motivation to Combine	185
2. Independent Claim 32	189
3. Dependent Claim 33.....	223
4. Dependent Claim 38.....	224
5. Dependent Claim 39.....	226
6. Dependent Claim 40.....	226
7. Dependent Claim 41	227
J. Ground 10: Claim 34 is Obvious Over Zhong, Datla, Gnanasambandam, MacRae, and Dang	228
1. Dependent Claim 34.....	229
K. Ground 11: Claims 35 and 36 are Obvious Over Zhong, Datla, Gnanasambandam, MacRae, and Kim	233
1. Dependent Claim 35.....	233
2. Dependent Claim 36.....	236
L. Ground 12: Claim 37 is Obvious Over Zhong, Datla, Gnanasambandam, MacRae, and Van Assel	237
1. Dependent Claim 37.....	237
M. Ground 13: Claims 1-9 and 11-41 are Directed to Ineligible Subject Matter 239	
1. The Challenged Claims are Directed to an Abstract Idea.....	239

2. The Challenged Claims Do Not Recite an Inventive Concept	255
3. Conclusion.....	264
XI. SECONDARY CONSIDERATIONS	265
XII. ADDITIONAL REMARKS	265

I, Mark A. Musen, hereby declare as follows:

I. INTRODUCTION

1. I have been retained on behalf of BMJ Publishing Group Limited (“Petitioner”) to provide my expert opinion on the patentability of claims 1–9 and 11–41 (the “Challenged Claims”) of U.S. Patent No. 12,518,882 (“the ‘882 patent”). I understand that Petitioner is filing a Petition for Post Grant Review (“PGR”) of the ‘882 patent before the Patent Trial and Appeal Board (“PTAB” or “Board”) of the United States Patent and Trademark Office (“USPTO”).

2. A detailed description of my professional qualifications, including a listing of my specialties, expertise, and professional activities, is contained in my curriculum vitae, a copy of which is provided as Exhibit 1004. A summary of my professional qualifications is set forth below.

3. I am being compensated for my work in this matter at my standard consulting rate of \$900 per hour. I am also being reimbursed for reasonable expenses that I may incur in connection with this matter. My compensation is in no way contingent upon the results of my study, the substance of my opinions or testimony, or the outcome of this proceeding.

4. Based upon the facts and information available to me to date, I detail my opinions relevant to this Petition for Post Grant Review below. I am prepared to

provide expert testimony on the opinions formed herein if asked about those issues by the Board or others.

5. In forming my opinions, I have relied upon my own knowledge, training, and experience in the fields of biomedical informatics, knowledge representation, medical ontologies, and clinical decision support systems. I may additionally discuss my own work, teachings, and knowledge of the state of the art in the relevant time period. I may also rely on handbooks, textbooks, technical literature, and the like to demonstrate the state of the art in the relevant period and the evolution of the technology at issue.

6. Throughout this Declaration, I refer to specific passages, line numbers, or page numbers of the relevant prior art or other technical documents. The citations are intended to be exemplary as opposed to being exhaustive and are not intended to convey that the citations are the only source of evidence to support the propositions for which they are cited.

7. My analysis is ongoing, and I reserve the right to revise, supplement, and amend my opinions based on new information that may become available, including any arguments or evidence that Patent Owner may present in this proceeding.

II. QUALIFICATIONS AND EXPERIENCE

A. Overview of Qualifications and Experience

8. I am currently a Professor of Biomedical Data Science and a Professor of Medicine (Computational Medicine) at Stanford University School of Medicine, where I also serve as Chief of the Division of Computational Medicine. I have held my appointment at Stanford since 1988. In 2022, I was appointed the Stanford Medicine Professor of Biomedical Informatics Research, an endowed chair.

9. My career at Stanford spans more than four decades. From 1995 to 2002, I served as Associate Professor of Medicine (General Internal Medicine, Medical Informatics). From 1989 to 2011, I held a concurrent appointment as Assistant Professor, then Associate Professor, and then Professor of Computer Science (by courtesy). From 1988 to 1995, I was an Assistant Professor of Medicine (General Internal Medicine). Before joining the Stanford faculty, I served as a Clinical Instructor in the Department of Medicine at Stanford University School of Medicine from 1984 to 1987, during which time I was also a Ph.D. candidate in the Medical Information Sciences Program (1983–1987) and a Henry J. Kaiser Family Foundation Fellow in General Internal Medicine at Stanford (1983–1987). I completed my residency in Internal Medicine at Stanford University Hospital (1981–1983) and my internship in Internal Medicine at Stanford University Hospital (1980–1981).

10. I received my Sc.B. in Biology from Brown University in 1977 and my M.D. from Brown University in 1980. I completed my Ph.D. in Medical Information Sciences at Stanford University in 1988. I am a Diplomate of the National Board of Medical Examiners (Certificate No. 227602, 1981) and a Diplomate of the American Board of Internal Medicine (Certificate No. 091133, retired). I hold a California Medical License (No. G044898, inactive).

11. I have spent over four decades conducting research and teaching in the fields of biomedical informatics, knowledge representation, ontology engineering, and clinical decision support. My research has centered on the development of methods and tools for creating and managing biomedical knowledge bases and ontologies—the very technologies that are at the heart of the ‘882 patent.

B. Research and Development

12. My research career has focused on biomedical knowledge representation, ontology development, and clinical decision support—all topics directly relevant to the technology claimed in the ‘882 patent. Among my major research endeavors:

13. **The Protégé Project (1984–present):** Since 1984, I have led the Protégé Project at Stanford, which has produced one of the world's most widely used platforms for ontology development and knowledge-base construction. Protégé is an open-source tool that enables researchers and practitioners to create, visualize, and

manage ontologies and knowledge graphs. The Protégé platform is used by hundreds of thousands of users worldwide and has been central to the development of biomedical ontologies, including those used for medical terminology standards.

14. The National Center for Biomedical Ontology (2005–2017): I served as Principal Investigator of the National Center for Biomedical Ontology (“NCBO”), an NIH-supported National Center for Biomedical Computing. Through the NCBO, I led the development of BioPortal, which became the world’s largest repository of biomedical ontologies, providing access to hundreds of biomedical ontologies including SNOMED-CT and other medical terminologies referenced in the ‘882 patent. This work is directly relevant to the ‘882 patent’s claims regarding knowledge graphs linked to medical ontologies.

15. EON Project (1988–2006): I led the EON project, which developed computer-based decision support systems for clinical protocols and guidelines. This work involved the creation of knowledge representations and reasoning methods for clinical decision-making—technology closely related to the medical reasoning capabilities claimed in the ‘882 patent.

16. The Center for Expanded Data Annotation and Retrieval (2014–present): I am Principal Investigator of CEDAR, an NIH Big-Data-to-Knowledge Initiative Center of Excellence focused on metadata standards and biomedical data integration.

17. **WHO Collaborating Centre (2014–present):** I serve as Director of the Stanford University Collaborating Centre for Classification, Terminology, and Standards for the World Health Organization. In this role, I have been directly involved in the development and maintenance of international medical classification and terminology systems, including work related to ICD-11, the International Classification of Diseases. I served as Chair of the ICD-11 Health Informatics and Modeling Topic Advisory Group (2008–2016) and as a member of the ICD-11 Revision Steering Committee (2008–2016). This work is directly relevant to the ‘882 patent’s claims concerning medical ontologies.

18. **BioPortal (2021–present):** I am currently Principal Investigator of the NIH-funded BioPortal project, an expansive knowledge base of biomedical entities and relations that provides tools for searching, browsing, and integrating biomedical ontologies and terminologies. The BioPortal platform hosts and provides API-based access to hundreds of biomedical ontologies, including SNOMED-CT and LOINC—the medical terminology systems specifically referenced in the ‘882 patent.

19. **RADx Data Hub (2022–2026):** I served as Principal Investigator of the Rapid Acceleration of Diagnostics (RADx) Data Hub, a national data repository supported by over \$44 million in NIH funding to harmonize and integrate COVID-related diagnostic studies.

20. Over the course of my career, I have secured tens of millions of dollars in research funding from the National Institutes of Health, the National Science Foundation, the World Health Organization, DARPA, and other sponsors for projects directly related to biomedical knowledge representation, ontologies, knowledge graphs, and clinical decision support.

21. I have authored or co-authored more than 155 refereed journal articles, numerous conference papers, and multiple book chapters and books. My publications have been cited extensively, and I have an h-index of 107 as measured by Google Scholar. My published work spans topics including biomedical ontology development, knowledge graph construction and management, clinical decision support, medical terminology standards (including SNOMED-CT), knowledge representation and reasoning, and Semantic Web technologies in biomedicine—all areas directly relevant to the technology of the ‘882 patent.

22. An h-index of 107 means that I have authored at least 107 publications that have each been cited at least 107 times by other researchers. This metric is widely used in academia to measure both the productivity and the citation impact of a scholar’s body of work. For context, an h-index above 40 is generally considered outstanding in most scientific fields, and an h-index above 60 is exceptional and is typically associated with senior researchers who have made sustained, field-shaping contributions. An h-index of 107 places my scholarly output among the most highly

cited researchers in the fields of biomedical informatics and computer science, reflecting the broad influence and adoption of my work on ontologies, knowledge representation, and clinical decision support over the course of more than four decades of research.

C. Teaching Experience

23. I have taught courses at Stanford University for over 35 years in the areas of biomedical informatics, knowledge representation, ontology engineering, and clinical decision support. I served as Professor of Computer Science (by courtesy) at Stanford from 1989 to 2011, in addition to my appointments in the School of Medicine. I have also served as a member of the Executive Committee of the Biomedical Informatics Graduate Program at Stanford from 1988 to 2026, and I served as Director of Admissions for the Medical Information Sciences Program from 1989 to 1997.

24. I have supervised the doctoral dissertations and postdoctoral research of more than 37 graduate students and postdoctoral researchers in biomedical informatics, knowledge representation, and ontology engineering. I have also served as an external examiner for 13 doctoral dissertations and habilitations at international universities, including Linköping University, Université Paris-Sud, Technical University of Vienna, University of Leipzig, Technical University of Berlin,

University of the Basque Country, Graz University of Technology, University of Barcelona, Amsterdam University Medical Center, and the University of Leiden.

D. Service to the Scientific Community

25. I have served the scientific community extensively through editorial, advisory, and leadership roles. I am a Founding Co-editor-in-chief of the journal *Applied Ontology* (2005–2016; Associate Editor 2017–present). I have served on the editorial boards of numerous journals in the fields of artificial intelligence and medical informatics, including *Artificial Intelligence in Medicine*, the *Journal of the American Medical Informatics Association*, the *Journal of Web Semantics*, the *International Journal of Human-Computer Studies*, and *Learning Health Systems*, among others.

26. I have served in leadership roles in major national and international scientific organizations. I was General Chair of the International Semantic Web Conference (2005), General Chair of the International Conference on Knowledge Capture (2011), and Program Chair of the American Medical Informatics Association Annual Symposium (2003). I have served on numerous NIH study sections and advisory boards, including the National Advisory Council on Biomedical Imaging and Bioengineering (2012–2016) and the NIH Big Data to Knowledge Centers Steering Committee (2015–2018).

E. Honors and Awards

27. I have been recognized with numerous honors and awards for my contributions to biomedical informatics and knowledge representation. I was elected a Member of the National Academy of Medicine in 2016—one of the highest honors in the fields of health and medicine. I received the Donald A.B. Lindberg Award for Innovation in Informatics (2006) and a Leadership Award (2007) from the American Medical Informatics Association. I was elected a Founding Fellow of the International Academy of Health Sciences Informatics (2017) and I received the Classic Paper Award from the Association for the Advancement of Artificial Intelligence (2018). In 2025, I received the Lifetime Achievement Award from the Knowledge Graph Conference and was named a Distinguished Fellow of the American College of Medical Informatics. I have received honorary doctorates from the University of A Coruña, Spain (2021) and the University of Fribourg, Switzerland (2023). I was elected a Fellow of the American College of Medical Informatics in 1989, a member of the American Society for Clinical Investigation in 1997, and a member of the Association of American Physicians in 2010. I also received a National Science Foundation Young Investigator Award in 1992.

III. PERSON OF ORDINARY SKILL IN THE ART

28. In evaluating the prior art references and other materials, I have used the perspective of a person of ordinary skill in the art (“POSITA”) to which the ‘882

patent is related as of the time of the patent's earliest claimed effective filing date. I am informed by Counsel that a POSITA is a hypothetical person presumed to be aware of all pertinent prior art and the conventional wisdom in the art, and who possesses ordinary creativity—not that of an automaton.

29. I understand that the ‘882 patent claims priority to U.S. Provisional Application No. 63/331,526, filed on April 15, 2022. For the purposes of my analysis, I have been asked to assume that April 15, 2022 is the effective filing date of each of the Challenged Claims. I am informed by Counsel that the relevant prior art is assessed as of the effective filing date of the patent.

30. In my opinion, a POSITA in the relevant field of the ‘882 patent, as of the April 15, 2022 priority date, would have had at least: (a) a master’s degree in biomedical informatics, computer science, health informatics, or a closely related field; and (b) at least two years of experience working with medical knowledge representation systems, biomedical ontologies, knowledge graphs, or clinical decision support systems. Alternatively, a person with a bachelor’s degree in one of these fields could qualify as a POSITA with additional years of relevant industry or research experience in these areas. Additional education could compensate for less professional experience, and vice versa.

31. A POSITA as of the relevant date would have had a good working knowledge of: biomedical ontologies and terminologies (including SNOMED-CT

and LOINC); knowledge graph construction, management, and querying; clinical decision support methodologies; knowledge representation and reasoning techniques, including hierarchical and graph-based reasoning; application programming interfaces (APIs) for accessing structured medical knowledge; and the integration of medical evidence from multiple sources into computational systems.

32. I meet and exceed the criteria described above. I am, and have been since well before April 2022, a person with at least ordinary skill in the art pertaining to the ‘882 patent. I would have qualified as such a person for many years preceding the earliest claimed priority date of the ‘882 patent.

IV. APPLIED LEGAL PRINCIPLES

33. I am not an attorney, and I do not offer legal opinions. However, I have been informed by Counsel for the Petitioner about the following legal standards, which I have applied throughout my analysis.

A. Prior Art

34. I am informed by Counsel that a patent claim is invalid if it is not novel or is obvious in view of the prior art. I understand that under the Leahy-Smith America Invents Act (“AIA”), which applies to the ‘882 patent, prior art includes, among other things, patents and printed publications that were available before the effective filing date of the claimed invention.

35. I am informed by Counsel that in this PGR proceeding, the prior art that may be evaluated to demonstrate unpatentability includes patents and printed publications, as well as other categories of prior art available in a PGR proceeding.

36. I understand that a reference qualifies as prior art under 35 U.S.C. § 102(a)(1) if it was patented, described in a printed publication, in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention. I also understand that a reference qualifies as prior art under 35 U.S.C. § 102(a)(2) if it was described in a patent or patent application that was effectively filed before the effective filing date of the claimed invention.

B. Anticipation

37. I am informed by Counsel that a patent claim is anticipated—and thus unpatentable—under 35 U.S.C. § 102 if each and every element of the claim, as properly construed, is found, either expressly or inherently, in a single prior art reference. The prior art reference need not use the identical words or phrases used in the claim, but the reference must describe each element of the claim with sufficient specificity that a person of ordinary skill in the art would recognize the disclosure of each element.

38. I understand that a feature is considered to be “inherently” disclosed in a prior art reference if the feature is necessarily present in the reference, even if it is not expressly described. I am further informed that the standard for anticipation is

the same as the standard for literal infringement, and that a prior art reference that would literally infringe the claim also anticipates the claim.

C. Obviousness

39. I am informed by Counsel that a patent claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious to a person of ordinary skill in the art at the time the invention was made (or, under the AIA, before the effective filing date of the claimed invention). Obviousness is assessed on a claim-by-claim basis.

40. I understand that the obviousness analysis should not be limited to a rigid application of the “teaching, suggestion, or motivation” test. Rather, the analysis should take into account common sense, the knowledge and creativity of a person of ordinary skill in the art, and the inferences and connections that such a person would draw from the prior art.

41. I understand that a claim may be obvious if, among other things: (a) the combination of known elements yields predictable results; (b) simple substitution of one known element for another produces predictable results; (c) known techniques are used to improve similar devices or methods in the same way; (d) a known technique is applied to a known device or method ready for improvement to yield predictable results; (e) the solution was “obvious to try” because it was chosen from

a finite number of identified, predictable solutions with a reasonable expectation of success; or (f) the work of the art led to variations of the prior art and market forces or design incentives prompted the adaptation.

42. I am informed by Counsel that the obviousness inquiry considers four factors: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims at issue; (3) the level of ordinary skill in the pertinent art; and (4) objective indicia of nonobviousness, sometimes referred to as secondary considerations, such as commercial success, long-felt but unsolved needs, failure of others, copying, unexpected results, praise by others, and licensing.

43. I understand that a prior art reference may “teach away” from a claimed combination if the reference criticizes, discourages, or otherwise discredits the claimed combination. However, mere preference for an alternative embodiment does not constitute teaching away. A reference does not teach away merely because it suggests an alternative approach or expresses a preference for a different solution.

D. Claim Construction Standard

44. I am informed by Counsel that in a PGR proceeding before the PTAB, claim terms are given their ordinary and customary meaning as would be understood by a person of ordinary skill in the art in light of the specification and prosecution history of the patent. I understand that the claims of the patent are read in view of

the specification, although limitations from the specification are not to be imported into the claims.

45. Unless I expressly state otherwise, I have applied the ordinary and customary meaning to each claim term in this Declaration, as understood by a person of ordinary skill in the art in view of the specification and prosecution history of the ‘882 patent.

E. Patent-Eligible Subject Matter

46. I am informed by Counsel that 35 U.S.C. § 101 provides that a patent may be obtained for “any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof.” I understand that the Supreme Court has held that laws of nature, natural phenomena, and abstract ideas are not patentable under Section 101.

47. I understand that there is a two-step framework for determining whether a claim is directed to patent-eligible subject matter. At step one, the court determines whether the claim is directed to a patent-ineligible concept (i.e., an abstract idea, law of nature, or natural phenomenon). If so, at step two, the court examines the elements of the claim—both individually and as an ordered combination—to determine whether the claim contains an “inventive concept” sufficient to transform the nature of the claim into a patent-eligible application.

48. I note that the claims of the '882 patent were initially rejected by the patent examiner under 35 U.S.C. § 101 and were subsequently allowed only after the applicant amended the claims to add specific limitations relating to API-based querying and outputting results to a software application for generating medical determinations. The relevance of the prosecution history to the scope and validity of the claims is discussed further below.

F. Written Description

49. I understand that 35 U.S.C. § 112 imposes certain requirements on the specification of a patent, including a written description requirement.

50. I understand that to satisfy the written description requirement, the patent specification must convey with reasonable clarity to those skilled in the art that, as of the filing date sought, the patentee was in possession of the claimed invention. I understand that the test for written description requires an objective inquiry into the four corners of the specification from the perspective of a person of ordinary skill in the art.

51. I understand the specification must actually or inherently disclose each and every element of a claim for that claim to meet the written description requirement. I understand that, based on that inquiry, the specification must describe an invention understandable to that skilled artisan and show that the inventor actually

invented the claimed invention. It is insufficient that undisclosed subject matter would have been obvious to a person of ordinary skill in the art.

52. I understand the written description requirement cannot be satisfied if the specification expresses a mere wish or plan for obtaining the claimed invention. I understand that actual possession or reduction to practice outside of the specification is not enough. Rather, I understand that it is the specification itself that must demonstrate possession of the claimed subject matter.

G. Burden of Proof

53. I am informed by Counsel that in this PGR proceeding, the petitioner bears the burden of proving that the challenged claims of the '882 patent are unpatentable by a preponderance of the evidence.

V. MATERIALS AND PRIOR ART CONSIDERED

54. In forming my opinions, I have considered the '882 patent and its prosecution history, the prior art references discussed below, and the other materials listed in this section. I have also relied upon my own knowledge, training, and decades of experience in the fields of biomedical informatics, knowledge representation, medical ontologies, knowledge graphs, and clinical decision support systems.

55. A list of the materials I have considered includes, but is not limited to, the following:

- The ‘882 patent (Ex. 1001);
- The prosecution history of the ‘882 patent (Ex. 1002);
- U.S. Patent Publication No. 2019/0252074 (Ex. 1005, “Datla”);
- U.S. Patent No. 11,636,949 (Ex. 1006, “Goldberg”);
- U.S. Patent No. 12,248,826 (Ex. 1007, “Gnanasambandam”);
- U.S. Patent Publication No. 2022/0208373 (Ex. 1008, “Dang”);
- U.S. Patent No. 11,194,860 (Ex. 1009, “Zhong”);
- “Mining Disease-Symptom Relation from Massive Biomedical Literature and Its Application in Severe Disease Diagnosis,” Eryu Xia et al., IBM Research (2018) (Ex. 1010, “Xia”);
- U.S. Patent Publication No. 2021/0233658 (Ex. 1011, “Van Assel”);
- U.S. Patent Publication No. 2024/0221952 (Ex. 1012, “Kim”);
- J. MacRae et al., “Access primary care Big Data: the development of a software algorithm to explore the rich content of consultation records,” BMJ Open (2015) (Ex. 1013, “MacRae”);
- Evidium Claim Chart for the ‘882 Patent (Excerpted) (Ex. 1019);
- U.S. Patent Publication No. 2022/0310260 (Ex. 1020, “Ullaskrishnan”);
- U.S. Patent Publication No. 2019/0026437 (Ex. 1021, “Syeda-Mahmood”);
- U.S. Patent No. 12,512,193 (Ex. 1022, “Marcjan”);
- U.S. Patent Publication No. 2023/0377748 (Ex. 1023, “Yang”);

- U.S. Patent No. 11,488,713 (Ex. 1024, “Velez”);
- U.S. Patent Publication No. 2018/0108443 (Ex. 1025, “Li”);
- U.S. Patent No. 12,014,135 (Ex. 1026, “Potts”);
- U.S. Patent No. 7,260,480 (Ex. 1027, “Brown”);
- Dan McCreary, Knowledge Graphs: “The Third Era of Computing” (Mar. 23, 2019) (Ex. 1028, “McCreary”);
- “DocKG: A Knowledge Graph Framework for Health with Doctor-in-the-Loop,” Ming Sheng et al. (Ex. 1029, “Sheng”);
- “Learning a Health Knowledge Graph from Electronic Medical Records,” Maya Rotmensch et al. (July 20, 2017) (Ex. 1030, “Rotmensch”);
- “Semantic Health Knowledge Graph: Semantic Integration of Heterogenous Medical Knowledge and Services,” Longxiang Shi et al. (Feb. 12, 2017) (Ex. 1031, “Shi”);
- “Clinical Decision-Support Systems,” Mark A. Musen et al. (2021) (Ex. 1032, “Musen Textbook”);
- “Artificial Intelligence in Medical Diagnosis,” Peter Szolovits et al. (1988) (Ex. 1033, “Szolovits”);
- “*Internist-I*, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine,” Randolph A. Miller et al. (1982) (Ex. 1034, “Miller”);

- “INTERNIST-I Properties: Representing Common Sense and Good Medical Practice in a Computerized Medical Knowledge Base,” Fred E. Masarie et al., Computers and Biomedical Research (1985) (Ex. 1035, “Masarie”);
- “Towards computer-assisted maintenance of medical knowledge bases,” Dario A. Giuse, Artificial Intelligence in Medicine (1990) (Ex. 1036, “Giuse”);
- “Automated Encoding of Clinical Documents Based on Natural Language Processing,” Carol Friedman et al. (2004) (Ex. 1037, “Friedman”);
- SNOMED CT Starter Guide (Version 2014-12-02) (Dec. 2014) (https://web.archive.org/web/20160718140111/http://doc.ihtsdo.org/download/doc_StarterGuide_Current-en-US_INT_20141202.pdf) (Ex. 1038);
- About LOINC, LOINC from Regenstrief (<https://web.archive.org/web/20220125211233/https://loinc.org/about>) (Ex. 1039);
- Mark Musen et al., “The National Center for Biomedical Ontology,” Journal of the American Medical Informatics Association (2012) (Ex. 1040);
- Natalya Noy et al., “BioPortal: ontologies and integrated data resources at the click of a mouse,” Nucleic Acids Research, 2009, Vol. 37 (May 2009) (Ex. 1041);

- History of The BMJ (<https://www.bmj.com/about-bmj/history-of-the-bmj>) (Ex. 1042);
- About – PubMed (<https://web.archive.org/web/20200309080055/https://pubmed.ncbi.nlm.nih.gov/about/>) (Ex. 1043);
- BioPortal API Documentation (<https://web.archive.org/web/20140725213752/http://data.bioontology.org/documentation>) (Ex. 1044).

56. I reserve the right to rely on additional materials in the future, including materials that may become available during the course of this proceeding or in response to any arguments or evidence presented by Patent Owner.

VI. TECHNOLOGY BACKGROUND

57. The ‘882 patent, titled “Translation of Medical Evidence Into Computational Evidence and Applications Thereof,” describes transforming unstructured medical evidence, stored in a computer system, into a computational form to create a knowledge graph including medical condition nodes, finding nodes, and investigation nodes. In this section, I provide an overview of the relevant technology that was well known to a POSITA before the earliest claimed priority date of the ‘882 patent (i.e., April 15, 2022). All of the technology concepts discussed below were well-established in the art before that date.

A. Medical Ontologies and Terminologies

58. A medical ontology is a structured, formal representation of medical knowledge that defines concepts such as medical conditions, findings, and investigations, and the relationships among them. Medical ontologies have been in use for decades and serve as foundational tools for organizing, integrating, and reasoning over medical information. Long before the priority date of the ‘882 patent, medical ontologies were widely used in clinical information systems, electronic health records, and clinical decision support applications.

1. SNOMED-CT

59. The Systematized Nomenclature of Medicine—Clinical Terms (“SNOMED-CT”) database is one of the most comprehensive and widely used clinical terminology systems in the world. *See* Ex. 1038; Ex. 1020 at ¶ [0061] (“SNOMED is considered to be the most comprehensive, multilingual clinical healthcare terminology in the world”), ¶ [0019] (“SNOMED ontology is . . . the most widely used and most complete clinical terminology.”). SNOMED-CT was first released in January 2002 and has been continuously maintained and updated since then. It is derived from prior versions of the terminology that date back to the 1960s. SNOMED is used in healthcare systems across more than 40 countries.

60. In or around 2013, the U.S. Government (specifically, the Office of the National Coordinator for Health Information Technology (“ONC”)) required that

SNOMED CT be included in EHR systems for them to be certified for Stage 2 “Meaningful Use.”

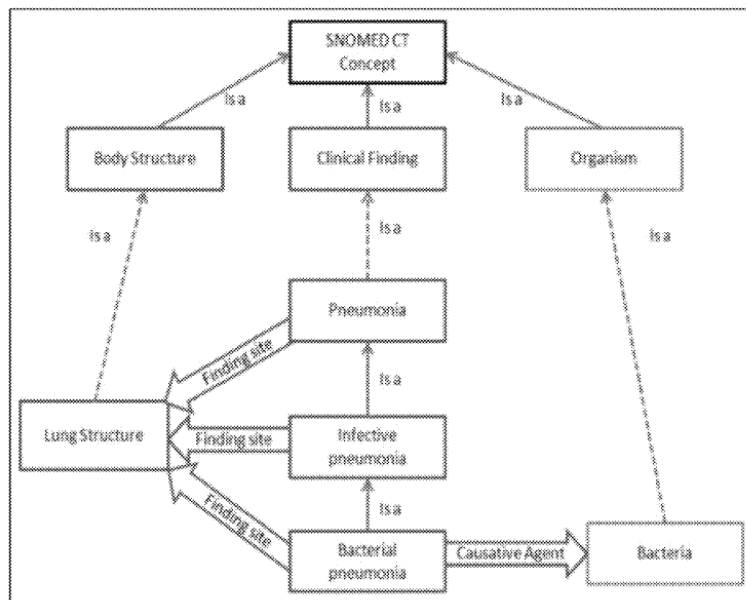
61. SNOMED-CT organizes clinical concepts including medical conditions, findings, and clinical observations. *See, e.g.*, Ex. 1006 at 1:65-2:1 (SNOMED “holds sets of relations and qualifiers that support detailed description of medical concepts, as well as names of findings and disorders that can be grouped in families”); Ex. 1020 at ¶ [0061] (“SNOMED comprehensive coverage includes: clinical findings, symptoms, diagnoses....”); Ex. 1022 at 11:7-11 (SNOMED “is used to standardize clinical findings, disorders, body structures, procedures, microorganisms, allergies, and various other clinical domains”).

62. By the time of the ’882 patent’s priority date, SNOMED-CT contained hundreds of thousands of clinical concepts organized across hierarchies of clinical findings, disorders, procedures, body structures, and other clinical domains. *See, e.g.*, Ex. 1006 at 1:54–56 (SNOMED covers “hundreds of thousands of medical terms arranged as an ontology”), 1:65–2:1; Ex. 1021 at ¶ [0030] (reporting 393,073 concepts in SNOMED CT 2011 standard); Ex. 1020 at ¶ [0061]; Ex. 1023 at ¶ [0074] (reporting 340,000 concepts in SNOMED).

63. SNOMED-CT is organized such that concepts are connected through hierarchical (“is-a”) and other relationships. *See, e.g.*, Ex. 1020 at ¶ [0022]; Ex. 1006 at 9:50–60, FIGS. 6-7. This approach enables traversal of ancestor–descendant

relationships among these concepts. *See, e.g.*, Ex. 1020 at ¶ [0022]; Ex. 1006 at 10:35–46.

64. For example, the figure below shows a SNOMED concept tree showing “is-a” edges including “Bacterial Pneumonia is-a Infective Pneumonia is-a Pneumonia.”



Ex. 1006 at FIG. 6

65. The ‘882 patent acknowledges that SNOMED-CT was a known medical terminology ontology. Ex. 1001 at 4:13–19 (explaining that “multiple clinical ontologies exist, including SNOMED-CT . . .” and “SNOMED-CT provides taxonomies of clinical terms (e.g., fever, heart valve replacements, etc.)”); *see also id.* at 5:14–19, 7:10–11, 8:48–53, 11:4–6, 12:16–21.

2. LOINC

66. The Logical Observation Identifiers, Names, and Codes (“LOINC”) database is a standard terminology for identifying medical laboratory observations, clinical measurements, and other health-related data. LOINC was first developed in 1994 by the Regenstrief Institute and has been adopted by healthcare organizations worldwide. Ex. 1039.

67. LOINC is relevant to the ‘882 patent’s investigation nodes because it codes laboratory tests and clinical observations. LOINC provides a universal coding system for laboratory tests, clinical observations, and related investigation results. *See, e.g.*, Ex. 1022 at 11:65-12:10 (“LOINC: Logical Observation Identifiers Names and Codes was created specifically to standardize the identification and reporting of medical laboratory observations, including measurements. It has been expanded to standardize clinical observations as well.”). By 2022, LOINC included codes for tens of thousands of laboratory and clinical observations.

68. The ‘882 patent acknowledges that LOINC was a known medical terminology. Ex. 1001 at 5:16–18 (describing the platform as using “logical observation identifiers names and codes (LOINC) that gives us reasoning for laboratory tests”); *see also id.* at 7:11–12 (listing “LOINC (Logical Observation Identifiers, Names, and Codes) database” among known medical evidence sources

and code systems); 20:31–40 (listing LOINC among known medical taxonomies to which EDF nodes link for inferential reasoning).

3. Other Ontologies and Terminologies.

69. In addition to SNOMED-CT and LOINC, numerous other medical ontologies and terminologies have been developed and widely used, including the International Classification of Diseases (“ICD”), RxNorm (for medications), the Gene Ontology, and the Medical Subject Headings (“MeSH”). *See, e.g.*, Ex. 1022 at 11:50-60, 12:5-15; Ex. 1021 at ¶ [0030]; Ex. 1005 at ¶ [0054]; Ex. 1024 at 4:55–65, 6:5–10.

70. My own work to develop the NCBO’s BioPortal system in the mid-2000s made hundreds of such ontologies publicly available and searchable, serving as a critical resource for the biomedical research community. Ex. 1040; Ex. 1041.

71. By the time of the ‘882 patent’s priority date, the landscape of biomedical ontologies was vast and well-established.

B. Medical Journals

72. Medical journals are peer-reviewed publications that share, among other things, clinical research findings, medical studies, and healthcare guidelines. Example medical journals include The British Medical Journal (“The BMJ”), the Journal of the American Medical Association (“JAMA”), The Lancet, and The New

England Journal of Medicine (“NEJM”). The BMJ is one of the world’s oldest medical journals, having published its first weekly edition in 1840. Ex. 1042.

73. PubMed is a free resource maintained by the National Library of Medicine that provides access to tens of millions of citations and abstracts from medical journals. PubMed provides links to full text articles when available, e.g., through PubMed Central (PMC). Ex. 1043.

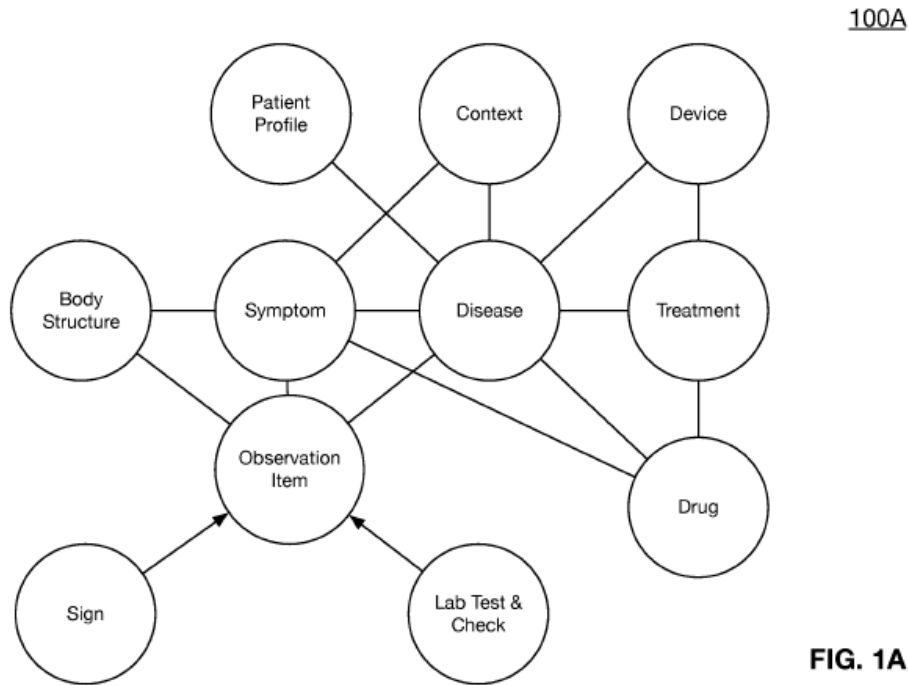
C. Knowledge Graphs in Healthcare

74. A knowledge graph is a structured representation of real-world entities (represented as “nodes”) and the relationships among them (represented as “edges”). Knowledge graphs organize information in a graph structure that enables efficient traversal, querying, and reasoning. *See, e.g.*, Ex. 1025 at ¶ [0003]; Ex. 1026 at 17:30–18:30.

75. In the healthcare domain, knowledge graphs have been used to represent medical conditions, findings (such as symptoms, signs, and test results), investigations (such as medical tests and procedures), procedures, medications, and the relationships among these entities. *See, e.g.*, Ex. 1005 at ¶ [0031]; Ex. 1006 at 6:10–30; Ex. 1011 at ¶ [0115].

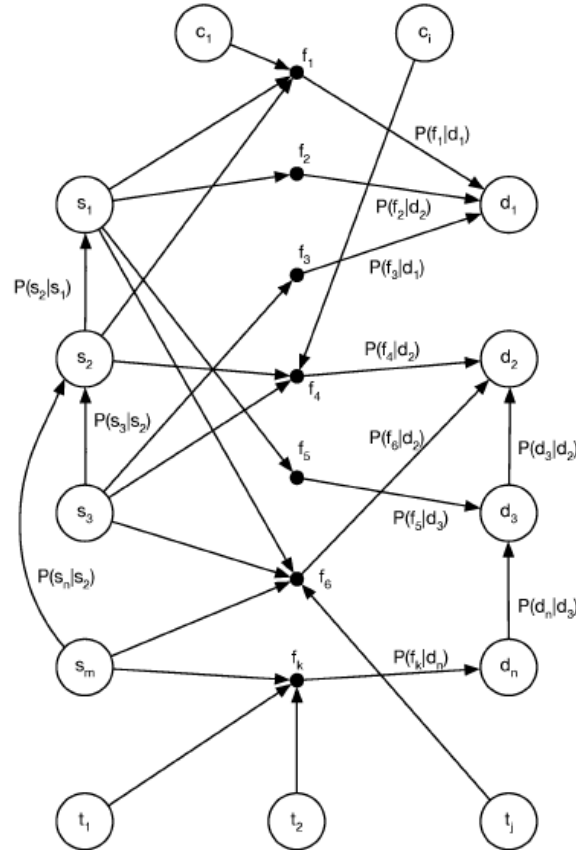
76. Medical knowledge graphs typically organize nodes into categories corresponding to different types of medical concepts, including medical condition nodes, finding nodes (which correspond to symptoms, signs, or test results

associated with one or more medical conditions), and investigation nodes (which correspond to medical tests or medical procedures associated with one or more medical conditions). For example, the graph below includes multiple node types that are connected by edges.



Ex. 1008 at FIG. 1A

77. As another example, the graph below shows how certain nodes (e.g., symptom nodes) can be linked via edges to multiple other nodes (e.g., a disease node).

100B**FIG. 1B**

Ex. 1008 at FIG. 1B

78. The relationships among these nodes, including ancestor–descendant relationships and associative relationships (e.g., “is-finding-of” or “is-test-for”), enable traversal and reasoning across the graph. *See, e.g.*, Ex. 1006 at 6:40–65, 8:5–30, 10:5–20, 10:25–50, FIGS. 6–7, 9–11; Ex. 1008 at ¶¶ [0027]–[0031], [0047]–[0050]; FIGS. 1A, 1B.

79. The use of knowledge graphs in healthcare was well-established before the ‘882 patent’s priority date. Before that date, the field was already addressing the

problem of transforming unstructured medical evidence into computational form; medical evidence has been historically represented primarily as prose documents in books and journals, and computational architectures based on text alone are not sufficient to represent, process, or reason controllably or reliably about patient-specific situations. *See, e.g.*, Ex. 1025 at ¶ [0034]; Ex. 1031 at pp. 2-3; Ex. 1030 at pp. 1-2; Ex. 1029 at p. 1; Ex. 1006 at 1:10-30; Ex. 1034 at 468-69; Ex. 1035 at pp. 458-60.

80. The construction of medical knowledge graphs from multiple evidence sources, including the medical literature (including journals like BMJ), clinical guidelines, medical ontologies (e.g., SNOMED and LOINC), electronic health records (“EHR”), and other medical evidence, was a widely practiced approach in the field. There were multiple ways to construct a knowledge graph. First, subject matter experts could manually create the knowledge graphs. Ex. 1006 at 6:5-30, 9:35-50, 10:50-60; Ex. 1034 at p. 469; Ex. 1035 at pp. 458-60.

81. Second, the knowledge graph could be created using techniques like natural language processing (“NLP”). *See, e.g.*, Ex. 1025 at ¶¶ [0032], [0034]; Ex. 1030 at pp. 1, 9; Ex. 1037 at pp. 392-93. For example, McCreary describes using machine learning to harvest relevant information to identify new knowledge for a knowledge graph that “becomes a repository of semantically precise vertices and

relationships with confidence weights retained from the machine learning processes.” Ex. 1028 at 5.

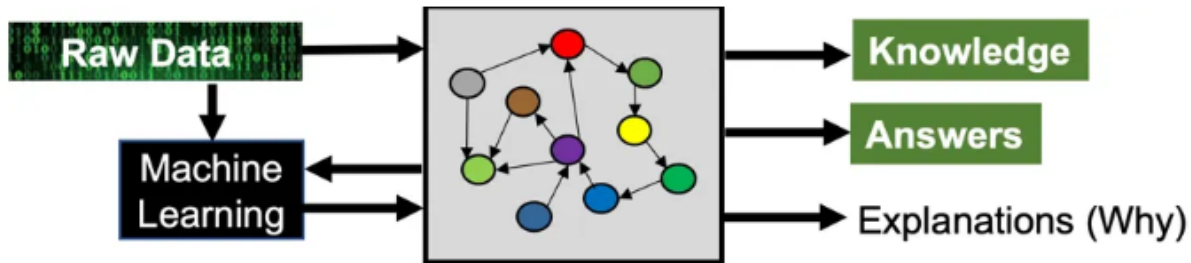


Figure 4: The Knowledge Graph Era: where machine learning continuously reads raw data, combines this with existing knowledge and produces new knowledge, answers and explanations

82. Another approach combined a human-in-the-loop with machine learning to construct knowledge graphs. *See* Ex. 1029 at pp. 1–2, 11; Ex. 1036 at 21–22, 25–26, and 29–31.

83. In sum, by 2022, the construction and use of medical knowledge graphs for integrating clinical evidence, supporting clinical reasoning, and enabling computerized decision support was already well known and widely used.

D. Clinical Decision Support Systems

84. Clinical decision support systems (“CDSS”) are computer-based tools that provide clinicians, patients, and other healthcare stakeholders with person-specific, intelligently filtered, and timely information to enhance health-related decisions. Ex. 1032 at p. 803. CDSS have been in use since the 1970s. *Id.* at p. 806; Ex. 1033 at p. 80. Among the earliest and most influential systems were MYCIN (developed at Stanford), which used rule-based reasoning to diagnose and treat

bacterial infections, and INTERNIST-1, which performed diagnostic decision support in a manner that attempted to mimic the reasoning methods of an expert physician. Ex. 1032 at p. 821; Ex. 1033 at pp. 80, 83; Ex. 1034 at pp. 468-69.

85. INTERNIST-1 was well known for modeling medical conditions and their associated findings using data structures that made explicit many of the knowledge-base distinctions described in the '882 patent, including the separate representation of medical conditions and their findings, the association of individual findings with multiple medical conditions, and the indication that certain medical conditions could occur only in the setting of specific patient characteristics (demographics). The program used its knowledge base to rank order possible medical conditions that might explain a patient's signs, symptoms, and laboratory-test results, and to suggest, through an interactive dialog, medical findings—if their values were to become known—that might allow the system to refine its list of possible medical conditions. *See* Ex. 1034 at pp. 469–70; Ex. 1035 at pp. 458–60, 462, 469–70. Although the system was developed over 50 years ago, INTERNIST-1 provided such an important demonstration of automated clinical reasoning—with performance that was deemed on par with that of expert physicians—that the system has remained a benchmark against which subsequent CDSS are frequently judged.

86. By the time of the '882 patent's priority date, CDSS had evolved substantially to incorporate knowledge graphs and ontology-based reasoning. Ex.

1032 at pp. 825–27. Modern CDSS leverage structured medical knowledge bases—including knowledge graphs linked to standard terminologies—to support diagnostic reasoning, treatment recommendations, and other clinical decisions. *Id.* at pp. 811, 825–27.

E. APIs for Accessing Medical Knowledge Graphs

87. An application programming interface (“API”) is a set of protocols, routines, and tools that allow different software applications to communicate with one another. The use of APIs to access medical knowledge bases and ontologies was well-established before the ‘882 patent’s priority date. *See, e.g.*, Ex. 1007 at 100:24–34; Ex. 1027 at 3:25–45.

88. For example, the BioPortal system that I developed at Stanford provides RESTful APIs that allow external software applications to search, browse, and query hundreds of biomedical ontologies, including SNOMED-CT and LOINC, and it has done so since it was first deployed in 2006. Ex. 1044. Similarly, the UMLS (Unified Medical Language System) maintained by the National Library of Medicine has long provided API-based access to its terminology services. The FHIR (Fast Healthcare Interoperability Resources) standard, published by HL7 International, includes terminology service APIs for accessing medical vocabularies. *See* Ex. 1032 at pp. 813, 832. By 2022, API-based querying of medical knowledge graphs to return

structured results to software applications for generating healthcare determinations was not only well known in the art but also standard practice.

89. In these systems, APIs enable a computer system to receive, via an application programming interface (API), a medical query from a software application for querying said knowledge graph, while providing the interface through which the external software application receives an output in response.

VII. OVERVIEW OF THE ‘882 PATENT

A. Summary of the ‘882 patent

90. The ‘882 patent claims priority to U.S. Provisional Application No. 63/331,526, filed April 15, 2022. It issued from a continuation of Application No. 18/446,954, filed August 9, 2023, which itself is a continuation of the application that issued as U.S. Patent No. 11,869,674 (“the ‘674 Patent”). The ‘882 patent is in the same patent family as U.S. Patent No. 12,518,883 (“the ‘883 Patent”), which was issued on the same date.

91. The ‘882 patent is entitled “Translation of Medical Evidence into Computational Evidence and Applications Thereof” and describes a “computational evidence platform” that “extracts clinical concepts from medical evidence sources and creates a database of elemental diagnostic factors and elemental investigations links to medical conditions.” Ex. 1001, Abstract. The specification frames the purported advance as converting existing narrative evidence into a “structured,

computational representation system making it useful by any application to query the evidence.” *Id.* at 2:48–50. As explained below, the specification itself identifies many of the building blocks recited in independent claims—the medical evidence sources (including published journals), the medical ontologies that supply taxonomies and hierarchies, the graph database, and the APIs—as pre-existing resources and technologies.

92. Claim 1 requires receiving “electronic text in a plurality of medical evidence sources,” where “at least a subset of said medical evidence sources includes a published journal.” Claims 21 and 32 likewise require entries that “originate in at least one medical evidence journal.” The specification describes these sources as existing third-party content and explains that “[c]urrent content providers, i.e. publishers, such as DynaMed, BMJ, UpToDate, etc., provide encyclopedia-style medical content look-up for clinicians covering diagnosis, and management of diseases,” and that they “assemble the published evidence through expert, human-centric activity—looking at published papers and published society guidelines.” *Id.* at 1:58–2:5.

93. The specification then lists the sources from which evidence is extracted, all of which pre-date the ‘882 patent: “Any of a wide variety of medical evidence sources 86 may be used and include: National Library of Medicine, JAMA Rational Clinical Evidence, BMJ Best Practice (or other British Medical Journal

databases), SNOMED-CT . . . , LOINC . . . database, CPT . . . codes, StatPearls content, . . . and other online sources such as Merck Manuals, WebMD, NORD (raredisease.org), the CDC (for endemic information), the WHO, etc.” *Id.* at 7:5–20. The specification is not limited to any particular source; the system “can extract clinical items from any textual data source.” *Id.* at 7:20–22. The sources are in conventional formats, “such as a JSON file, XML file, or other electronic format,” or Web sites “marked up in HTML.” *Id.* at 7:22–25. Figure 2 depicts this input simply as a generic “Evidence Source” box (86) feeding an NLP engine (74). *Id.* at FIG. 2.

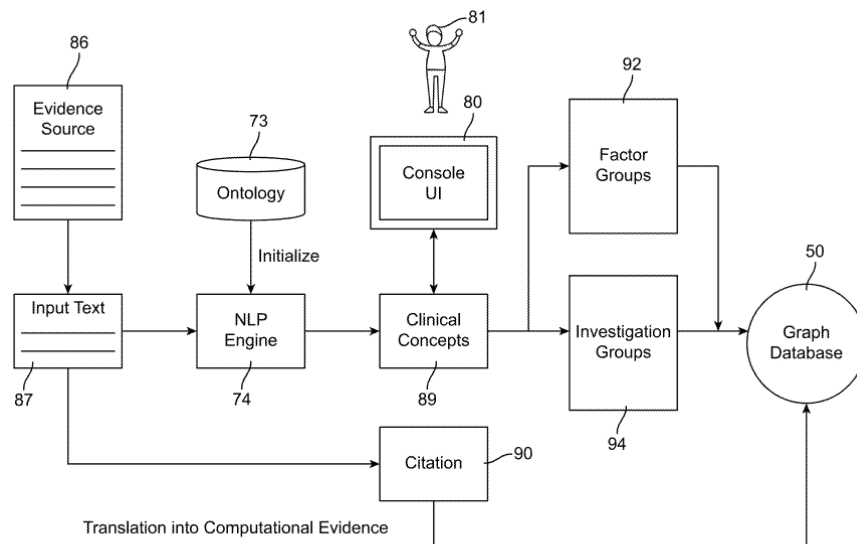


FIG. 2

Ex. 1001 at FIG. 2

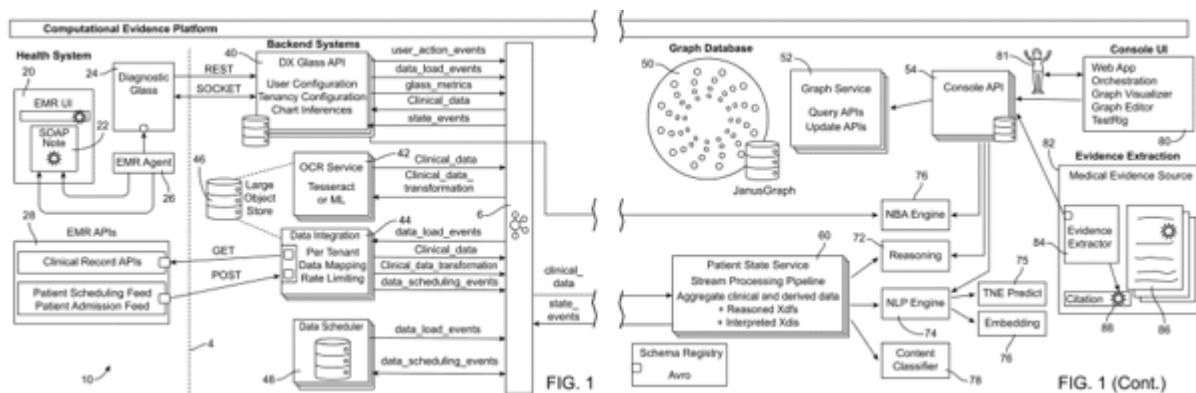
94. Claim 1 recites that the medical evidence sources “provide different taxonomies that arrange one or more of said diagnostic factors and/or one or more

of said medical conditions in different hierarchies,” and that the graph database cross-link factors and conditions “that are arranged in different hierarchies in said medical evidence sources.” The specification attributes those taxonomies and hierarchies to pre-existing ontologies and states: “It is realized that multiple clinical ontologies exist, including SNOMED-CT,” and that “SNOMED-CT provides taxonomies of clinical terms (e.g., fever, heart valve replacements, etc.)”; “one taxonomy provides a hierarchy of diagnostic factors and another taxonomy provides a hierarchy of conditions.” *Id.* at 4:13–21. The specification’s criticism of SNOMED-CT is not that it lacks hierarchies but that “[t]hese are hierarchies of concepts, not relationships.” *Id.* at 4:24–25.

95. The specification further confirms that the system uses these existing ontologies as-is: “the process uses a clinical terminology ontology with pre-coordinated and post-coordinated expressions from SNOMED-CT, uses logical observation identifiers names and codes (LOINC) that gives us reasoning for laboratory tests, and uses RxNORM that gives us reasoning for pharmaceuticals.” *Id.* at 5:14–19. The elemental diagnostic factors “may also be connected to other ontologies, including but not limited to SNOMED-CT, for example to provide standardized synonyms,” and each is “linked to a taxonomy of clinical concepts (e.g., SNOMED) in order to perform inferential reasoning.” *Id.* at 10:46–48, 11:3–6. The reasoning service infers factors “using the SNOMED ontology polyhierarchy

and walking the relevant ancestor-descendant chains.” *Id.* at 8:50–55. In my opinion, the specification thus identifies the “different taxonomies” and “different hierarchies” recited in the claims with third-party ontologies that, as I explained above, were in widespread use for decades before the priority date. *Supra* § VI.A.

96. The specification does not describe any new database. Instead, it makes clear that “[a]ny of a wide variety of other types of databases may be used to store this information such as a relational database or object store or object-oriented database,” and that “[i]n one particular implementation a JanusGraph graph database is used having nodes and arcs.” *Id.* at 8:13–20. JanusGraph is an off-the-shelf, open-source graph database. Figure 1 likewise depicts graph database 50 as a generic “JanusGraph” store. *Id.* at FIG. 1. The specification describes database 50 interchangeably as “a computational database, a knowledge base, an evidence base, an inferential database, or an inferential database of computational elements.” *Id.* at 8:23–26.



Ex. 1001 at FIG. 1 (combined)

97. The ‘882 patent also describes establishing APIs for the graph database. But the only APIs the specification discusses in any detail are existing industry standards: “EMR APIs 28 is a collection of APIs such as FHIR, USCDI (United States Core Data for Interoperability), CDS Hooks, and vendor-specific APIs which allow EMR agent 26, user interface 20 and other components to communicate with the EMR system.” *Id.* at 6:4–8. The specification likewise acknowledges that patient data “is accessed via various methods including but not limited to FHIR (Fast Healthcare Interoperability Resources) standards,” and that its application is integrated “via relevant standards such as a SMART . . . on FHIR application for EHRs (such as Epic and Cerner).” *Id.* at 5:52–54, 6:44–47. Indeed, the specification concedes that “standards exist enabling interoperable exchange of patient data (e.g., FHIR).” *Id.* at 3:59–60.

B. Prosecution History of the ‘882 patent

98. I have reviewed the prosecution history of the ‘882 patent. Ex. 1002. I understand that the prosecution history is relevant to understanding the scope and meaning of the claims, as well as the representations made by the applicant to obtain the patent.

99. The application that issued as the ‘882 patent was filed on February 4, 2025, as Application No. 19/045,329. Ex. 1002 at p. 146. The application was filed with 30 claims, including 4 independent claims and 26 dependent claims. Ex. 1002

at p. 146. The applicant requested and was granted prioritized examination (Track I) on March 18, 2025. Ex. 1002 at pp. 141–42.

100. The Patent Office issued an Office Action rejecting all 30 pending claims. *See* Ex. 1002 at pp. 93–119. The Examiner rejected claims 1–30 as directed to an abstract idea without significantly more. Ex. 1002 at pp. 93–102. Specifically, the Examiner determined that the claims recited steps that “can be practically performed by a human, either mentally or with the use of pen and paper,” placing them in the “Mental Processes” grouping of abstract ideas. Ex. 1002 at pp. 94–95. The Examiner found that the additional elements amounted to insignificant extra-solution activity and did not integrate the abstract idea into a practical application or provide an inventive concept. Ex. 1002 at pp. 95–96. The same analysis was applied separately to claims 1–11, claims 12–23, and claims 24–30. Ex. 1002 at pp. 93–103.

101. The Examiner also rejected claims 1–7, 10, and 16–23 as anticipated by Iliff (U.S. Patent Application Publication No. 2005/0010444). Ex. 1002 at pp. 107–15. The Examiner rejected claims 24–29 as anticipated by a second Iliff reference (U.S. Patent Application Publication No. 2003/0163299). Ex. 1002 at pp. 115–18.

102. The applicant filed an amendment and response on September 5, 2025. Ex. 1002 at pp. 55–76. The response included claim amendments and arguments against the § 101 and § 102 rejections.

103. The applicant amended the independent claims to add new limitations, most significantly adding to each independent claim the requirement to “establish one or more application programming interfaces (APIs) for said graph database, said one or more APIs allowing an application to integrate with said graph database to access said separate entries that are linked in said graph database in order to generate a healthcare determination.” Ex. 1002 at pp. 57–67. The applicant also added new claims 31–42. Ex. 1002 at pp. 57–67.

104. The applicant argued that the newly added API limitation overcame the § 101 rejection because an API is “a software construct that a human cannot practically perform in his or her mind” and “the API allows the software application to integrate with the graph database to access the entries.” Ex. 1002 at pp. 68–69. The applicant further argued that the claimed system addressed a specific technical problem—integrating diverse medical evidence sources into a computational form—and described “a specific data architecture (similar to the Enfish case) that can integrate and normalize information from different medical evidence sources.” Ex. 1002 at p. 68.

105. Regarding the anticipation rejection, the applicant argued that claim 1 requires receiving text from medical evidence sources and storing diagnostic factors as separate entries in a graph database, whereas Iliff discloses taking a concept from a patient’s EMR and storing it back into the same patient EMR rather than a graph

database of cross-linked entries derived from medical evidence sources. Ex. 1002 at pp. 69–70. The applicant further argued that Iliff did not disclose “said first entry of said first diagnostic factor in said graph database does not include information specific to said first medical condition.” Ex. 1002 at p. 70. The applicant also argued that Iliff did not disclose unique identifiers linking diagnostic factor entries to medical ontologies (claim 4), demographics entries linked between medical conditions and diagnostic factors (claim 7), or traversal of medical ontology links to infer other diagnostic factors (claim 18). Ex. 1002 at pp. 70–72.

106. The Patent Office issued a Notice of Allowance on December 9, 2025. Ex. 1002 at pp. 11–21. The Examiner provided the following reasons for allowing the claims:

- **Claims 1–11, 31, 35, 39:** The Examiner stated that the prior art did not teach or suggest the combination of (a) storing diagnostic factors and medical conditions as cross-linked separate entries in a graph database where the diagnostic factor entry does not include information specific to the medical condition, together with (b) establishing APIs for the graph database allowing an application to integrate with the database to generate a healthcare determination. Ex. 1002 at p. 17.
- **Claims 12, 13, 15, 32, 36, 40:** The Examiner stated that the prior art did not teach or suggest storing diagnostic factors and medical conditions as cross-

linked nodes in a graph database, including a demographics node located between a medical condition node and a diagnostic factor node, together with establishing APIs and matching patient data to display a healthcare determination. Ex. 1002 at pp. 17–18.

- **Claims 16–23, 33, 37, 41:** The Examiner stated that the prior art did not teach or suggest the combination of matching diagnostic factors to entries in a graph database that are cross-linked to medical conditions, where the entries originate in a medical evidence journal providing different taxonomies, together with establishing APIs, traversing links to collect weight values, summing those weights, and displaying linked medical conditions in order of total weight. Ex. 1002 at pp. 18–19.
- **Claims 24–30, 34, 38, 42:** The Examiner stated that the prior art did not teach or suggest identifying unknown diagnostic factor entries linked to medical condition entries that originate in a medical evidence journal providing different taxonomies, together with establishing APIs, identifying a rank for each unknown diagnostic factor stored in a database entry link, and displaying unknown diagnostic factors in order of rank as a healthcare determination. Ex. 1002 at pp. 19–20.

107. The ‘882 patent issued on January 6, 2026 with 41 total claims, including four independent claims (claim 1, claim 15, claim 21, and claim 32). *See* Ex. 1001.

VIII. CLAIM CONSTRUCTION

108. I have been instructed that, unless otherwise noted herein, the Challenged Claims of the ‘882 patent are to be given their ordinary and customary meaning as understood by a POSITA at the time of the alleged invention in light of the specification and prosecution history. I have thus applied the plain and ordinary meaning of the claim terms in my analysis below. I reserve the right to supplement my opinions to the extent that one or both parties later propose claim constructions for any claim terms in the ‘882 patent.

IX. OVERVIEW OF PRIOR ART REFERENCES

A. Datla (Ex. 1005)

109. Datla discloses an NLP engine that receives natural language input, extracts patient symptoms and queries a knowledge graph to generate ranked diagnoses. Datla’s system includes a “natural language processing engine configured to extract the at least one patient symptom and at least one demographic parameter from the received natural language input.” Ex. 1005, claim 1. Datla’s system “may” map topical keywords extracted from patient “descriptions, summaries or notes” to “categories represented in” ontologies such as “SNOMED CT for diagnoses.” Ex.

1005 at ¶ [0054]. The knowledge graph has “symptoms as root nodes connected by edges to the diseases and medical conditions associated with those symptoms,” and the system generates a diagnosis graph as a subset of the knowledge graph via an activation-decay process that produces “a ranked list of one or more medical conditions, diagnoses, treatments, and/or tests for the patient.” Ex. 1005 at ¶¶ [0006]–[0008], [0028], [0036], [0050]–[0057]; claim 1.

B. Goldberg (Ex. 1006)

110. Goldberg discloses a hybrid knowledge graph diagnostic system that integrates clinical data with ontology-based knowledge graphs, including SNOMED and LOINC. Goldberg’s system includes “a healthcare knowledge graph database for storing a medical hybrid knowledge graph, the hybrid knowledge graph including healthcare data from one or more ontology-based knowledge graphs and clinical data from one [sic] more sources; a data processing engine running a diagnostic algorithm, designed to process a patient condition data and to process a hybrid knowledge graph database, and select one or more pathways on the knowledge graph to generates [sic] one or more differential diagnoses for suspected disorders.” Ex. 1006, Abstract. Goldberg’s hybrid knowledge graph includes six distinct node sets: (1) diseases, (2) associated conditions, (3) symptoms, (4) complications, (5) laboratory tests, and (6) treatments. *Id.* at 16:8–24. The system integrates SNOMED’s “is-a” hierarchies and LOINC for laboratory test ontologies, and its

diagnostic algorithm selects pathways to generate differential diagnoses. *Id.* at 5:47–60, 6:5–28, 7:25–50, 15:40–65, 16:10–21.

C. Gnanasambandam (Ex. 1007)

111. Gnanasambandam discloses an API deployment platform for healthcare knowledge graphs. The system “publish[es] . . . at least one API that uses artificial intelligence to perform analytics based on a knowledge graph pertaining to ontological data of a medical condition.” Ex. 1007 at 1:32–42; claim 1. The platform makes these APIs available to client devices running software applications, which subscribe to and invoke them. *Id.* at 100:24–34, 101:3–24, 102:22–46. Gnanasambandam’s knowledge graphs and logical structures “may be generated and updated continuously or on a periodic basis by an artificial intelligence engine with evidence-based guidelines, physician research, patient notes in EMRs, physician feedback, and so forth.” *Id.* at 8:24–29. The key disclosures of Gnanasambandam include its API architecture for medical knowledge graph access, AI-based analytics, and programmatic access by external software applications.

D. Dang (Ex. 1008)

112. Dang discloses a graph-based clinical diagnosis system employing a three-tier knowledge graph architecture. Dang’s FIG. 1A meta-model knowledge graph expressly includes multiple node types, including “Lab Test & Check” and “Observation Item” as dedicated node types alongside Disease, Symptom, Sign, and

other node types. Ex. 1008 at ¶ [0027]; FIG. 1A. FIG. 1B discloses a three-tier operational graph with factual nodes (including “symptom nodes {s1, s2, s3 . . . sm}; context nodes {c1 . . . cp}, and test nodes {t1, t2 . . . tj}”), feature nodes {f1, f2 . . . fx}, and disease nodes {d1, d2, d3 . . . dn}. Ex. 1008 at ¶¶ [0027]–[0031]; FIG. 1A, 1B. The graph includes conditional probabilities $P(f|d)$ on the edges between feature nodes and disease nodes and supports iterative next-inquiry generation for clinical diagnosis. Ex. 1008 at ¶¶ [0028]–[0031]; FIG. 1B.

E. Zhong (Ex. 1009)

113. Zhong discloses a system for “question generation to obtain more related medical information based on observed symptoms from a patient.” Ex. 1009 at Abstract. Zhong’s system implements an iterative pipeline: observed symptoms are collected, possible diseases are identified from a knowledge graph, candidate symptoms (those not yet observed) are identified and scored, and the highest-ranked candidates are presented to the clinician as the next actions to take. *Id.* at 3:39–4:31; FIGS. 3–6. The process repeats “until a disease is diagnosed or the patient reports no more observed symptoms.” *Id.* at 4:27–31.

114. Zhong’s Figures 3 through 6 describe a single, nested, iterative pipeline. Figure 3 is the master-level flow defining the complete diagnostic loop. Figures 4, 5, and 6 each elaborate on specific steps within Figure 3: Figure 4 expands on candidate-symptom generation and initial probability-based selection (Steps 310–

315 of Figure 3); Figure 5 expands on discriminative scoring and refined selection (Step 330 of Figure 3); and Figure 6 expands on conflict filtering, question generation, and iterative updating (Steps 335–345 of Figure 3).

F. Van Assel (Ex. 1011)

115. Van Assel discloses a computer-implemented medical diagnosis system. Its diagnostic engine “may be connected to a knowledge graph 150, also referred to as a knowledge base, the knowledge graph being a large structured medical knowledge base, or medical ontology, linking together medical diseases, symptoms, risk factors etc.” Ex. 1011 at ¶ [0115]. The diagnostic engine “is configured to compute probabilities that diseases are present based on evidence (e.g. the presence of one or more symptoms, diseases and/or risk factors) provided, and derive a diagnosis from these probabilities.” *Id.* at ¶ [0113].

116. Van Assel teaches that each item of medical data carries an indication of whether it is present, absent, or unknown. The evidence used for diagnosis “may further comprise the absence of items of medical data, such as symptoms, diseases, risk factors, physiological data, recommendations and/or behaviours, identified as absent by the patient in the input.” Ex. 1011 at ¶ [0112]. Van Assel also teaches recording when each item was reported. The clinical history “may include temporal information about concepts—that is, when were they were [sic] reported.” *Id.* at ¶ [0263].

G. Kim (Ex. 1012)

117. Kim discloses a method and computer system “for supporting data-based clinical decision-making” in which “[m]edical concepts and relations between medical concepts contained in a structured medical report and/or a template for such reports are extracted” and “then integrated into a graph database.” Ex. 1012, Abstract; *see also id.*, Title (54) (“Data-Based Clinical Decision-Making Utilising Knowledge Graph”). Kim explains that “[a]n example of a graph database is a so-called knowledge graph, in which the concepts and their relations between each other represent the knowledge of a specific knowledge domain, e.g. medicine.” *Id.* at ¶ [0038].

118. Kim teaches annotating the entries of its graph database with unique identifiers from standard medical ontologies. Kim states that “annotating the entries of the graph database using ontology-identifiers allows to recognize semantically equivalent concepts and relations” and that “the entries of the graph database may themselves be annotated with identifiers from several different ontologies.” Ex. 1012 at ¶¶ [0050]–[0051]. Kim gives a SNOMED-CT example: “the concept ‘Lung cancer’ can be represented through the RadLex code RID45686 or the SNOMED CT code SCTID: 93880001.” *Id.* at ¶ [0051]. Kim also teaches that “the graph database includes synonyms, translations, concepts and relationships.” *Id.* at ¶ [0054].

X. INVALIDITY ANALYSIS OF THE CHALLENGED CLAIMS

119. In this section, I provide my opinions regarding why the challenged claims of the ‘882 patent are unpatentable as obvious under 35 U.S.C. § 103 and for being directed to ineligible subject matter under 35 U.S.C. § 101. In forming these opinions, I have considered the ‘882 patent (Ex. 1001), its prosecution history (Ex. 1002), the prior art references identified below, and all other materials listed in Section V. I have also relied on my own knowledge, training, and decades of experience in the fields of biomedical informatics, knowledge representation, medical ontologies, knowledge graphs, and clinical decision support systems.

A. Ground 1: Claims 1-3, 8-9, and 11-14 are Obvious Over Datla, Goldberg, and Gnanasambandam

120. In my opinion, claims 1-3, 8-9, and 11-14 of the ‘882 patent are obvious over Datla, Goldberg, and Gnanasambandam.

1. Motivation to Combine

121. A POSITA would have been motivated to combine Datla and Goldberg because both concern medical knowledge graphs used for clinical decision support. Goldberg teaches integrating clinical knowledge with ontology-based knowledge graphs such as SNOMED and LOINC, while Datla builds a literature-derived

knowledge graph linking symptoms to conditions. Ex. 1006 at 5:51–58; Ex. 1005 at ¶¶ [0031], [0036].

122. Datla teaches that its system “may” map topical keywords extracted from patient “descriptions, summaries or notes” to “categories represented in” controlled clinical ontologies, including “SNOMED CT for diagnoses, LOINC for tests, and/or RxNorm for treatments.” Ex. 1005 at ¶ [0054]. Datla also states that the corpus can be “any source of information.” Ex. 1005 at ¶ [0030]. A POSITA would recognize that constructing the graph from those ontologies would ensure that the concepts to which Datla may map a patient scenario correspond directly to standardized terms in the knowledge graph and improve the reliability of a user’s query. Ex. 1005 at ¶ [0055].

123. Goldberg explains that ontologies such as SNOMED capture “health terms and their semantic relations,” but that “[c]linical relations were out of scope for such Ontologies,” so that associations between diagnostic factors and medical conditions are not included; Datla’s literature-derived knowledge graph supplies clinical relations between symptoms and conditions. Ex. 1006 at 1:54–2:6; Ex. 1005 at ¶¶ [0031], [0036]. The references are thus complementary, and SNOMED’s coverage of “hundreds of thousands of medical terms” would improve the completeness of Datla’s knowledge graph. Ex. 1006 at 1:54–56.

124. A POSITA would have had a reasonable expectation of success because Datla discloses that its corpus can be “any source of information” and that graph edges can be added according to hierarchical relations, while Goldberg demonstrates deriving graph content from SNOMED and LOINC and linking it with literature-derived clinical relations. Ex. 1005 at ¶ [0030], claim 16; Ex. 1006 at 10:4–23. The combination applies a known technique to a known system to yield a predictable graph constructed from medical evidence sources that provide different taxonomies.

125. A POSITA would have been motivated to combine Datla and Gnanasambandam because they operate in the field of medical knowledge-graph-based decision support. Gnanasambandam builds knowledge graphs representing knowledge of diseases and uses them for analytics, diagnoses, and healthcare determinations, while Datla accepts natural-language input from a medical professional and provides possible diagnoses, tests, and treatments. Ex. 1007 at 51:27–42, 55:50–58, 100:24–47; Ex. 1005 at ¶ [0028]. Publishing APIs would make Datla’s knowledge graph accessible to external applications in the deployment pattern taught by Gnanasambandam. Ex. 1007 at 100:24–47; 102:22–46.

126. Adding an API layer to a knowledge graph-based diagnostic system involves applying a known technique (API publication, as taught by Gnanasambandam) to a known system (Datla’s knowledge graph) to yield the predictable result of an API-accessible medical knowledge graph system that

generates healthcare determinations in response to application queries. A POSITA would have had a reasonable expectation of success because API publication for knowledge-based systems was well-established in the art, and no technical barrier would have impeded the combination.

2. Independent Claim 1

127. Each element of independent claim 1 of the ‘882 patent is disclosed or rendered obvious by the combination of Datla, Goldberg, and Gnanasambandam. I address each element below.

a. Element [1pre]

128. Element [1pre] recites: “A computer system for integrating medical evidence text into a computational form, said computer system comprising.”

129. To the extent the preamble is limiting, Datla discloses a computer system for integrating medical evidence text into a computational form. Datla is directed to “automated methods and systems to provide a clinical diagnosis of a patient’s symptoms based on a corpus of medical knowledge.” Ex. 1005 at ¶ [0001]. Datla’s system “accepts natural language input from a medical professional about a patient’s scenario, processes the input, and provides one or more possible diagnoses, tests, and/or treatments.” Ex. 1005 at ¶ [0028]. The system integrates medical evidence text into a computational form: a knowledge graph with nodes and

weighted edges representing clinical concepts and their relationships. Ex. 1005 at ¶¶ [0030]–[0031].

b. Element [1a]

130. Element [1a] recites: “one or more processors; and memory that stores instructions, wherein the instructions, when executed by the one or more processors, cause the one or more processors to.”

131. Datla discloses a computer system with one or more processors and memory storing instructions. For example, FIG. 5 illustrates system 500 including “Processor 530” and “Memory 540” components.

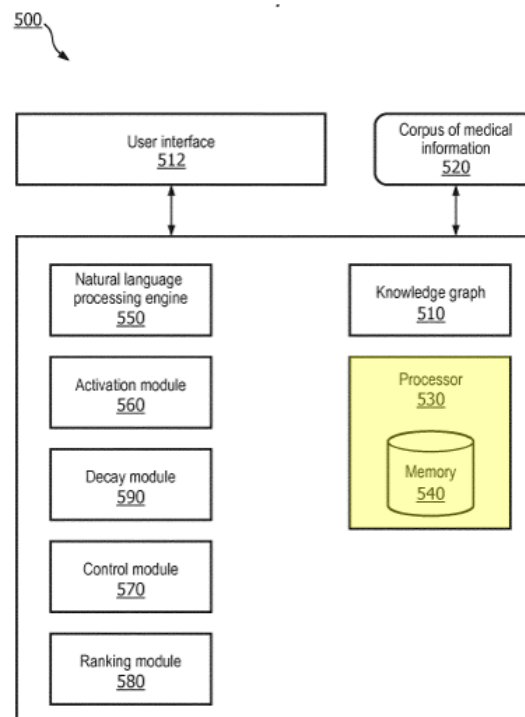


FIG. 5

Ex. 1005 at FIG. 5 (annotated)

132. Datla discloses that “system 500 comprises a processor which performs one or more steps of the method Processor 530 may be formed of one or multiple modules, and can comprise, for example, a memory 540. Processor 530 may take any suitable form, including but not limited to a microcontroller, multiple microcontrollers, circuitry, a single processor, or plural processors. Memory 540 can take any suitable form, including a non-volatile memory and/or RAM.” Ex. 1005 at ¶ [0079]; *see also id.* at ¶ [0019] (“a processor or controller may be associated with one or more storage media (generically referred to herein as ‘memory,’ e.g., volatile and non-volatile computer memory such as RAM, PROM, EPROM, and EEPROM, floppy disks, compact disks, optical disks, magnetic tape, etc.). In some implementations, the storage media may be encoded with one or more programs that, when executed on one or more processors and/or controllers, perform at least some of the functions discussed herein.”).

133. As set forth below, in the combination, the memory stores instructions that when executed by the one or more processors, cause the one or more processors to perform the functionality recited in elements [1b]–[1h].

c. Element [1b]

134. Element [1b] recites: “receive electronic text in a plurality of medical evidence sources, said electronic text pertaining to a plurality of medical conditions,

wherein at least a subset of said medical evidence sources includes a published journal.”

135. Datla’s system receives electronic text by collecting and ingesting content from its corpus of medical information, which includes multiple source categories: “The corpus of medical information can be any source of information, including but not limited to medical journals, newspapers, online sources such as Wikipedia, and other sources.” Ex. 1005 at ¶ [0030] (emphasis added). These sources contain electronic text pertaining to a plurality of medical conditions. For example, Datla processes clinical medicine articles covering a wide range of conditions: “The first section of such a Wiki page normally contains an introduction to the diagnosis. Among several other sections inside the Wiki page, the ‘Signs and symptoms’ section describes the classic and common signs and symptoms for the diagnosis.” Ex. 1005 at ¶ [0040].

136. Datla’s system receives the electronic text from the corpus of medical information (e.g., medical journals) as input to its knowledge graph construction pipeline: “[T]he system utilizes a graph-building approach that centers on three steps: (i) topical keyword analysis in which the most clinically relevant keywords from the given topic descriptions, summaries, and clinical notes are identified” Ex. 1005 at ¶ [0034]. The system thus receives and processes the electronic text of the medical evidence sources to build its knowledge graph. Datla’s knowledge graph

links symptoms “to hundreds of other nodes,” reflecting the breadth of conditions covered. Ex. 1005 at ¶ [0031].

137. Datla’s construction of the knowledge graph is shown as step 112 in the flow diagram of FIG. 1.

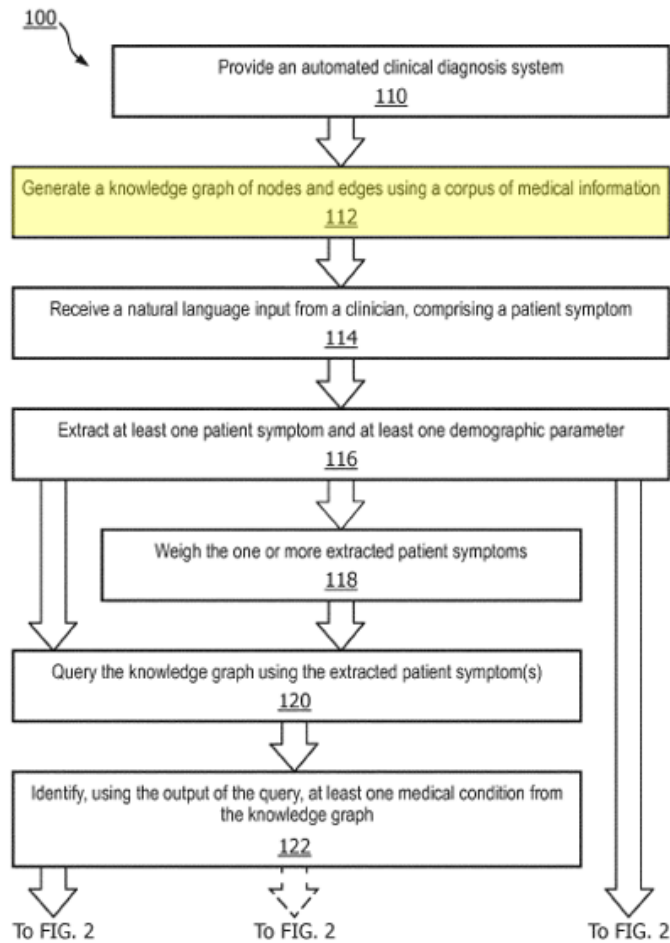


FIG. 1

Ex. 1005 at FIG. 1 (annotated)

138. FIG. 3 of Datla depicts a knowledge graph 310.

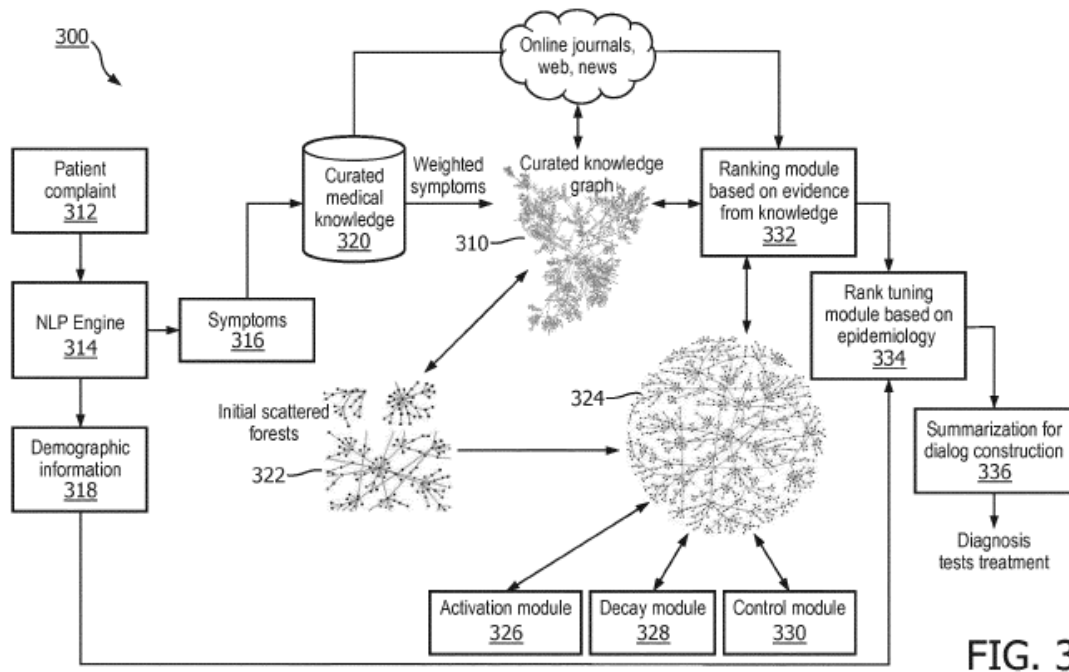


FIG. 3

Ex. 1005 at FIG. 3

139. Datla discloses that at least a subset of its medical evidence sources includes a published journal. For example, Datla expressly identifies “medical journals” as a source category for its corpus. Ex. 1005 at ¶ [0030]. A POSITA would understand that Datla’s reference to “medical journals” would include medical journals such as, for example, The BMJ. Indeed, the ‘882 patent acknowledges The BMJ as a known medical journal existing prior to the ‘882 patent’s priority date. Ex. 1001 at 1:59–62, 7:6–19. Datla also processes abstracts of medical journal articles from over 1.25M PubMed Central articles, which Datla describes as a “freely available online database of full-text biomedical articles.” Ex. 1005 at ¶¶ [0047]–[0048]. A POSITA would understand that PubMed is an index of published medical journal articles, and articles indexed therein include published journal articles. I

incorporate by reference my discussion of medical journals in the Technology Background section above.

d. Element [1c]

140. Element [1c] recites: “retrieve a plurality of diagnostic factors that are associated with one or more of said plurality of medical conditions.”

141. Datla discloses retrieving a plurality of diagnostic factors associated with one or more medical conditions. Datla’s system processes medical evidence source text to extract symptoms, signs, diagnoses, tests, and treatments, each associated with one or more medical conditions.

142. For example, Datla discloses: “The first section of such a Wiki page normally contains an introduction to the diagnosis. Among several other sections inside the Wiki page, the ‘Signs and symptoms’ section describes the classic and common signs and symptoms for the diagnosis. Each collected Wikipedia page is turned into a key–value pair by using the following principle: the free text from the first section and the sections for sign and symptoms is the key and the title of the page is the value.” Ex. 1005 at ¶ [0040]. In this example, the page title (a medical condition) is the value; the extracted signs and symptoms (diagnostic factors) are the keys. Datla further processes this text to “generate the diagnoses, tests, and treatments using the underlying clinical contexts.” Ex. 1005 at ¶ [0034].

143. A POSITA would understand that the same or similar process described for Wiki pages would be employed for the other sources of information described in Datla, including medical journals. Ex. 1005 at ¶ [0030].

e. Element [1d]

144. Element [1d] recites: “wherein said medical evidence sources provide different taxonomies that arrange one or more of said diagnostic factors and/or one or more of said medical conditions in different hierarchies, and wherein a first diagnostic factor is discussed in part of said electronic text as being indicative of a first medical condition.”

145. This claim element requires that “said medical evidence sources provide different taxonomies.” The grammatical subject “medical evidence sources” refers to the set of sources from which the system receives electronic text, not a single source in isolation.

146. The specification of the ‘882 patent provides examples of different taxonomies that arrange one or more of said diagnostic factors and/or one or more of said medical conditions in different hierarchies. For example, the specification references SNOMED (which I describe above in the Technology Background section): “SNOMED-CT provides taxonomies of clinical terms (e.g., fever, heart valve replacements, etc.) . . . one taxonomy provides a hierarchy of diagnostic factors and another taxonomy provides hierarchy of conditions.” Ex. 1001 at 4:17–21. The

specification also says “[t]hese are hierarchies of concepts, not relationships.” *Id.* at 4:24–27. The specification repeatedly refers to SNOMED as a taxonomy. *See, e.g., id.* at 10:50 (“a taxonomy such as SNOMED-CT”), 11:4–5 (“a taxonomy of clinical concepts (e.g., SNOMED)”), 15:61 (“a taxonomy such as SNOMED”), 16:17–18 (“an established taxonomy of clinical concepts such as SNOMED-CT”), 17:24–25 (“the international taxonomy SNOMED-CT”), 20:33–34 (“a medical taxonomy (e.g., SNOMED, SNOMED-X, LOINC, RxNorm, ICD, CPT, etc.)”).

147. **Datla Alone:** Datla discloses and renders obvious that the medical evidence sources provide different taxonomies arranging diagnostic factors and/or medical conditions in different hierarchies. As discussed above, Datla meets the portion of element [1d] that recites that a first diagnostic factor is discussed in part of said electronic text as being indicative of a first medical condition. Since Datla is extracting relationships between symptoms and conditions from the text, it is clear that the text discusses that the symptom (diagnostic factor) is indicative of the medical condition. *See, e.g.,* Ex. 1005 at ¶¶ [0031]–[0032], [0040].

148. As discussed above, Datla constructs its knowledge graph from a corpus of medical information that “can be any source of information, including but not limited to medical journals, newspapers, online sources such as Wikipedia, and other sources.” Ex. 1005 at ¶ [0030]. Datla discloses sources that are themselves organized hierarchically: “when utilizing an online source *with a hierarchy* and a

master top category (such as ‘clinical medicine’), all pages including and under the master top category are analyzed and the *hierarchy* of the sub-categorized pages is maintained, with information being extracted from each page.” *Id.* (emphasis added). A hierarchy of categories and sub-categories under a master top category is a classification of concepts arranged from general to specific.

149. Datla retains these hierarchical relationships when constructing the knowledge graph. Datla describes generating the knowledge graph by “adding a first node representing a first clinical concept, a second node representing a second clinical concept, and an edge between the first node and the second node according to a hierarchical relation of the corpus of medical information.” Ex. 1005 at claim 16. Datla likewise explains that some embodiments “use the Wiki pages under the clinical medicine category to build a knowledge graph,” and that “[t]he hierarchy of each Wiki page is preserved to encode its distinguishing characteristics with respect to other pages.” Ex. 1005 at ¶ [0035].

150. The concepts classified within Datla’s hierarchically organized source include medical conditions and diagnostic factors. Datla builds a graph in which “each node is a medical condition, diagnosis, test, procedure, medication or any other clinical concept.” Ex. 1005 at ¶ [0035]. For example, Datla uses “Wikipedia clinical medicine category pages to build a directed knowledge graph 310, which possesses symptoms as root nodes connected by edges to the diseases and medical

conditions associated with those symptoms.” Ex. 1005 at ¶ [0036]. Datla explains that “the title of each Wikipedia page is the name of the diagnosis described by the page.” Ex. 1005 at ¶ [0040].

151. A POSITA would have understood that the “clinical medicine” category hierarchy that Datla describes classifies diseases and medical conditions in one branch and signs and symptoms (i.e., diagnostic factors) in a separate branch. These two branches are different taxonomies: one arranges diagnostic factors in a hierarchy, and the other arranges medical conditions in a different hierarchy. This is the same structure the ‘882 patent attributes to SNOMED-CT. Because Datla maintains “the hierarchy of the sub-categorized pages” of such a source, Datla’s medical evidence sources provide different taxonomies arranging diagnostic factors and medical conditions in different hierarchies. Ex. 1005 at ¶ [0030].

152. Additionally, Datla’s corpus is not limited to a single source. Datla identifies its “[c]urated knowledge source” as including “Wikipedia, medical textbooks, biomedical literature.” Ex. 1005 at FIG. 4. Datla also recognizes that “[t]he information on one page or from one source may be inherently related to other pages, sources, or medical conditions.” Ex. 1005 at ¶ [0030]. A POSITA would have understood that medical textbooks classify medical conditions according to organizing principles. Such a classification is a hierarchy of conditions different from the category hierarchy of an online source such as Wikipedia. A POSITA

would therefore have understood that Datla’s medical evidence sources, taken together, provide different taxonomies. At a minimum, it would have been obvious to a POSITA to construct Datla’s knowledge graph from a corpus that includes such differently organized sources. Ex. 1005 at ¶ [0030], FIG. 4.

153. Datla discloses that its system may “map [topical keywords] to categories represented in one or more ontologies, including but not limited to the following controlled clinical ontologies: SNOMED CT for diagnoses, LOINC for tests, and/or RxNorm for treatments.” Ex. 1005 at ¶ [0054]. The ‘882 patent itself identifies SNOMED-CT, LOINC, and RxNorm as medical taxonomies. Ex. 1001 at 20:33–34.

154. Datla discloses that the corpus used to build the knowledge graph “can be any source of information” and “is not limited to” the listed examples. Ex. 1005 at ¶ [0030]. A POSITA would understand, or at a minimum find it obvious, to use SNOMED, LOINC, and RxNorm as sources within the corpus used to build the knowledge graph. A POSITA would have been motivated to include these ontologies among the sources from which the knowledge graph is built to ensure that the ontology categories to which Datla may map a patient scenario correspond to nodes in the knowledge graph. A POSITA would have had a reasonable expectation of success because Datla already discloses adding graph edges “according to a hierarchical relation of the corpus of medical information.” Ex. 1005 at claim 16. In

such a system, the medical evidence sources would provide different taxonomies: SNOMED CT arranging diagnoses and clinical findings, LOINC arranging laboratory tests, and RxNorm arranging treatments.

155. **Datla and Goldberg:** To the extent Datla alone does not disclose or render obvious element [1d], this element would have been obvious over Datla in view of Goldberg.

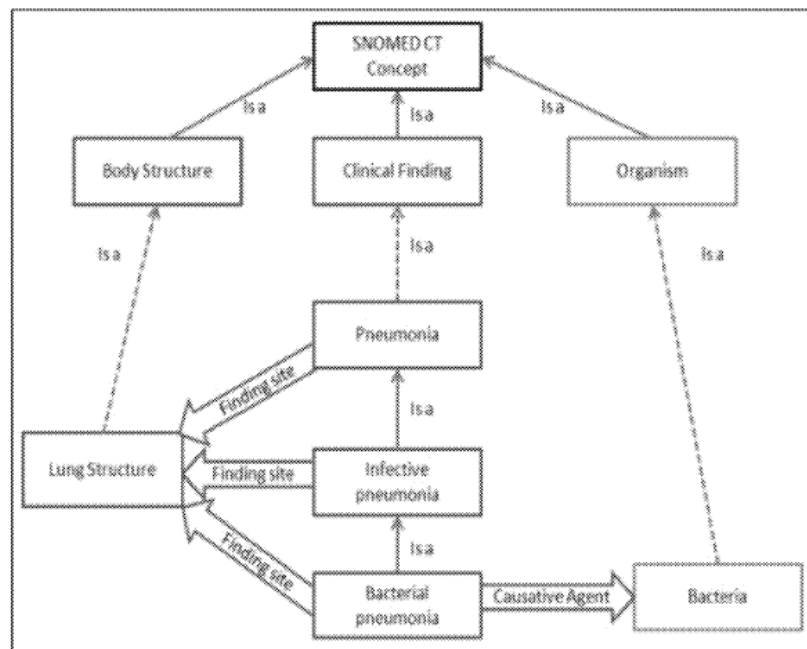
156. Goldberg discloses constructing a diagnostic knowledge graph from SNOMED and LOINC. Goldberg discloses a platform that enables “the translation of medical knowledge from multiple sources (e.g., books, articles, publications, personal knowledge etc.), into a hybrid knowledge graph which integrates clinical knowledge along with other ontology based knowledge graphs, such as SNOMED, LOINC or other types of medical databases.” Ex. 1006 at 5:51–58.

157. Goldberg describes the construction of that graph from these specific sources: “a first knowledge graph is derived from a healthcare Ontology which includes a large vocabulary of medical concepts. (e.g., SNOMED)”; “[a] second knowledge graph is derived from a different source such as an ontology dedicated to laboratory tests and procedures (e.g., LOINC)”; and “[a] third knowledge graph is created or generated by medical experts or healthcare professionals, who research medical publications to build a clinical probability graph that describes clinical

statistical information that links medical concepts that are described in the Ontological graphs.” Ex. 1006 at 6:12–24.

158. Goldberg describes ontologies (which include SNOMED) as “covering hundreds of thousands of medical terms arranged as an ontology,” which holds “names of findings and disorders that can be grouped in families.” Ex. 1006 at 1:54–2:1. Goldberg illustrates this hierarchical structure with “a SNOMED concept tree, described in a knowledge graph format.” Ex. 1006 at 9:50–52, 4:63–65, FIG. 6.

FIG. 6



Ex. 1006 at FIG. 6

159. Goldberg carries the hierarchy into its hybrid knowledge graph. For example, in Goldberg’s hybrid graphs, “darker arrows represent SNOMED ‘type-of’ relations” and “[l]ighter arrows reflect clinical relations taken from medical

literature.” Ex. 1006 at 10:4–23, FIGS. 9–10. Goldberg further explains that its model “may also take into account sub-type relations which are part of the ontology (e.g. body structures, or subtype of diseases such as iron-deficiency-anemia being a sub-type of anemia).” Ex. 1006 at 10:41–46.

160. In Goldberg, SNOMED provides different taxonomies arranging diagnostic factors (findings) and medical conditions (disorders) in different hierarchies, exactly as the ‘882 patent describes. LOINC provides a further, different taxonomy arranging laboratory tests. A knowledge graph constructed from medical evidence sources that include SNOMED and LOINC, as in Goldberg, is constructed from “medical evidence sources [that] provide different taxonomies that arrange one or more of said diagnostic factors and/or one or more of said medical conditions in different hierarchies.”

161. A POSITA would have been motivated to include SNOMED CT and LOINC among the sources from which Datla’s knowledge graph is constructed, as Goldberg teaches, for at least the following reasons.

162. First, Datla teaches that its system “may” map topical keywords to “categories represented in” controlled clinical ontologies, including “SNOMED CT for diagnoses, LOINC for tests, and/or RxNorm for treatments,” and broadly states that the corpus used to construct the knowledge graph can be “any source of information.” Ex. 1005 at ¶¶ [0030], [0054]. A POSITA would have recognized that

constructing the knowledge graph from those ontologies would ensure that the concepts to which Datla may map a patient scenario correspond directly to nodes in the knowledge graph. This would improve the reliability of Datla's query of the knowledge graph. Ex. 1005 at ¶ [0055].

163. Second, Goldberg explains why a POSITA would combine ontology sources with literature-derived clinical knowledge. Goldberg teaches that ontologies such as SNOMED capture “health terms and their semantic relations,” but that “[c]linical relations were out of scope for such Ontologies”; for example, “the ontology describes chest pain and MI separately, but it does not describe the relation between the two as would appear in a medical article dealing with MI.” Ex. 1006 at 1:54–2:6. Goldberg identifies the advantage of “a computerized means that integrates medical term ontologies as described above along with operational databases, such as costs of lab tests, medication side effects and traditional medical knowledge, to improve healthcare.” Ex. 1006 at 2:24–30. Datla's literature-derived knowledge graph supplies clinical relations between symptoms and conditions. Ex. 1005 at ¶¶ [0031], [0036]. SNOMED CT and LOINC supply the standardized concepts and hierarchies that such literature-derived knowledge lacks. The references are thus complementary.

164. Contemporaneous literature confirms that a POSITA constructing a knowledge graph from biomedical literature would have looked to ontologies for the

concepts and hierarchies used to build the graph. For example, Xia constructed a “biomedical knowledge graph” from “15,970,134 MEDLINE/PubMed citation records” using ontologies because they “provide[] synonyms for concepts, thus facilitating extracting and combining concepts from texts” and “provide[] disease hierarchy information to manage disease of different levels and to enable automatic tracing of a disease to its ancestors for accurate counting.” Ex. 1010 at 1118–19. Xia described “leveraging the hierarchy information of concepts” as an improvement over “previous disease-symptom relation mining work,” *id.* at 1118, and recognized that studies of knowledge acquisition from text used ontologies including “SNOMED-CT for their wide application,” *id.* at 1123.

165. Third, SNOMED contributes breadth of coverage. Goldberg describes SNOMED as “covering hundreds of thousands of medical terms.” Ex. 1006 at 1:54–56. A POSITA seeking to improve the completeness of Datla’s knowledge graph would have been motivated to draw on such a comprehensive source.

166. A POSITA would have had a reasonable expectation of success in making this combination. Datla’s corpus “can be any source of information.” Ex. 1005 at ¶ [0030]. Datla already discloses generating the knowledge graph by adding an edge between nodes “according to a hierarchical relation of the corpus of medical information.” Ex. 1005 at claim 16. The “type-of” relations of SNOMED are hierarchical relations of exactly this kind. Goldberg demonstrates that deriving

knowledge graph content from SNOMED and LOINC, and linking it with literature-derived clinical relations, was a known and workable technique. Ex. 1006 at 10:4–23. The combination applies a known technique (Goldberg’s use of SNOMED and LOINC as knowledge graph sources) to a known system (Datla’s knowledge graph built from a medical corpus) to yield the predictable result of a knowledge graph constructed from medical evidence sources that provide different taxonomies.

f. Element [1e]

167. Element [1e] recites: “create a graph database cross-linking said one or more diagnostic factors and said one or more medical conditions that are arranged in different hierarchies in said medical evidence sources.”

168. Datla discloses creating a graph database cross-linking diagnostic factors and medical conditions. Datla’s knowledge graph cross-links diagnostic factors and medical conditions through weighted edges: “According to an embodiment, the knowledge graph 310 is a tree-like structure with a plurality of nodes connected by one or more edges. Each root node of the graph is a symptom, and the remaining nodes are conditions, diagnoses, tests, procedures, medications, or other clinical concepts. An edge is a relationship between two nodes.” Ex. 1005 at ¶ [0031]. The same symptom node links to multiple condition nodes through weighted edges: “a symptom of a fever will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios.” Ex. 1005 at ¶

[0031]. This constitutes a graph database cross-linking diagnostic factors and medical conditions.

169. Datla further discloses that the system depicted in FIG. 5, which contains knowledge graph 510, includes a database. Ex. 1005 at ¶ [0078] (“a system 500 for automated clinical diagnosis” that “can comprise any of the elements, engines, database, processors, and/or other components described or otherwise envisioned herein.”). System 500 comprises knowledge graph 510 “which is generated as described or otherwise envisioned herein from a corpus of medical information 520.” *Id.* A POSITA would understand that creating the knowledge graph within system 500 would involve creating a graph database to store the knowledge base. Because system 500 expressly includes both a knowledge graph and a database among its components, and because the knowledge graph is the data structure that cross-links diagnostic factors and medical conditions through nodes and weighted edges, a POSITA would recognize that Datla’s knowledge graph constitutes a graph database.

g. Element [1f]

170. Element [1f] recites: “store said one or more diagnostic factors and said one or more medical conditions that are cross linked as separate entries in said graph database, wherein a first entry of said first diagnostic factor indicative of said first

medical condition is linked to a second entry of said first medical condition in said graph database.”

171. Datla discloses storing diagnostic factors and medical conditions as separate entries in the graph database. In Datla’s knowledge graph, each diagnostic factor (symptom, sign, test, etc.) and each medical condition is stored as its own separate node, with edges representing the cross-links between them. “Each root node of the graph is a symptom, and the remaining nodes are conditions, diagnoses, tests, procedures, medications, or other clinical concepts. An edge is a relationship between two nodes.” Ex. 1005 at ¶ [0031]. Each symptom node is a separate entry. Each condition node is a separate entry. The weighted edges between them constitute cross-linking.

172. Datla discloses that a first entry of said first diagnostic factor indicative of said first medical condition is linked to a second entry of said first medical condition in said graph database. In Datla’s knowledge graph, a first entry (e.g., a symptom node representing a first diagnostic factor (e.g., fever) that is indicative of a first medical condition) is linked through an edge to a second entry (e.g., a condition node representing that first medical condition). “[A] symptom of a fever will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios.” Ex. 1005 at ¶ [0031]. The fever symptom node (first entry)

is linked through edges to condition nodes (second entries) representing the medical conditions of which fever is a symptom.

h. Element [1g]

173. Element [1g] recites: “wherein said first entry of said first diagnostic factor in said graph database does not include information specific to said first medical condition represented by said second entry.”

174. Datla discloses this negative limitation. In Datla’s knowledge graph, each diagnostic factor node is condition-independent: the node represents the diagnostic factor itself, not its relationship to a particular condition. The condition-specific relationship information (i.e., the edge weight reflecting how strongly the diagnostic factor is associated with a particular condition) resides on the edges, not in the diagnostic factor node itself. Ex. 1005 at ¶ [0032]. Datla is clear that the same symptom node can link to multiple conditions through different weighted edges. Ex. 1005 at ¶ [0031] (“a symptom of a fever will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios”). This confirms that the symptom node does not include information specific to any one medical condition. Datla’s architecture reflects a standard design principle in knowledge graph engineering.

175. Modern knowledge graphs tend to be based on triple-store architectures (where everything is a node-relation-node grouping) or labeled property graphs

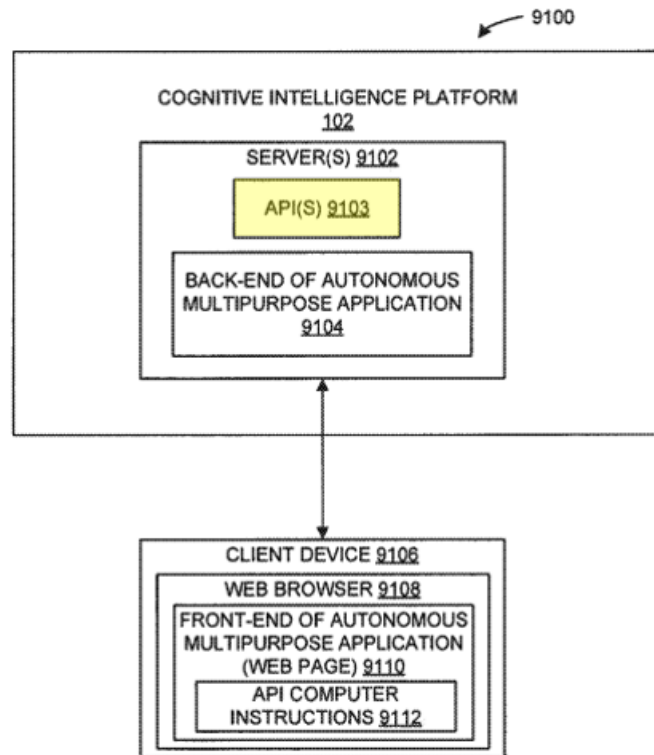
(where the edges themselves may take on properties, as in Datla). I discussed INTERNIST-1 in the Technology Background section and how a POSITA would be familiar with INTERNIST-1. Since the advent of INTERNIST-1 decades ago, a goal in knowledge representation has been to remove unnecessary dependencies between entries in the knowledge base to facilitate flexibility and the ability to perform inferences. A POSITA would understand that this was a routine design principle in knowledge engineering. Based on these well-known design principles, a POSITA would understand, or at a minimum find obvious, that the diagnostic factor nodes in Datla's knowledge graph would not include information specific to a medical condition.

i. Element [1h]

176. Element [1h] recites: “establish one or more application programming interfaces (APIs) for said graph database, said one or more APIs allowing an application to integrate with said graph database to access said separate entries that are linked in said graph database in order to generate a healthcare determination.”

177. Datla does not expressly disclose APIs for accessing its knowledge graph. As I discussed in the Technology Background section above, using APIs to access knowledge graphs was well known and widely used in the art prior to the '882 patent, and I incorporate by reference that discussion herein. In the combination, Gnanasambandam supplies this element.

178. Gnanasambandam discloses publishing APIs for a medical knowledge graph: “Application back-end component 9104 is configured to publish one or more APIs 9103. As described previously, the one or more APIs 9103 may use artificial intelligence to perform analytics based on a knowledge graph pertaining to ontological data of a medical condition.” Ex. 1007 at 100:24–34; FIG. 91.



Ex. 1007 at FIG. 91 (annotated)

179. Gnanasambandam further discloses providing the APIs in a user interface: “As further shown in FIG. 92, at step 9204, method 9200 continues with providing the at least one API in a user interface of a computing device.” Ex. 1007 at 102:22–24.

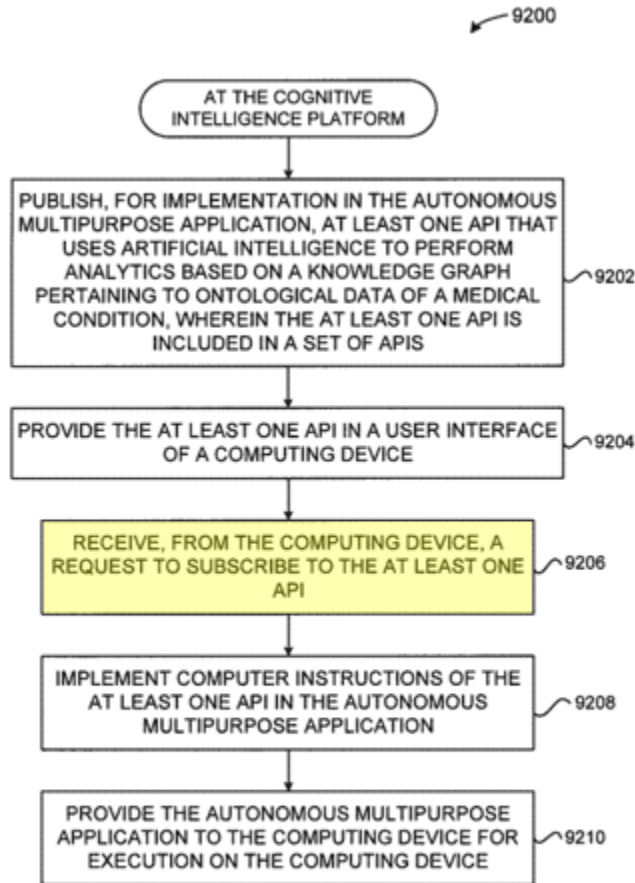


FIG. 92

Ex. 1007 at FIG. 92 (annotated)

180. In the combination, Datla discloses generating a healthcare determination by accessing linked entries in the knowledge graph. Datla's system queries the knowledge graph using extracted patient symptoms, traverses edges to connected condition nodes, and generates ranked diagnoses as the healthcare determination. "[T]he ranked medical conditions and/or diagnosis are provided to the clinician. The medical conditions and/or diagnosis can be provided to the user

via any user interface that allows information to be conveyed, such as a speaker or screen, among many other types of user interfaces.” Ex. 1005 at ¶ [0062].

181. Gnanasambandam’s APIs provide the interface mechanism through which an external application accesses the knowledge graph to generate those determinations. In the combined system, Gnanasambandam’s published APIs serve as the interface through which an application integrates with Datla’s knowledge graph to access linked entries and generate healthcare determinations.

182. A POSITA would have been motivated to combine Datla and Gnanasambandam in this manner for at least the following reasons.

183. First, Gnanasambandam, like Datla, operates in the field of medical knowledge graph-based decision support. Gnanasambandam builds knowledge graphs representing knowledge of diseases and uses them for analytics, diagnoses, and healthcare determinations. Ex. 1007 at 51:27–34, 55:50–58, 100:24–28. A POSITA building a medical knowledge graph system would naturally look to Gnanasambandam for techniques to deploy that system as a service.

184. Second, a POSITA would have recognized that publishing APIs for a knowledge graph-based system is the standard way to make such a system accessible to external applications, including clinical decision support tools, electronic health record systems, and other healthcare applications. Gnanasambandam teaches this deployment pattern: the back-end component publishes APIs that perform analytics

based on a knowledge graph, and these APIs are provided in a user interface to a client device. Ex. 1007 at 100:24–34; 102:22–24.

185. Third, Datla is designed for external integration because its system “accepts natural language input from a medical professional about a patient’s scenario, processes the input, and provides one or more possible diagnoses, tests, and/or treatments.” Ex. 1005 at ¶ [0028]. This input–output architecture is naturally suited for API-based access. A POSITA would have recognized that wrapping Datla’s diagnostic inference functionality in Gnanasambandam’s API layer would allow any healthcare application to query the knowledge graph and receive diagnostic determinations.

186. Adding an API layer to a knowledge graph-based diagnostic system involves applying a known technique (API publication, as taught by Gnanasambandam) to a known system (Datla’s knowledge graph) to yield the predictable result of an API-accessible medical knowledge graph system that generates healthcare determinations in response to application queries. A POSITA would have had a reasonable expectation of success because API publication for knowledge-based systems was well-established in the art, and no technical barrier would have impeded the combination.

3. Dependent Claim 2

187. Dependent claim 2 recites: “A computer system as recited in claim 1, wherein the instructions, when executed, further cause the one or more processors to: display on a computing device a portion of said graph database which includes said second entry of said first medical condition and each of said diagnostic factor entries and the links between said second entry and said diagnostic factor entries.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

188. Datla generates the portion of the graph that claim 2 requires. Datla’s processor queries the knowledge graph “to generate a diagnosis graph as a subset of the knowledge graph,” and identifies “a ranked list of one or more medical conditions” from that diagnosis graph. Ex. 1005 at ¶ [0007]. The diagnosis graph is “a connected digraph that represents the connected symptoms,” *id.* at ¶ [0006]. It is built by expanding from the patient’s symptom nodes to the connected condition nodes. *Id.* at ¶¶ [0055]–[0057]. The diagnosis graph is therefore a portion of Datla’s graph database. It includes a medical condition node, the symptom nodes linked to it, and the edges between them. Datla provides its results to the user “via any user interface that allows information to be conveyed, such as a speaker or screen.” *Id.* at ¶ [0062]. Datla does not expressly state that the diagnosis graph itself is displayed. Goldberg teaches that display.

189. Goldberg discloses displaying a portion of a medical knowledge graph that includes a disease, its associated findings, and the links between them. Goldberg describes “transmitting . . . a medical knowledge graph to a user, the medical knowledge graph depicting a plurality of nodes and a plurality of connectors between the plurality of nodes,” where the connectors represent “linkages between the plurality of nodes.” Ex. 1006 at 14:32–42. Goldberg’s graph includes node sets representing diseases, symptoms, and laboratory tests. *Id.* at 15:60–18:38.

190. Goldberg discloses a “mobile app or web interface enables browsing the above mentioned Associative Medical Knowledge Base,” and that interface “may show any depth of associations. (e.g. directly related symptoms of a certain disease, or symptoms related via one or more complications).” *Id.* at 8:63–67, 9:24–29. Goldberg uses a differential diagnosis list “as a gateway for lookup of a specific disease,” and shows “all or a selected level of associations known in the literature for this disease, both direct and in-direct.” *Id.* at 12:6–11. For example, when looking up Pneumonia, “[t]he system may subsequently show all associations with findings, lab tests, etc.” *Id.* at 11:61–63.

191. Goldberg’s FIGS. 9 and 10 illustrate such displays. They are hybrid graphs showing the “clinical relations connecting Pneumonia and Fever” and the “connections between Myocardial infarction and abdominal pain,” in which arrows represent causal relationships or clinical associations. *Id.* at 5:5–10, FIGS. 9–10.

Goldberg thus displays a portion of the graph that includes a disease node, the finding nodes linked to it, and the links between them.

192. A POSITA would have been motivated to display Datla’s diagnosis graph as Goldberg teaches, for three reasons. First, Datla seeks to provide a clinical assistant that recommends diagnoses and treatments “in addition to the rationale for each option.” Ex. 1005 at ¶ [0003]. Goldberg teaches a way to show that rationale. Its application “can show a ‘path to decision’. I.e. the clinical reasoning path of associations which were used in order to make a certain diagnosis or recommend a next step.” Ex. 1006 at 8:40–46. Second, Datla already generates the relevant portion of the graph as a subset of the knowledge graph connecting the patient’s symptoms to candidate conditions. Ex. 1005 at ¶¶ [0007], [0056]. Displaying that existing subset requires no change to how Datla builds or traverses its graph. Third, displaying the condition, its linked findings, and the links between them allows the clinician to verify the basis for a suggested diagnosis. Goldberg explains that, for a displayed path, “the original citations may be read, searched, or otherwise accessed.” Ex. 1006 at 11:61–67. Clinicians may be more likely to trust a decision-support suggestion when they can see why the system made it. Displaying knowledge-graph content was a routine technique.

193. In the combined system, an application accessing the graph database through Gnanasambandam’s APIs, as described for Element [1h], *supra* § X.A.2.i,

would display on a computing device a portion of the graph database that includes the entry for the first medical condition, the diagnostic factor entries linked to it, and the links between them.

4. Dependent Claim 3

194. Dependent claim 3 recites: “A computer system as recited in claim 1 wherein said each separate entry in said graph database is a node.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

195. Datla discloses the additional limitation of claim 3. As I explained for Element [1f], each diagnostic factor and each medical condition in Datla’s knowledge graph is stored as a separate entry, and that entry is a node. *Supra* § X.A.2.g. Datla describes the graph this way: “[T]he knowledge graph 310 is a tree-like structure with a plurality of nodes connected by one or more edges. Each root node of the graph is a symptom, and the remaining nodes are conditions, diagnoses, tests, procedures, medications, or other clinical concepts.” Ex. 1005 at ¶ [0031]; *see also id.* at ¶ [0035] (“each node is a medical condition, diagnosis, test, procedure, medication or any other clinical concept, and each edge is a relation between two nodes”). Datla’s claims likewise recite “a knowledge graph . . . comprising a plurality of nodes, at least some of the nodes comprising a respective patient symptom.” *Id.* at claim 1. Each clinical concept in Datla’s graph, whether a symptom, condition, diagnosis, test, procedure, or medication, is stored as its own

node. Each separate diagnostic factor and medical condition entry in Datla's graph database is a node.

5. Dependent Claim 8

196. Dependent claim 8 recites: "A computer system as recited in claim 1, wherein at least one of said diagnostic factors for a second medical condition being the same as one of said diagnostic factors for said first medical condition, wherein said second medical condition being linked to said at least one diagnostic factor and said first medical condition being linked to said at least one diagnostic factor." I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

197. Datla discloses this limitation. Datla's knowledge graph is built around symptom nodes that are shared across multiple medical conditions. Datla describes a knowledge graph "which possesses symptoms as root nodes connected by edges to the diseases and medical conditions associated with those symptoms." Ex. 1005 at ¶ [0036]; *see also id.* at ¶ [0006] (describing a digraph "which has symptoms as leaf nodes which are connected to diseases and medical conditions"). In Datla, "[a]n edge is a relationship between two nodes," and "a symptom of a fever will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios." *Id.* at ¶ [0031]. Based on these teachings, a POSITA would

understand that a single diagnostic factor node, such as fever, is linked by edges to multiple disease and medical condition nodes.

198. Datla thus discloses a diagnostic factor (e.g., fever) that is a diagnostic factor for a first medical condition and is the same diagnostic factor for a second medical condition. Datla also discloses that the first medical condition and the second medical condition are each linked to that diagnostic factor. Datla's contrasting example where a symptom of nystagmus "will have few nodes as it is a symptom of fewer patient scenarios" confirms this. Ex. 1005 at ¶ [0031]. The number of edges from a symptom node reflects the number of conditions associated with that symptom. A symptom associated with two or more conditions is therefore linked to two or more condition nodes.

6. Dependent Claim 9

199. Dependent claim 9 recites: "A computer system as recited in claim 8 wherein traversing said graph database from said at least one diagnostic factor to said second medical condition produces a second rank for said at least one diagnostic factor in the context of said second medical condition which is different from a first rank produced by traversing from said at least one diagnostic factor to said first medical condition." I incorporate by reference my discussion of independent claim 1 and dependent claim 8 above. *Supra* §§ X.A.2, X.A.5.

200. In the ‘882 patent, “[r]ank may be expressed in a source using numerical values, or words and phrases to refer to the likelihood that group of factors is caused by a particular condition, etc.” Ex. 1001 at 11:46–49; *see also id.* at 17:30–35. The same factor may be ranked differently for different conditions: “because fever 310 may be ranked differently for condition 300 and condition 340, these rankings are held in nodes 304 and 344.” *Id.* at 19:66–20:4. Claim 9 does not require that the rank be stored in a particular structure and only requires that traversal from a shared factor to two different conditions produce two different ranks.

201. Datla discloses the structure claim 9 requires. As I explained for claim 8, a single symptom node in Datla, such as fever, “will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios.” Ex. 1005 at ¶ [0031]; *supra* § X.A.5. Datla’s edges carry weights that reflect the association between the two nodes they connect: “edges between the nodes are also weighted based on a relationship between the nodes,” where “[t]he relationship may be the frequency with which the two nodes are associated within a source or a corpus,” and “[t]he weight may be variable and based on the strength of the relationship (for example, the frequency that the two nodes appear together).” Ex. 1005 at ¶ [0032]; *see also id.* at ¶ [0010] (“at least some of the edges of the knowledge graph are weighted”). Fever does not appear with the same frequency across hundreds of different conditions in the medical literature. A POSITA would

understand that at least some of the frequency-based weights on fever's edges differ from one another. In other words, the weight on the edge from fever to a first condition differs from the weight on the edge from fever to a second condition. Each weight reflects the strength of fever's association with that condition and is a rank for fever in the context of that condition.

202. To the extent Datla does not expressly describe its edge weights as condition-specific likelihoods applied during traversal, Goldberg confirms and makes explicit what a POSITA would understand from Datla. Goldberg's knowledge base "holds a probabilistic association between various disease and disorder features," and a textbook chapter or article describing a disease "may be translated into a number of such associations indicating the probability which links each feature with the disease." Ex. 1006 at 5:66–6:7. Because each probability links a feature to a particular disease, the same feature carries a separate probability for each disease that it is associated with. Goldberg shows this with a "feature by feature comparison for two selected diseases (CROHN's and CELIAC)" and "a list of diseases with likelihood to present given features (Failure to thrive & ataxia)." *Id.* at 4:56–63, 9:1–23 (describing FIGS. 4–5). Goldberg's diagnostic application uses these associations when it evaluates a patient and "takes into consideration the strength of the various associations as indicated in the knowledge base" and selects

one or more pathways on the knowledge graph to generate differential diagnoses. *Id.* at 7:30–36.

203. It would have been obvious to apply Datla’s edge weights when activation propagates from a symptom node to a condition node. Goldberg teaches that the diagnostic algorithm should take the strength of each feature–disease association into account when it evaluates the patient. Ex. 1006 at 7:25–36. A POSITA would have recognized that Datla’s edge weights exist to capture exactly that strength of association. Using those weights during traversal would have produced one rank for fever in the context of the first condition and a different rank in the context of the second condition. This would have improved diagnostic accuracy in a predictable way, and it required no change to Datla’s graph structure. Ex. 1005 at ¶¶ [0010], [0032].

204. I note that per the plain claim language, the claimed rank need not be held in a separate node. Claim 9 is satisfied whether the condition-specific rank is carried on an edge, as in Datla, or held in an intermediate data structure.

7. Dependent Claim 11

205. Dependent claim 11 recites: “A computer system as recited in claim 1 wherein the instructions, when executed, further cause the one or more processors to: retrieve at least one laboratory test that is associated with one or more of said plurality of medical conditions; and store said laboratory test as a separate entry in

said graph database, one of said medical conditions being linked to said laboratory test, wherein an entry of said laboratory test in said graph database does not include information specific to said medical condition linked to said laboratory test.” I incorporate by reference my discussion of independent claim 1 above, including Elements [1f] and [1g]. *Supra* § X.A.2. Claim 11 applies to laboratory tests the same structure that claim 1 recites for diagnostic factors.

206. **Separate test entries linked to conditions.** Datla’s knowledge graph stores tests as their own nodes: “Each root node of the graph is a symptom, and the remaining nodes are conditions, diagnoses, tests, procedures, medications, or other clinical concepts. An edge is a relationship between two nodes.” Ex. 1005 at ¶ [0031] (emphasis added); *see also id.* at ¶ [0035] (“each node is a medical condition, diagnosis, test, procedure, medication or any other clinical concept, and each edge is a relation between two nodes.”) (emphasis added).

207. A POSITA would understand that the “tests” represented in Datla’s knowledge graph include laboratory tests. Laboratory tests are among the most common tests described in the clinical–medicine sources from which Datla builds its graph. Ex. 1005 at ¶ [0030]. Datla’s processing of clinical information is consistent with this understanding. Datla extracts “lab results” from clinical input, *id.* at ¶¶ [0006], [0052], and teaches that its system “may” map topical keywords to “categories represented in” ontologies including “LOINC for

tests,” *id.* at ¶ [0054]. Datla also links test nodes to condition nodes. Its system identifies tests for a patient from the diagnosis graph, and “may recommend a test to distinguish between two possible diagnoses.” *Id.* at ¶¶ [0007], [0063].

208. Retrieving a laboratory test associated with a condition. Datla discloses retrieving tests associated with medical conditions from its corpus of medical information. Datla constructs its knowledge graph “from a corpus of medical information,” including “medical journals, newspapers, online sources such as Wikipedia, and other sources,” with “information being extracted from each page.” Ex. 1005 at ¶ [0030]. The resulting nodes include “tests.” *Id.* at ¶¶ [0031], [0035]. Each test node in Datla’s graph is therefore a test retrieved from the corpus. Datla mines tests associated with particular diagnoses. Datla uses “the list of possible diagnoses . . . to extract a list of candidate Wikipedia articles to mine related tests, and treatments (from sections and subsections of the Wikipedia article),” *id.* at ¶ [0037], and extracts “corresponding treatment and test information . . . from the curated corpus,” *id.* at ¶ [0073].

209. A POSITA would understand that the “tests” represented in Datla’s knowledge graph include laboratory tests. As I described above, laboratory tests are among the most common tests described in the clinical-medicine sources from which Datla builds its graph. *See id.* at ¶ [0030]. Datla’s processing of clinical information is consistent with this understanding. Datla extracts “lab results” from clinical

input, *id.* at ¶¶ [0006], [0052], and teaches that its system “may” map topical keywords to “categories represented in” ontologies including “LOINC for tests,” *id.* at ¶ [0054]. LOINC is, as Goldberg explains, “an ontology dedicated to laboratory tests and procedures.” Ex. 1006 at 6:17–20. I incorporate by reference my discussion of LOINC in the Technology Background section above.

210. **Test entries lacking condition-specific information.** For the reasons I provided for Element [1g], Datla places the relationship between two nodes on the edge connecting them, not in either node. “[E]dges between the nodes are also weighted based on a relationship between the nodes,” such as “the frequency with which the two nodes are associated within a source or a corpus.” Ex. 1005 at ¶ [0032]; *supra* § X.A.2.h. That design applies equally to test nodes. A test that distinguishes between two possible diagnoses is associated with both. Ex. 1005 at ¶ [0063]. Holding information specific to either condition in the test node itself would be inconsistent with that use. Datla therefore discloses, or at minimum renders obvious, laboratory test entries that do not include information specific to the medical conditions linked to them.

211. To the extent Datla is found not to disclose that its test nodes exclude condition-specific information, this would have been obvious in view of Goldberg. Goldberg’s hybrid knowledge graph integrates “a second knowledge graph . . . derived from . . . an ontology dedicated to laboratory tests and procedures (e.g.,

LOINC),” together with a separate “clinical probability graph that describes clinical statistical information that links medical concepts that are described in the Ontological graphs.” Ex. 1006 at 6:17–24. Goldberg explains that such ontologies are “restricted to health terms and their semantic relations,” and that “[c]linical relations were out of scope for such Ontologies.” *Id.* at 1:59–61. In Goldberg, a laboratory test entry describes the test itself and its association with a particular disease is not held in the test entry.

212. A POSITA would have been motivated to represent Datla’s laboratory test nodes the same way, for two reasons. First, Datla teaches that its system “may” map topical keywords to “categories represented in” ontologies including “LOINC for tests.” Ex. 1005 at ¶ [0054]. Goldberg teaches a known way to incorporate LOINC tests into a medical knowledge graph while keeping disease associations out of the test entries. Second, Datla uses its graph to “recommend a test to distinguish between two possible diagnoses” and to generate testing plans. *Id.* at ¶¶ [0012], [0063]. That use requires a single test entry that can be linked to more than one condition, and Goldberg’s approach provides one. These reasons supplement the motivation to combine Datla and Goldberg that I discussed for claim 1. *Supra* §§ X.A.1, X.A.2.e.

8. Dependent Claim 12

213. Dependent claim 12 recites: “A computer system as recited in claim 1 wherein said graph database includes more than twenty thousand entries.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2. As I explained for dependent claim 3, each node in Datla’s knowledge graph is a separate entry. *Supra* § X.A.4.

214. Datla does not state a node count. However, a POSITA would have understood from Datla, and would have found obvious, that Datla’s knowledge graph contains well over twenty thousand entries. Datla builds its knowledge graph from a broad corpus of medical information, “including but not limited to medical journals, newspapers, online sources such as Wikipedia, and other sources.” Ex. 1005 at ¶ [0030]. When Datla uses an online source with a master top category such as “clinical medicine,” “all pages including and under the master top category are analyzed,” and information is extracted from each page. *Id.* Datla’s nodes are not limited to one type of concept. They include symptoms, “conditions, diagnoses, tests, procedures, medications, or other clinical concepts.” *Id.* at ¶ [0031]; *see also id.* at ¶ [0035] (“each node is a medical condition, diagnosis, test, procedure, medication or any other clinical concept”). Datla also reports that a single symptom node, fever, “will be connected by edges to hundreds of other nodes.” *Id.* at ¶ [0031]. A graph in which one common symptom connects to hundreds of other nodes, and

which covers every clinical-medicine page along with the tests, procedures, and medications drawn from them, would contain tens of thousands of nodes. Datla also teaches that the graph grows over time: it “is updated continuously and/or periodical[ly] as the medical knowledge source is updated,” for example “with new journal articles, new online sources, or other possible sources of new information.” *Id.* at ¶ [0032].

215. Goldberg confirms that clinical knowledge graphs routinely exceed twenty thousand entries. Goldberg’s knowledge base “integrates two or more separate knowledge graphs,” including “a first knowledge graph [that] is derived from a healthcare Ontology which includes a large vocabulary of medical concepts. (e.g., SNOMED),” and “[a] second knowledge graph . . . derived from . . . an ontology dedicated to laboratory tests and procedures (e.g., LOINC).” Ex. 1006 at 6:10–34. Goldberg describes SNOMED as “an encompassing Ontology covering hundreds of thousands of medical terms arranged as an ontology.” *Id.* at 1:54–2:1. Hundreds of thousands of medical terms represents far more than twenty thousand entries. For the reasons given in my analysis of claim 1, a POSITA would have been motivated to combine Goldberg’s teachings with Datla. *Supra* § X.A.2. The resulting knowledge graph, which draws on standard clinical vocabularies, would have contained more than twenty thousand entries.

216. Additionally, the number of entries recited in claim 12 would have been obvious as a matter of routine scaling. A graph with fewer than twenty thousand entries differs from one with more than twenty thousand entries only in how many conditions, symptoms, tests, and other concepts it represents. The additional entries are more instances of the same node types Datla already discloses, connected by the same kind of weighted edges. Ex. 1005 at ¶¶ [0031]–[0032]. Adding more entries is nothing more than a duplication of the components Datla already discloses. Datla expressly contemplates this growth through continuous updating from new sources. *Id.* at ¶ [0032]. Goldberg similarly discloses that its knowledge base “may be updated on an ongoing basis” by “a large group of contributors,” because of “[t]he amount of available medical knowledge and its growth rate.” Ex. 1006 at 7:51–56.

217. Scaling the graph past twenty thousand entries would produce no new or unexpected result. The system would work the same way and would apply the same symptom weighting, the same activation-and-decay traversal, and the same ranking to a larger set of the same types of nodes. Ex. 1005 at ¶¶ [0053], [0055]–[0059]. The only predictable effect would be broader coverage of medical conditions, which is why a POSITA would be motivated to build a larger graph. A diagnostic system intended to support differential diagnosis across clinical medicine would naturally cover thousands of conditions, and therefore tens of thousands of entries once the related symptoms, tests, and treatments are included. Implementing

a graph of that size was well within the abilities of a POSITA. As I described in the Technology Background, graph and ontology-based systems with hundreds of thousands of concepts, including SNOMED CT itself, were routine. *Supra* § VI.

218. The ‘882 patent confirms that the twenty-thousand-entry threshold has no technical significance. The specification does not describe any benefit, change in operation, or unexpected result tied to twenty thousand entries. It states only that “an actual graph database will be much more complex and contain many hundreds of thousands of nodes or more, meaning that an automated search algorithm is used.” Ex. 1001 at 27:67–28:3.

9. Dependent Claim 13

219. Dependent claim 13 recites: “A computer system as recited in claim 1 wherein said graph database includes more than one hundred thousand entries.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

220. Claim 13 differs from claim 12 only in the threshold: more than one hundred thousand entries rather than more than twenty thousand. For the reasons below, my analysis of claim 12 applies equally to claim 13.

221. First, Goldberg supplies or renders obvious the higher threshold. As I explained for claim 12, Goldberg’s knowledge base integrates a knowledge graph derived from SNOMED. Goldberg describes SNOMED as “an encompassing

Ontology covering hundreds of thousands of medical terms.” Ex. 1006 at 6:12–34, 1:54–56. That vocabulary exceeds one hundred thousand entries before counting Goldberg’s LOINC-derived graph or its clinical probability graph. For the reasons I gave for claims 1 and 12, a POSITA would have been motivated to combine Goldberg’s teachings with Datla. *Supra* § X.A.2. The resulting graph would contain well over one hundred thousand entries.

222. Second, my routine-scaling analysis for claim 12 applies with equal force. Datla’s graph covers every page under a master category such as “clinical medicine.” It is updated continuously with new journal articles and other sources, and it has no stated upper limit. Ex. 1005 at ¶¶ [0030]–[0032]. Going from twenty thousand to one hundred thousand entries adds more instances of the same node types, connected by the same weighted edges. The system applies the same weighting, traversal, and ranking to the larger graph, and no new or unexpected result follows. The ‘882 patent confirms that the threshold is not critical. It states that “an actual graph database will be much more complex and contain many hundreds of thousands of nodes or more,” and ties no benefit or change in operation to one hundred thousand entries. Ex. 1001 at 27:67–28:3.

10. Dependent Claim 14

223. Dependent claim 14 recites: “A computer system as recited in claim 1 wherein said healthcare determination is a recommendation, an evaluation, a result,

or a conclusion.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

224. The claim is disjunctive and requires only that the healthcare determination be “a recommendation, an evaluation, a result, or a conclusion.” Datla’s output satisfies each alternative.

225. Datla is configured to “identify a ranked list of one or more medical conditions, diagnoses, treatments, and/or tests for the patient from the diagnosis graph,” and those identified conditions, diagnoses, treatments, and/or tests “are provided to the user via the user interface.” Ex. 1005 at ¶ [0007]; *see also id.* at ¶ [0062] (“the ranked medical conditions and/or diagnosis are provided to the clinician”). The ranked list is a “result,” because it is the output of querying the knowledge graph. It is also an “evaluation” of the patient’s likely conditions and a “conclusion.” Datla identifies “the most probable diagnosis given a clinical narrative,” *id.* at ¶ [0057], and provides “the highest ranked one or more medical conditions, diagnoses, treatments, and/or tests for the patient” to the user, *id.* at ¶ [0011].

226. Datla’s output is also a “recommendation.” Datla describes a system that “provides one or more possible diagnoses, tests, and/or treatments.” Ex. 1005 at ¶ [0028]. Datla states that the system “can effectively recommend the diagnosis, and retrieve summarized test and treatment options from curated data sources for patient

scenario.” *Id.* at ¶ [0061]. Datla also states that the system “may recommend a test to distinguish between two possible diagnoses.” *Id.* at ¶ [0063]. A POSITA would understand these outputs as recommendations presented to the clinician for decision-making.

227. As I explained for element [1h], in the combined system, an application uses Gnanasambandam’s APIs to access the linked entries of Datla’s knowledge graph and generate these outputs. *Supra* § X.A.2.i. Datla’s ranked diagnoses, recommended tests, and treatment plans are therefore the claimed “healthcare determination,” and each is a recommendation, an evaluation, a result, or a conclusion.

B. Ground 2: Claims 4-5 are Obvious Over Datla, Goldberg, Gnanasambandam, and Kim

228. In my opinion, claims 4-5 of the ‘882 patent are obvious over Datla, Goldberg, Gnanasambandam, and Kim.

1. Motivation to Combine

229. A POSITA would have been motivated to incorporate Kim’s teaching of ontology-annotated graph entries into the Datla–Goldberg–Gnanasambandam system for at least the following reasons.

230. First, these references are in the same field and directed to the same problem. Kim, like Datla and Goldberg, concerns medical knowledge graphs that support clinical decision-making. Ex. 1012 at ¶¶ [0038]–[0039]. Datla and Goldberg

both extract clinical concepts from heterogeneous textual sources. A system that ingests concepts from multiple sources would beneficially recognize when different sources use different words for the same concept. Kim addresses exactly that problem. For example, Kim explains that “the expressions ‘lung tumor’ and ‘tumor located in the lung’ means semantically the same and will therefore be annotated with the same identifiers of an ontology.” Ex. 1012 at ¶ [0050].

231. Second, Datla and Goldberg already point toward standard ontologies. Datla teaches that its system “may” map topical keywords extracted from patient descriptions, summaries, or notes to “categories represented in” ontologies SNOMED CT, LOINC, and RxNorm (Ex. 1005 at ¶ [0054]) and Goldberg derives its ontology graphs from SNOMED and LOINC. Kim supplies the implementation detail that follows naturally: store the ontology code, and the ontology’s synonyms, in the graph entry itself.

232. Third, the combination involves a known technique yielding predictable results. Annotating graph entries with ontology codes and synonyms was a known technique, and Kim describes off-the-shelf tools for doing it. Kim explains that “the annotation of template elements with unique identifiers can be (semi-) automated using machine learning-based medical concept annotation tools,” such as the open-source MedCAT tool. Ex. 1012 at ¶ [0053]. Adding those annotations to

Datla's nodes would have been a predictable use of a known technique for its established function.

233. Kim's teachings do not depend on Kim's structured-report setting. Kim's annotations serve to recognize semantically equivalent concepts, map one ontology to another, and prevent redundant concepts. Ex. 1012 at ¶¶ [0050]–[0051], [0054]. Those benefits apply equally to any medical knowledge graph built from multiple sources, including Datla's and Goldberg's.

2. Dependent Claim 4

234. Dependent claim 4 recites: “A computer system as recited in claim 1 wherein said each diagnostic factor entry includes a unique identifier which links to an entry in a medical ontology representing said each diagnostic factor.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

235. Kim teaches graph entries that include unique ontology identifiers. Kim discloses concepts “represented as nodes” that “may be identified by a unique ontology code” and explains that “[u]sing unique node ids/codes, e.g. ontology codes, allows the unambiguous identification of every concept instance.” Ex. 1012 at ¶ [0041]; *see also id.* at ¶¶ [0050]–[0051]. Kim confirms that an “ontology code” is a “unique identifier.” *Id.* at ¶ [0052].

236. Kim further teaches that “unique identifiers of ontology concepts are assigned to individual nodes” and that “the entries of the graph database may themselves be annotated with identifiers from several different ontologies.” Ex. 1012 at ¶¶ [0051], [0060]. Kim’s identifier links to an entry in a medical ontology. Kim’s identifiers are codes drawn from the ontologies themselves, for example “the SNOMED CT code SCTID: 93880001” and “the RadLex code RID45686” for lung cancer. *Id.* at ¶ [0051]. A POSITA would understand that a SNOMED CT concept identifier (SCTID) designates, and thereby links to, the specific SNOMED CT concept entry that represents the clinical concept.

237. Kim also teaches using the identifier to reach related ontology concepts: “[o]nce an ontology code (unique identifier) has been assigned . . . related concepts as represented in the graph database can be suggested.” Ex. 1012 at ¶ [0052].

238. It would have been obvious to include, in each of Datla’s diagnostic factor nodes, the unique SNOMED CT (or other standard ontology) identifier corresponding to that factor, as Kim teaches. Datla teaches that, in an embodiment, its system “may extract term frequency-inverse document frequency (TFIDF) weighted topical keywords from the given descriptions, summaries or notes and map them to categories represented in one or more ontologies, including but not limited to the following controlled clinical ontologies: SNOMED CT for diagnoses, LOINC for tests, and/or RxNorm for treatments.” Ex. 1005 at ¶ [0054]. Datla does not state

that the nodes of its knowledge graph carry ontology identifiers; Kim supplies that teaching. Kim teaches that graph-database entries are identified by a “unique ontology code” and “may themselves be annotated with identifiers from several different ontologies.” Ex. 1012 at ¶¶ [0041], [0051]. In the combined system, each diagnostic factor entry would include a unique identifier linking to the entry in the medical ontology that represents that factor.

239. A POSITA would have been motivated to include Kim’s ontology identifier in each diagnostic factor entry for several reasons.

240. First, the identifier resolves terminology variation. Datla builds its graph from multiple sources that describe the same clinical concept in different words. Datla acknowledges the problem: “certain diagnosis terms from MIMIC-II do not exactly match the Wikipedia page titles.” Ex. 1005 at ¶ [0040]. Kim teaches that ontology identifiers solve it, because semantically equivalent expressions “will therefore be annotated with the same identifiers of an ontology.” Ex. 1012 at ¶ [0050]. A POSITA would have recognized that a stable identifier on each diagnostic factor node ensures that a factor described differently in different sources is stored and matched as a single node.

241. Second, the identifier makes Datla’s own query-side mapping usable for matching. Datla teaches that its system “may” map topical keywords extracted from “the given descriptions, summaries or notes” to “categories represented in”

ontologies including “SNOMED CT for diagnoses, LOINC for tests, and/or RxNorm for treatments,” and Datla queries its knowledge graph with the extracted patient symptoms. Ex. 1005 at ¶¶ [0054]–[0055]. A POSITA would have recognized that storing the same standard identifiers on the graph’s diagnostic factor nodes would allow a mapped patient symptom to be matched to the corresponding node through a shared code, rather than through text matching, which Datla acknowledges is imperfect. *See* Ex. 1005 at ¶ [0040] (“certain diagnosis terms from MIMIC-II do not exactly match the Wikipedia page titles”). Storing the ontology code in each node gives the combined system a common key for reconciling concepts across those ontologies.

242. Third, the identifier provides access to the ontology’s structure. SNOMED holds sets of relations and qualifiers that support detailed description of medical concepts with findings and disorders grouped in families. Ex. 1006 at 1:65–2:1. Goldberg’s FIG. 6 illustrates SNOMED is-a relations, for example between bacterial pneumonia, infective pneumonia, and pneumonia. *Id.* at 9:50–55, FIG. 6. A POSITA would have understood that storing the SNOMED code in the diagnostic factor node is the conventional way to connect that node to the ontology’s hierarchy. Kim confirms the benefit, because the identifier allows for “related concepts as represented in the graph database” to be identified. Ex. 1012 at ¶ [0052].

243. A POSITA would have reasonably expected success in including Kim’s ontology identifier in each of Datla’s diagnostic factor nodes. Goldberg shows that clinical knowledge graphs had already been combined with standard ontologies. Automated tools for assigning ontology identifiers were also available. Kim identifies MedCAT, which “is based on an algorithm mapping the terms of any kind of textual data to the concepts of existing ontologies.” Ex. 1012 at ¶ [0053]. The modification would also leave Datla’s operation intact. An identifier is a property stored on a node. It does not alter the weighted edges or the activation that flows “directly from the root nodes to the entire graph.” Ex. 1005 at ¶¶ [0032], [0036].

3. Dependent Claim 5

244. Dependent claim 5 recites: “A computer system as recited in claim 1 wherein said each diagnostic factor entry includes at least one synonym of said diagnostic factor represented by said each diagnostic factor entry.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

245. Kim teaches graph entries that include synonyms. Kim teaches that “the graph database includes synonyms, translations, concepts and relationships, and the proposed methods and systems allow to add new synonyms, translations, concepts and relationships.” Ex. 1012 at ¶ [0054]. Kim ties those synonyms to individual concept entries. Kim explains that “in order to add translations or synonyms for terms of existing ontology concepts to one or more entries of the graph database, the

respective concept might have to be identified using existing terms describing it.” *Id.* at ¶ [0055]. Kim also teaches searching “all preexisting concept synonyms within the graph database,” *id.* at ¶ [0054], and that “synonyms and translations are also covered by the terminology system of the graph database and can be matched to the entered node labels.” *Id.* at ¶ [0060].

246. It would have been obvious to include, in each of Datla’s diagnostic factor nodes, at least one synonym of the diagnostic factor that the node represents, as Kim teaches. For example, in the combined system, the node for fever would include the synonyms by which that finding is described in the medical literature and in the corresponding ontology entry. A POSITA would have been motivated to include synonyms in each diagnostic factor entry for several reasons.

247. First, synonyms prevent duplicate nodes. Kim teaches that “the addition of redundant concepts could be prevented by searching through all preexisting concept synonyms within the graph database.” Ex. 1012 at ¶ [0054]. Datla’s graph is updated continually as new material arrives: “the graph may be updated with new journal articles, new online sources, or other possible sources of new information.” Ex. 1005 at ¶ [0032]. Without synonyms stored on each node, the same diagnostic factor described in different words in a new source (for example, “pyrexia” rather than “fever”) would risk being added as a separate node. That would fragment the

edges that link the factor to its associated conditions. Storing synonyms on each node lets the system recognize the new term as an existing factor.

248. Second, synonyms improve matching of input text to graph nodes. Datla's system queries its knowledge graph with symptoms extracted from clinical notes. Ex. 1005 at ¶¶ [0052], [0055]. Datla recognizes that clinical terms do not always match their reference terms exactly. *Id.* at ¶ [0040]. Kim teaches that synonyms stored in the graph's terminology system "can be matched to the entered node labels," Ex. 1012 at ¶ [0060]. Kim also suggests that "medical synonym dictionaries can be applied to infer most likely matches," particularly where users enter free text. *Id.* at ¶ [0073]. A POSITA would have recognized that storing synonyms on each diagnostic factor node improves the likelihood that a patient symptom expressed in a common form is matched to the correct node, which improves diagnostic accuracy.

249. Third, synonyms complement the identifier of claim 4. Kim links synonyms to ontology concepts. It explains that synonyms may be added "for terms of existing ontology concepts." Ex. 1012 at ¶ [0055]. Standard ontologies such as SNOMED CT and UMLS provide multiple terms for each concept. A POSITA who linked each diagnostic factor node to an ontology entry, as in claim 4, would have found it natural to store that entry's terms with the node as well.

250. **Reasonable expectation of success.** Storing one or more text strings as a node property is routine. Kim confirms that its graph database already “includes synonyms” and allows new synonyms to be added. Ex. 1012 at ¶ [0054]. A POSITA would therefore have reasonably expected success.

C. Ground 3: Claim 6 is Obvious Over Datla, Goldberg, Gnanasambandam, and Van Assel

251. In my opinion, claim 6 of the ‘882 patent is obvious over Datla, Goldberg, Gnanasambandam, and Van Assel.

1. Dependent Claim 6

252. Dependent claim 6 recites: “A computer system as recited in claim 1 wherein said each diagnostic factor entry includes a temporality attribute, a negation attribute or an experiencer attribute.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

253. The ‘882 patent describes the negation attribute as “a binary value indicating whether the factor is present or not present.” Ex. 1001 at 21:31–32.

254. Datla already determines negation status. Datla teaches that its system may “filter out negated clinical concepts in order to give more weight to positive clinical manifestations.” Ex. 1005 at ¶ [0054]. To filter a concept as negated, the system must first classify it as present or absent.

255. Van Assel teaches recording that status as a value associated with each item of medical data. Specifically, Van Assel “assigns 1, -1, or 0 to the extracted

concept to indicate whether the concept is validly present (1), validly absent (-1) or this is unknown (0).” Ex. 1011 at ¶ [0245]. Van Assel explains that this value is information about the concept itself: “Asthma is an item of medical data, while the information about the medical data is the confidence, i.e. whether the concept is validly present, validly absent, or this is unknown.” *Id.* at ¶ [0253]. Van Assel applies the same state to the nodes of its diagnostic model: “A node corresponding to an item of medical data can be present, ‘Evidence(node=node, state=PRESENT)’, or absent, ‘Evidence(node=node, state=ABSENT).” *Id.* at ¶ [0248]. Van Assel also represents absence in its concept language through unary concepts such as “Not Headache.” *Id.* at ¶ [0122].

256. It would have been obvious to include, in each of Datla’s diagnostic factor nodes, a negation attribute indicating whether the factor is present or not present, for at least three reasons. First, Van Assel is in the same field. Like Datla, it performs automated clinical diagnosis using a medical knowledge graph and extracts clinical concepts from text with NLP. Ex. 1011 at ¶¶ [0110], [0113], [0115]; Ex. 1005 at ¶ [0052]. Second, Datla matches extracted patient findings to nodes of its knowledge graph. Ex. 1005 at ¶¶ [0052], [0055]. Without a negation attribute, “fever” and “no fever” would map to the same node. Storing the negation status that Datla already computes on the node keeps the graph’s representation in the same form as the patient data matched against it. Third, Van Assel teaches that absence is

itself diagnostic evidence and uses it directly. Its evidence “may further comprise the absence of items of medical data,” and in its model “PRESENT becomes probability 1 and ABSENT becomes probability 0.” Ex. 1011 at ¶¶ [0112], [0248]. A negation attribute lets the combined system either weight positive findings more heavily, as Datla prefers, or use absent findings as evidence, as Van Assel teaches.

257. A POSITA would have reasonably expected success. The attribute is a simple value stored with a node. Van Assel implements such values (−1, 0, and 1) and PRESENT/ABSENT node states. Ex. 1011 at ¶¶ [0248], [0253]. Datla’s NLP already produces the negation status needed to populate it. Ex. 1005 at ¶ [0054]. Adding the attribute changes neither Datla’s weighted edges nor its activation-and-decay traversal. *Id.* at ¶¶ [0032], [0036].

D. Ground 4: Claims 7 and 15-20 are Obvious Over Datla, Goldberg, Dang, and Gnanasambandam

258. In my opinion, claims 7 and 15-20 of the ‘882 patent are obvious over Datla, Goldberg, Dang, and Gnanasambandam.

1. Motivation to Combine

259. For the reasons I provide for Ground 1, a POSITA would have been motivated to combine Datla, Goldberg, and Gnanasambandam. *Supra* § X.A.1. A POSITA would also have been motivated to add Dang to that combination, for the following reasons.

260. Dang, like Datla, builds a symptom–disease knowledge graph from medical literature and uses it for diagnosis. Dang generates its graph by “receiving as input one or more electronic documents constituting medical guidelines; extracting a set of diseases and symptoms from the medical guidelines . . . and generating models of the relations among symptoms and diseases based on the extracted information.” Ex. 1008 at ¶ [0025]. A POSITA working on Datla’s graph would naturally have looked to Dang for ways to structure it.

261. Dang supplies graph structure that Datla leaves unspecified. Datla requires demographic knowledge in or with its graph. Dang’s meta-model includes a context node linked to both a disease node and a symptom node, as well as intermediate feature nodes that carry disease-specific conditional probabilities. Ex. 1008 at ¶¶ [0027]–[0031], FIGS. 1A–1B. Dang’s diagnostic process detects context nodes alongside symptom and test nodes. Ex. 1008 at ¶ [0047]. Dang also provides explicit formulas that break each disease score into feature-level contributions, which makes the ranking more transparent than is the case with Datla’s activation–decay process. Ex. 1008 at ¶¶ [0049]–[0050].

262. Datla tunes its diagnoses “to improve the accuracy of the recommendations relative to the patient scenario of interest.” Ex. 1005 at ¶ [0061]. Dang’s structures let the system account for demographics and feature groupings at the level of each symptom–disease relationship, which serves that goal.

263. Dang’s nodes are ordinary node types connected by edges, and Datla’s graph is already a graph of nodes and weighted edges. Ex. 1005 at ¶¶ [0031]–[0032]; Ex. 1008 at ¶¶ [0027], [0047]. Adding node types and storing values as node properties were routine graph-database design choices. The change leaves Datla’s construction and traversal intact.

2. Independent Claim 15

264. Each element of independent claim 15 of the ‘882 patent is disclosed or rendered obvious by the combination of Datla, Goldberg, Dang, and Gnanasambandam. I address each element below.

a. Element [15pre]

265. Element [15pre] recites: “A computer system for determining a state of a patient, said computer system comprising.”

266. As discussed in my analysis of claim 1, *supra* § X.A.2, Datla discloses a computer system for determining a state of a patient. I incorporate that analysis by reference here. Datla is directed to “automated methods and systems to provide a clinical diagnosis of a patient’s symptoms.” Ex. 1005 at ¶ [0001]. The system accepts patient data, processes it, and provides diagnostic output—thereby determining a state of a patient.

b. Element [15a]

267. Element [15a] recites: “one or more processors; and memory that stores instructions, wherein the instructions, when executed by the one or more processors, cause the one or more processors to.”

268. As discussed in my analysis of element [1a], *supra* § X.A.2.b, Datla discloses one or more processors and memory storing instructions. Ex. 1005 at ¶ [0079]; FIG. 5. I incorporate that analysis by reference here. As discussed below, in the combination, the memory stores instructions that cause the processor(s) to perform the recited functionality in claim 15.

c. Element [15b]

269. Element [15b] recites: “extract from a plurality of medical evidence sources a plurality of diagnostic factors that are indicative of one or more of a plurality of medical conditions.”

270. Claim 15 uses the phrase “extract from” rather than claim 1’s “receive electronic text in” and “retrieve.” As discussed in my analysis of elements [1b]–[1d], *supra* §§ X.A.2.c–X.A.2.e, Datla discloses extracting diagnostic factors from medical evidence sources and identifies diagnostic factors indicative of medical conditions. Ex. 1005 at ¶¶ [0034], [0054]. I incorporate my analysis of elements [1b]–[1d] by reference here.

d. Element [15c]

271. Element [15c] recites: “wherein at least a subset of said medical evidence sources includes a published journal and wherein said medical evidence sources provide different taxonomies that arrange one or more of said diagnostic factors and/or one or more of said medical conditions in different hierarchies, and wherein a first diagnostic factor is discussed in part of said medical evidence sources as being indicative of a first medical condition.”

272. As discussed in my analysis of elements [1b]–[1d], *supra* §§ X.A.2.c–X.A.2.e, Datla discloses that at least a subset of its medical evidence sources includes a published journal. *See* Ex. 1005 at ¶¶ [0030], [0033], [0047]. As discussed in my analysis of element [1d], *supra* § X.A.2.e, Datla, alone or in combination with Goldberg, discloses or renders obvious that said medical evidence sources provide different taxonomies that arrange diagnostic factors and/or medical conditions in different hierarchies. *See* Ex. 1005 at ¶ [0054] (system “may” map topical keywords “to categories represented in one or more ontologies, including but not limited to the following controlled clinical ontologies: SNOMED CT for diagnoses, LOINC for tests, and/or RxNorm for treatments”). The indicative relationship between diagnostic factors and medical conditions is established through Datla’s knowledge graph construction. I incorporate by reference my analysis of elements [1b]–[1d] here.

e. Element [15d]

273. Element [15d] recites: “create a graph database cross linking said one or more diagnostic factors and said one or more medical conditions that are arranged in different hierarchies in said medical evidence sources.”

274. As discussed in my analysis of element [1e], *supra* § X.A.2.f, Datla discloses creating a graph database cross-linking diagnostic factors and medical conditions arranged in different hierarchies. Datla’s system builds a knowledge graph with nodes and weighted edges representing clinical concepts and their relationships. Ex. 1005 at ¶¶ [0030]–[0031]. I incorporate that analysis by reference here.

f. Element [15e]

275. Element [15e] recites: “store each of said diagnostic factors and said medical conditions that are cross linked as its own node in said graph database, said one or more medical conditions being linked to said diagnostic factors, wherein each of said diagnostic factor nodes do not include information specific to said medical conditions.”

276. This element is substantively the same as Elements [1f] and [1g] of claim 1. I incorporate my analysis of those elements by reference. *Supra* §§ X.A.2.g-X.A.2.h. I address the differences in wording below.

277. First, element [15e] requires that each diagnostic factor and medical condition be stored as “its own node,” where claim 1 requires “separate entries.” Datla expressly describes its knowledge graph 310 as “a tree-like structure with a plurality of nodes connected by one or more edges,” in which “[e]ach root node of the graph is a symptom, and the remaining nodes are conditions, diagnoses, tests, procedures, medications, or other clinical concepts.” Ex. 1005 at ¶ [0031]. As I explained for dependent claim 3, each clinical concept in Datla’s graph, including each diagnostic factor and each medical condition, is stored as its own node. *Supra* § X.A.4.

278. Second, element [15e] requires that “said one or more medical conditions [be] linked to said diagnostic factors.” In Datla, “[a]n edge is a relationship between two nodes,” and a symptom such as fever “will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios.” Ex. 1005 at ¶ [0031]. Datla’s condition nodes are therefore linked to its diagnostic factor nodes. The same is true in Element [15f], in which a demographics node sits on the linking path between a condition node and a factor node.

279. Third, element [15e] applies the negative limitation to “each of said diagnostic factor nodes,” not only to a “first entry.” My analysis of element [1g] applies equally to every diagnostic factor node in Datla. Datla does not describe a diagnostic factor node that stores condition-specific information. Instead, Datla

places the relationship between two nodes on the edge connecting them: “edges between the nodes are also weighted based on a relationship between the nodes,” such as “the frequency with which the two nodes are associated within a source or a corpus.” Ex. 1005 at ¶ [0032]. That design holds for every symptom node in the graph, because the same symptom node connects to many conditions through different weighted edges. *Id.* at ¶ [0031]. Placing condition-specific information in one symptom node would be incompatible with that node serving “many patient scenarios.” *Id.*

280. My analysis for element [15f] reinforces this conclusion. Demographic information relevant to a factor in the context of a condition is stored in a separate context/demographics node between the condition node and the factor node, not in the factor node. *Infra* § X.D.2.g. That is the same arrangement the ‘882 patent describes for keeping EDFs condition-independent: demographics “may be held separate from the EDF in a demographics node, thus also ensuring that the EDF remains independent and applicable to any of a number of conditions.” Ex. 1001 at 11:59–62. A POSITA implementing the combination would therefore keep each diagnostic factor node free of condition-specific information.

281. Accordingly, Datla, alone and in combination with Gnanasambandam, Dang, and Goldberg, discloses or renders obvious Element [15e].

g. Element [15f]

282. Element [15f] recites: “for at least one of said diagnostic factors, store demographics required to indicate one of said medical conditions on the basis of said at least one diagnostic factor in a demographics node located between said one of said medical condition nodes and a node representing said at least one diagnostic factor.”

283. This claim element requires a “demographics node.” The specification explains:

Each EDF may also have associated a variety of attributes associated with particular conditions such as: age and sex restriction (i.e., if the factor is only applicable to persons of a particular age and only to male or female patients. These condition-specific attributes are held in a demographics node . . . unique to the condition and the EDF, but separate from the EDF.

Ex. 1001 at 10:53–59; *see also id.* at 11:60–64 (“demographics that link the factor to a specific condition (e.g., age, sex,) may be held separate from the EDF in a demographics node, thus also ensuring that the EDF remains independent and applicable to any of a number of conditions.”). The specification also discloses:

Demographics node 374 includes demographics of EDF node 376 which must be true for that factor in the context of condition 370. By way of example, if the factor of EDF node 376 is only present in certain genders, or only below or above certain ages in the context of condition 370 (i.e., the factor can only indicate the condition if the patient meets the demographics) and that information is held in node 374.

Ex. 1001 at 21:14–20.

284. Datla’s user interface is “configured to receive natural language input from a user, the input comprising information about at least one patient symptom and at least one demographic parameter about the patient.” Ex. 1005 at claim 1. At step 114, the information provided “contains information about at least one patient symptom and at least one demographic parameter for the patient.” *Id.* at ¶ [0050]. Datla explains that this “may be age, height, weight, medical background, sex, or any of a wide variety of other demographic information.” *Id.* at ¶ [0051]. Datla “extracts at least one patient symptom from the received information and at least one demographic parameter for the patient from the received information.” *Id.* at ¶ [0052]; *see also id.* at FIG. 1 (step 116), FIG. 3 (demographics 318).

285. Datla collects this information because demographics affect which diagnoses are likely. According to Datla, “the demographic information obtained from the clinical narrative is leveraged to fine-tune the ranking. For example, if a disease is not common for a demographic, its rank is lowered.” Ex. 1005 at ¶ [0061]; *see also id.* at ¶ [0072] (describing element 334). Datla also states that “the possible diagnoses are tuned based on epidemiology to improve the accuracy of the recommendations relative to the patient scenario.” *Id.* at ¶ [0006].

286. A POSITA would have understood that a patient’s demographic parameters have no diagnostic value on their own. A patient’s age or sex matters only when it is compared with stored medical knowledge about how demographics

relate to medical conditions. Datla's own example shows this. The rule that "if a disease is not common for a demographic, its rank is lowered" can be applied only if the system already holds information about the demographics in which each disease is, or is not, common.

287. Datla identifies the knowledge graph as the place where that comparison happens: "the extracted one or more patient symptoms and the extracted one or more pieces of demographic information are utilized to query the knowledge-graph 310 and provide one or more medical conditions, diagnoses, treatment plans, or testing plans." Ex. 1005 at ¶ [0076]. Datla likewise teaches that candidate articles from the Wikipedia clinical knowledge base can be "filtered using various criteria e.g., location, gender, match with topical keywords etc." *Id.* at ¶ [0037].

288. A POSITA reading Datla would therefore have understood or found obvious that Datla's knowledge graph must contain, or be associated with, demographics required to indicate one of said medical conditions on the basis of said at least one diagnostic factor. Only then could a demographic parameter extracted from the patient's record be matched against the knowledge graph and affect the resulting diagnosis.

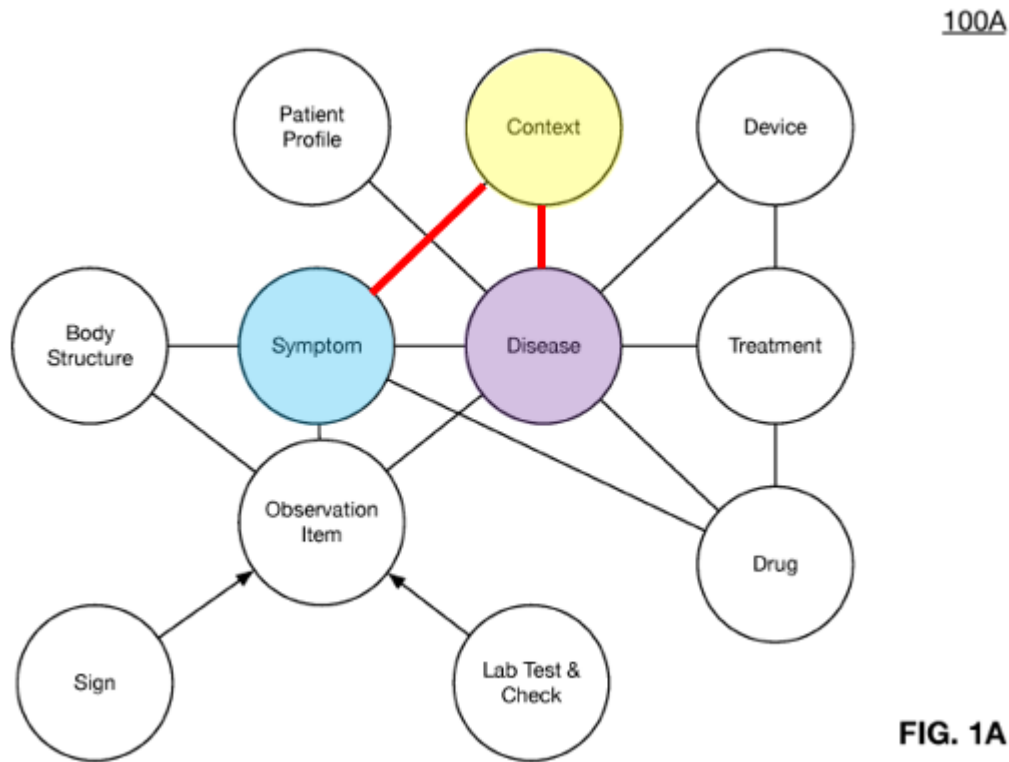
289. Datla does not specify how demographics are represented within the knowledge graph. For example, Datla does not expressly describe or illustrate storing demographics in a separate node. A POSITA implementing Datla's teaching

that demographic information is “utilized to query the knowledge-graph 310” would therefore have looked to known ways of representing demographic knowledge in a clinical knowledge graph. Organizing the nodes within a knowledge graph involves routine design considerations and skill. Constructing a knowledge graph containing a separate demographic node as claimed would have been routine for a POSITA.

290. Dang discloses a knowledge graph architecture that provides exactly such a structure: a context node linked to both a disease node and a symptom node. Dang describes generating a “symptom–disease knowledge graph by: receiving as input one or more electronic documents constituting medical guidelines; extracting a set of diseases and symptoms from the medical guidelines . . . and generating models of the relations among symptoms and diseases based on the extracted information.” Ex. 1008 at ¶ [0025].

291. Dang’s FIG. 1A shows the “symptom–disease meta-model knowledge graph 100A,” which “includes a set of nodes and edges; the nodes include a disease node, a symptom node, a body structure node, a context node, a device node, a treatment node, a drug node, an observation item node, a sign node, a lab test and check node, and a patient profile node.” Ex. 1008 at ¶ [0027]. Dang explains that “[t]he nodes may be connected according to any known method in the art where, using natural language processing and other techniques, relations are identified as between the entities or concepts denoted by two nodes in the graph.” *Id.*

292. As shown in the annotated version of Dang's FIG. 1A below, the Context node is connected by one edge to the Disease node and by another edge to the Symptom node.



Ex. 1008 at FIG. 1A (annotated)

293. The Context node is on the path of edges that links the Disease node to the Symptom node. A POSITA would have understood the Context node to be “located between” the Disease node and the Symptom node, as element [15f] requires. That understanding is consistent with how the ‘882 patent itself uses the term. In FIG. 4A of the ‘882 patent, condition 300 is linked to factor group 304, which in turn is linked to demographics nodes 306 and 308 and then to EDF nodes

310 and 312. Ex. 1001 at FIG. 4A. The ‘882 patent nonetheless describes the demographics node as a node “in between the EDF node and the condition.” *Id.* at 25:25–30. A node is thus “between” a condition and a diagnostic factor under the ‘882 patent when it lies on the linking path between them. It need not be the only link between them, and it need not be directly adjacent to both.

294. A POSITA would have understood that Dang’s Context node’s placement reflects its function. Because the Context node is linked to both the Disease node and the Symptom node, a POSITA would have understood or found obvious that it represents contextual information bearing on the relationship between the disease and the symptom, that is, the circumstances in which the symptom is associated with the disease. Dang’s diagnostic process confirms that context nodes take part in diagnosis alongside symptom nodes: the “graph-based diagnosis process 500 detects factual nodes (for example, symptom nodes, test nodes, context nodes).” Ex. 1008 at ¶ [0047]. Dang therefore teaches a knowledge graph in which contextual information is stored in its own node, separate from the symptom node, and placed between the symptom and the disease.

295. As explained above, a POSITA would have understood that Datla’s knowledge graph contains, or is associated with, medical knowledge relating demographics to medical conditions. Otherwise, Datla’s extracted demographic parameters could not be “utilized to query the knowledge-graph 310,” Ex. 1005 at ¶

[0076], or used to lower the rank of a disease that “is not common for a demographic,” *id.* at ¶ [0061].

296. Datla does not specify how that demographic knowledge is represented in the knowledge graph. In my opinion, it would have been obvious to a POSITA to store Datla’s demographic knowledge in a node like Dang’s Context node, which Dang places between a Disease node and a Symptom node. Ex. 1008 at FIG. 1A, ¶ [0027]. Dang shows that contextual information bearing on a disease–symptom relationship can be held in a node of its own, separate from the symptom node, and used in diagnosis. Dang’s “graph-based diagnosis process 500 detects factual nodes (for example, symptom nodes, test nodes, context nodes).” *Id.* at ¶ [0047]. A POSITA would have recognized that a patient’s age or sex is exactly this kind of contextual information. Dang itself treats a patient’s age as diagnostically relevant in its example of a patient who “is 60 years old.” *Id.* at ¶ [0036].

297. Goldberg corroborates that a POSITA would have understood demographic information to be clinical knowledge stored in a diagnostic knowledge graph, not merely patient input. Goldberg describes a curation mechanism using “a templates based editor supporting data entry. Such templates are designed to support different types of clinical facts e.g. etiological, demographic, causal relations between disorders, findings and complications, risk factors, etc.” Ex. 1006 at 13:19–22. Goldberg’s diagnostic application then uses that knowledge base “to create a

differential diagnosis list based on inputs entered for a specific patient (e.g. demographics, history, symptoms, signs, lab results, etc.).” *Id.* at 8:14–16. Goldberg thus confirms the understanding I described above for Datla. Patient demographics are an input to diagnosis because they are compared with demographic knowledge stored in the knowledge graph.

298. Goldberg also shows why a POSITA would have stored the demographics in a node between a symptom and a condition, as in Dang. Goldberg teaches that demographic and subpopulation qualifications can be specific to the relationship between a clinical feature and a disease. In Goldberg’s FIG. 11 model, the legend identifies a “Qualified feature array (describes a subpopulation),” and the rules linking concepts include “Complication, association, sign, symptom, lab, risk factor.” Ex. 1006 at FIG. 11. Goldberg explains that its figure “shows how any concept can be qualified, and how any concepts can have different types of relations with other such qualified concepts,” and that its models “support the different types of probabilistic rules which are described in medical literature, such as complications, symptoms, lab results, risk factors, etc.” *Id.* at 10:35–46, 10:47–50.

299. Goldberg’s FIG. 4 shows that the diagnostic significance of a demographic qualifier depends on the disease involved. The risk factor “Periods of life (qualifier value):ADULTHOOD” carries a likelihood of 30–40 for celiac disease and 40–50 for Crohn’s disease. Ex. 1006 at FIG. 4.

300. A POSITA would have understood from these teachings that the demographics bearing on a diagnosis are not always a fixed property of a disease as a whole. They can depend on which clinical feature is present and which disease that feature is being evaluated against. This is why a POSITA would have stored the demographics in Dang's Context node. Datla's own example is framed at the level of a disease ("if a disease is not common for a demographic, its rank is lowered"). Ex. 1005 at ¶ [0061]. Goldberg teaches that demographic and subpopulation qualifications may attach to a particular feature's relationship with a particular disease. A node linked to the disease alone could not capture that distinction. Dang's Context node is linked to both the Disease node and the Symptom node, so that it can.

301. A POSITA implementing Datla's demographic comparison in light of Goldberg would therefore have found it obvious to store, in a Context node like Dang's, the demographics required for the linked symptom to indicate the linked disease. The result is a demographics node located between a medical condition node and a node representing a diagnostic factor that stores "demographics required to indicate one of said medical conditions on the basis of said at least one diagnostic factor," as element [15f] requires.

302. Zhong further corroborates that a POSITA would have understood demographics to warrant their own node in a diagnostic knowledge graph. Zhong

describes a diagnostic knowledge graph “where a node (e.g., node 210) in the graph is an entity, such as a symptom, a dimension, a demographic, and a disease, and an edge (e.g., edge 220) in the graph represents a relationship between two entities.” Ex. 1009 at 3:20–25. Zhong also recognizes that demographics can determine whether a symptom is relevant to a patient’s diagnosis: “if a symptom relates to males only but the patient is female, that selected candidate [symptom] has a conflict and is disregarded.” *Id.* at 4:18–20; *see also id.* at 6:4–5 (“symptoms related to demographics that are inconsistent with the patient may be discarded”). Zhong thus confirms that representing demographics as a distinct node in a diagnostic knowledge graph, separate from symptom and disease nodes, was known before the ‘882 patent.

303. A POSITA would have been motivated to make this modification for at least three reasons.

304. First, a POSITA would understand Datla requires demographic knowledge in or with its knowledge graph so that extracted demographics can be used to query the graph and adjust diagnoses, but Datla does not say how that knowledge is represented. Ex. 1005 at ¶¶ [0052], [0061], [0076]. Dang supplies a known node type for contextual information within a symptom–disease knowledge graph. Using it for Datla’s demographic knowledge is the application of a known structure to its known purpose. Ex. 1008 at FIG. 1A, ¶ [0047].

305. Second, the modification would improve diagnostic accuracy, which Datla expressly seeks. Datla tunes its diagnoses by demographics “to improve the accuracy of the recommendations relative to the patient scenario of interest.” Ex. 1005 at ¶ [0061]. Goldberg teaches that a demographic’s diagnostic significance can differ between diseases and depend on the clinical feature involved. Ex. 1006 at FIG. 4, FIG. 11. Storing demographics for each symptom–disease relationship lets the system account for those differences. It also lets the demographic comparison be made as the knowledge graph is traversed, not only after a ranked list has been produced.

306. Third, the modification requires no redesign. Datla’s knowledge graph is already a graph of nodes and weighted edges built from a corpus of medical information. Ex. 1005 at ¶¶ [0031]–[0032]. Dang’s Context node is simply another node type connected by edges, and Dang’s diagnostic process already detects context nodes alongside symptom nodes. Ex. 1008 at ¶¶ [0027], [0047]. Adding such nodes to Datla’s knowledge graph and populating them with demographic criteria such as an age range or sex would have been a routine design choice.

307. A POSITA would have had a reasonable expectation of success. Datla, Dang, and Goldberg each describe a computer-implemented clinical knowledge graph derived from medical literature and used for diagnosis. Ex. 1005 at ¶ [0006]; Ex. 1008 at ¶ [0025]; Ex. 1006 at 6:10–34. Storing an attribute value such as an age

range or sex in a graph node, and comparing it with a patient's extracted demographics, was a routine graph-database technique. Datla already extracts the patient demographics to be compared. Ex. 1005 at ¶ [0052]. Goldberg shows that demographic clinical facts and subpopulation-qualified rules can be represented in a clinical knowledge graph and used for diagnosis. Zhong confirms a POSITA's understanding that demographic nodes were used in diagnostic knowledge graphs.

308. Accordingly, in my opinion the combination of Datla, Gnanasambandam, Dang, and Goldberg renders obvious element [15f].

h. Element [15g]

309. Element [15g] recites: “establish one or more application programming interfaces (APIs) for said graph database, said one or more APIs allowing an application to integrate with said graph database to access said diagnostic factors and said medical conditions that are linked in said graph database in order to generate a healthcare determination.”

310. Element [15g] is substantively the same as element [1h] of claim 1, except that it recites access to “said diagnostic factors and said medical conditions that are linked in said graph database” rather than “said separate entries that are linked in said graph database.” For the reasons discussed in my analysis of element [1h], *supra* § X.A.2.i, Gnanasambandam's APIs allow an application to integrate with the knowledge graph to access the linked diagnostic factors and medical

conditions, which in the combined system are stored as nodes, in order to generate a healthcare determination.

i. Element [15h]

311. Element [15h] recites: “match patient data from a patient medical chart to at least one of said diagnostic factor nodes and to said demographics node, in order to display said healthcare determination that said patient has one of said medical conditions.”

312. **Matching patient data to diagnostic factor nodes.** Datla discloses matching patient data to diagnostic factor nodes in its knowledge graph. Datla’s system extracts patient symptoms from the clinical information it receives and queries the knowledge graph with them. Ex. 1005 at ¶¶ [0052], [0055]. Datla explains that querying the knowledge graph includes “assigning, to a node of the knowledge graph comprising the extracted one or more patient symptoms, the assigned weight as an activation weight.” *Id.* at ¶ [0055]. Datla also generates “an initial scattered forest . . . with the start points being root nodes associated with the extracted and weighted one or more symptoms.” *Id.* at ¶ [0067]. When “the TF-IDF weighted clinical concepts extracted from the clinical narrative are used to query the knowledge graph, some embodiments perform all one-hop expansion of the symptom nodes.” *Id.* at ¶ [0057]. A POSITA would have understood that identifying the node “comprising” the patient’s extracted symptom is matching the patient’s data

to that diagnostic factor node. As explained for elements [15b]–[15e], Datla’s symptom nodes are the claimed diagnostic factor nodes. *Supra* §§ X.D.2.c–X.D.2.f.

313. **Patient data from a patient medical chart.** The ‘882 patent uses “patient medical chart” to refer to the patient’s record in the electronic health record (“EHR”) or electronic medical record (“EMR”) system. The ‘882 patent describes a clinician who uses the EHR system “to open the patient’s electronic chart,” after which the platform “downloads the patient’s medical history from his or her chart.” That history includes “any clinical notes that a clinician had entered previously or has just entered, including simple notes, SOAP notes, structured column notes, consultant notes, etc.” Ex. 1001 at 22:53–67; *see also id.* at 6:32–34 (“the patient’s medical chart in the EMR system”).

314. Datla processes chart content. Datla identifies “the most probable diagnosis given a clinical narrative, such as a summary, description, or note.” Ex. 1005 at ¶ [0057]; *see also id.* at ¶ [0034] (keywords identified from “clinical notes”). Datla’s evaluation topics include “actual admission notes derived from MIMIC-III.” *Id.* at ¶ [0048]. Datla also describes MIMIC-II data containing “comprehensive clinical data obtained from hospital medical information systems,” from which diagnoses are separated “from each medical record.” *Id.* at ¶ [0039]. A POSITA would have understood that the clinical notes Datla analyzes are the content of a patient’s medical chart. At a minimum, it would have been obvious to take those

notes directly from the patient’s electronic chart, where clinicians record them, rather than require the same information to be entered again.

315. Dang expressly teaches using patient data from the patient’s EMR with nodes of its knowledge graph. Dang describes “a patient whose electronic medical record (EMR) for a given visit to a doctor indicates (or confirms) that the patient suffers from a runny nose, a fever, has a body temperature of 38° C. (this may be indicated, for example, via the checks and tests nodes depicted in symptom–disease meta-model knowledge graph 100A/100B), and is 60 years old (this may be indicated, for example, via the patient profile node of symptom–disease meta-model knowledge graph 100A/100B).” Ex. 1008 at ¶ [0036]. Dang’s “graph-based diagnosis process 500 detects factual nodes (for example, symptom nodes, test nodes, context nodes)” using data from “patient complaint records,” and it may do so “using semantic analysis and matching.” *Id.* at ¶ [0047]. Dang measures “a similarity between an input text [and] one or more nodes in the knowledge graph data structure,” yielding a confidence score. *Id.* at ¶ [0009]; *see also id.* at ¶ [0047] (“Confidence of fact = similarity (input->node)”).

316. Gnanasambandam also teaches that “the processing device of a server may receive an electronic medical record including notes pertaining to a patient.” Ex. 1007 at 61:16–18; *see also id.* at 56:62–64. Gnanasambandam processes those notes to obtain indicia and identifies “a similarity between the indicia and a

knowledge graph.” *Id.* at 57:24–25, 61:30–34. For example, “‘high blood pressure’ may be extracted as indicia from the sentence ‘Patient X has high blood pressure’, and ‘high blood pressure’ is a health artifact at a node in the knowledge graph 500.” *Id.* at 57:41–45, 61:36–40.

317. A POSITA would have been motivated to use patient data from the patient’s medical chart in the combined system. Datla already analyzes clinical notes and narratives. Ex. 1005 at ¶¶ [0034], [0057]. Dang and Gnanasambandam show that the patient’s EMR was a known source of that data for knowledge-graph–based diagnosis. Ex. 1008 at ¶ [0036]; Ex. 1007 at 56:62–64, 61:16–18. Obtaining the patient data directly from the chart would reduce manual entry and fit the system into the clinician’s existing workflow. That is a predictable use of a known data source for its known purpose.

318. **Matching patient data to the demographics node.** The combined system also matches patient data to the demographics node. As explained for element [15f], it would have been obvious to store in a Context node like Dang’s the demographics required for a linked symptom to indicate a linked disease. *Supra* § X.D.2.g. The ‘882 patent explains that demographics nodes are used by comparing the patient’s demographics with the demographics stored in the node: the demographics nodes “corresponding to matched EDF nodes are taken into account when reasoning occurs to determine if the patient has the particular condition related

to that EDF node.” Ex. 1001 at 24:40–44. Datla extracts the patient’s demographic parameters, such as “age, height, weight, medical background, sex,” and uses them to “query the knowledge-graph 310.” Ex. 1005 at ¶¶ [0051], [0076]. Dang’s diagnostic process detects context nodes, alongside symptom and test nodes, as factual nodes matched from patient records. Ex. 1008 at ¶ [0047]. In the combined system, the patient’s demographics extracted from the patient medical chart would be matched to the demographics stored in the demographics node located between the medical condition node and the matched diagnostic factor node.

319. Goldberg corroborates this understanding. Goldberg’s diagnostic method receives patient health data and evaluates it “to determine a set of nodes in the plurality of nodes that represents the health data.” Ex. 1006 at claim 1. Goldberg’s application creates “a differential diagnosis list based on inputs entered for a specific patient (e.g. demographics, history, symptoms, signs, lab results, etc.).” *Id.* at 8:14–16. Goldberg thus confirms that a POSITA would have understood patient demographics and history to be patient data that is compared with stored knowledge to generate a diagnosis.

320. **Displaying the healthcare determination.** The combined system displays the healthcare determination that the patient has one of the medical conditions. Datla identifies “the most probable diagnosis” from the knowledge graph, and “one or more top-ranked diseases and medical conditions can be extracted

from the knowledge graph.” Ex. 1005 at ¶¶ [0057], [0059]. “[T]he ranked medical conditions and/or diagnosis are provided to the clinician,” and they “can be provided to the user via any user interface that allows information to be conveyed, such as a speaker or screen.” *Id.* at ¶ [0062]. Datla further teaches that “the highest ranked one or more medical conditions, diagnoses, treatments, and/or tests for the patient is provided to the user.” *Id.* at ¶ [0011]. Dang likewise outputs “the top disease list as a suspected disease list,” and the output “may be to a user via a notification including via a graphical user interface.” Ex. 1008 at ¶ [0056]. A POSITA would have understood the display of Datla’s highest-ranked, “most probable” diagnosis to be a display of a healthcare determination that the patient has that medical condition.

321. Finally, “said healthcare determination” refers to the healthcare determination generated in element [15g] by an application that accesses the graph database through the established APIs. As explained for element [15g], Gnanasambandam teaches APIs that allow applications to access the knowledge graph. *Supra* § X.D.2.h. In the combined Datla–Gnanasambandam system, the diagnosis that Datla displays is the healthcare determination generated through those APIs.

322. Accordingly, the combination of Datla, Gnanasambandam, Dang, and Goldberg discloses or renders obvious element [15h].

3. Dependent Claim 16

323. Dependent claim 16 recites: “A computer system as recited in claim 15, wherein the instructions, when executed, further cause the one or more processors to: determine a rank of said likelihood that said patient has one of said medical conditions by reference to a node intermediate between one of said medical condition nodes and said diagnostic factor nodes that holds said rank of said diagnostic factors.” I incorporate by reference my discussion of independent claim 15 above. *Supra* § X.D.2.

324. At the outset, I note that claim 16 recites “said likelihood that said patient has one of said medical conditions.” Neither claim 16 nor claim 15 introduces any “likelihood.” The term “likelihood” does not appear in claim 15. The language in claim 15 closest to claim 16’s recited “said likelihood that said patient has one of said medical conditions” is element [15h]’s “healthcare determination that said patient has one of said medical conditions.” Both phrases concern whether “said patient has one of said medical conditions,” and no other language in claim 15 addresses the patient’s condition. For purposes of my analysis, I have interpreted “said likelihood that said patient has one of said medical conditions” as the likelihood underlying the healthcare determination recited in element [15h], that is, the likelihood that the patient has a particular medical condition.

325. I also note that claim 16 uses the term “rank” inconsistently. Claim 16 first recites determining “a rank of said likelihood that said patient has one of said medical conditions,” which ranks medical conditions for a particular patient. Claim 16 then requires an intermediate node “that holds said rank of said diagnostic factors,” which ranks diagnostic factors. Grammatically, the second “said rank” refers back to the “rank” introduced earlier in the claim. But the two instances of “rank” describe different quantities (i.e., the first instance of “rank” refers to the likelihood that the patient has one of the medical conditions and the second instance of “rank” refers to diagnostic factors). I have interpreted “determine a rank of said likelihood” as ranking the patient’s possible medical conditions by that likelihood, consistent with the specification’s description of “the likely, ranked conditions of the patient.” Ex. 1001 at 9:8–9. I have interpreted “said rank of said diagnostic factors” as the condition-specific rank of each linked diagnostic factor held in the intermediate node, consistent with the specification’s factor group nodes. *Id.* at 20:60–21:8.

326. While I have offered interpretations of these antecedent basis issues, my opinions do not depend on resolving these issues in a particular way. As I explain below, the combination of Datla, Goldberg, Gnanasambandam, and Dang satisfies claim 16 whether “said rank” refers to a patient-level ranking of conditions, a factor-level rank held in the intermediate node, or both. Datla identifies “a ranked list of

one or more medical conditions . . . for the patient.” Ex. 1005 at ¶ [0007]. Dang determines disease confidence from values computed at its intermediate feature nodes, calculates a probability rate for each disease, and sorts the diseases by that rate. Ex. 1008 at ¶¶ [0049]–[0050], [0056]. In the combined system, the intermediate feature node holds the condition-specific rank of the linked diagnostic factors.

327. Datla determines a rank of the likelihood that the patient has a medical condition. Specifically, Datla identifies “a ranked list of one or more medical conditions . . . for the patient from the diagnosis graph.” Ex. 1005 at ¶ [0007]. Datla holds the condition-specific strength of each symptom–condition relationship on the weighted edge between the two nodes. *Id.* at ¶ [0032]. Placing that value on an edge is a routine implementation/design choice. Knowledge graphs are commonly built either as labeled property graphs, where edges carry properties as in Datla, or as triple stores, where every item of information is a triple that links two nodes to each other via a single relation. A POSITA would have understood that a value carried on an edge in a labeled property graph can equally be held in a node placed on that link in a triple store.

328. Dang discloses such an intermediate node. Dang’s knowledge graph 100B “includes the following three sets of nodes: a set of factual nodes (symptom nodes, test nodes and context nodes), a set of feature nodes, and a set of disease nodes.” Ex. 1008 at ¶ [0028]. Symptom nodes connect to “one or more features.” *Id.*

at ¶ [0030]. Each disease node “is connected to one or more feature nodes,” and each connection “may have a corresponding conditional probability P ,” where $P(f_x|d_n)$ “indicates, for a pair of disease and feature (d_n , f_x), the likelihood of the feature f_x occurring when the disease d_n [is] observed.” *Id.* at ¶ [0031].

329. As shown in FIG. 1B, each feature node (annotated in yellow) lies on the path between a disease node and the symptom nodes. Ex. 1008 at FIG. 1B. The feature nodes are also disease-specific. Dang defines $F(d)$ as “a set of features of a disease d .” *Id.* at ¶ [0032].

330. Dang determines the likelihood of each disease by reference to these feature nodes. Dang’s diagnosis process detects factual nodes, infers evidential nodes, and then “infers (step 506) disease feature nodes from the evidential nodes,” computing “Confidence of feature = Prod(Confidence of evidence * weight).” Ex. 1008 at ¶¶ [0047]–[0049]. It then “infers (step 508) possible disease features from the feature nodes inferred (step 506),” using “a confidence value calculated based on feature contributions.” *Id.* at ¶ [0050]. Dang then calculates a probability rate for each disease, sorts the possible diseases by that rate, and outputs the top diseases. *Id.* at ¶ [0056]. Dang therefore determines a rank of the likelihood that the patient has a disease by reference to intermediate feature nodes.

331. Dang associates $P(f|d)$ with the connection between a disease and a feature. Ex. 1008 at ¶ [0031]. It would have been obvious to hold that value, which

is the rank of the diagnostic factors in the context of that disease, in the feature node itself. Because each feature node belongs to the feature set of a particular disease (*id.* at ¶ [0032]), a value stored in the feature node is already specific to that disease. Moving it from the connection into the node changes neither its meaning nor how it is used. This is the edge-versus-node design choice I described for Datla above. In the combined system, the intermediate feature node would hold the rank of the linked diagnostic factors for that condition. The system would determine the rank of the likelihood that the patient has that condition by reference to that node, as Dang's steps 506–508 do. Ex. 1008 at ¶¶ [0049]–[0050].

332. A POSITA would have been motivated to add Dang's intermediate feature nodes to the claim 15 combination for at least three reasons. First, Dang is already part of that combination, for the same field-of-endeavor reasons I gave for element [15f]. *Supra* § X.D.2.g. Second, Dang provides explicit formulas that break each disease score into its feature-level contributions. Ex. 1008 at ¶¶ [0049]–[0050]. That makes the ranking more transparent than Datla's activation-decay process, which does not expose a discrete scoring formula. Ex. 1005 at ¶¶ [0057], [0069]–[0071]. It also mirrors how clinicians combine individual findings into clinical features before weighing a diagnosis. Third, holding condition-specific ranks in an intermediate node keeps each diagnostic factor node condition-independent, as element [15e] requires, just as Datla's edges do. *Supra* § X.D.2.f.

333. A POSITA would have reasonably expected success. Dang’s context nodes, which I relied on for element [15f], already feed into its feature nodes in FIG. 1B. Ex. 1008 at FIG. 1B. Dang’s feature nodes therefore sit in the same graph as the demographics node. Storing a numerical value as a node property, and reading it during traversal, was a routine graph-database operation.

4. Dependent Claim 17

334. Dependent claim 17 recites: “A computer system as recited in claim 15, wherein the instructions, when executed, further cause the one or more processors to: perform said extract said plurality of diagnostic factors using natural language processing (NLP) that is initialized using existing diagnostic factor nodes of said graph database.” I incorporate by reference my discussion of independent claim 15 above. *Supra* § X.D.2.

335. Datla builds its knowledge graph by processing the text of its medical corpus. When it uses an online source with a “clinical medicine” master category, “all pages including and under the master top category are analyzed . . . with information being extracted from each page,” and relations between pages “may be otherwise constructed or extracted.” Ex. 1005 at ¶ [0030]. Datla weights graph edges by “the frequency with which the two nodes are associated within a source or a corpus.” *Id.* at ¶ [0032]. It also weights patient symptoms by “log inverse frequency of their usage in the medical corpus utilized to generate the knowledge graph.” *Id.*

at ¶ [0053]. Counting how often clinical concepts appear, and co-appear, in free text requires recognizing those concepts in the text. A POSITA would have understood or found obvious that Datla uses natural language processing for this task.

336. To the extent Datla does not expressly label its construction step as NLP, it would have been obvious to perform that step with Datla’s own NLP engine. Datla already discloses “a natural language processing engine” that “extract[s] keywords related to symptoms, such as lab results, procedures, and/or demographic information.” Ex. 1005 at ¶ [0052]; *see also id.* at ¶¶ [0065], [0081]. Datla also contemplates continuously updating the graph “with new journal articles, new online sources, or other possible sources of new information.” *Id.* at ¶ [0032]. Datla’s corpus includes “medical journals, newspapers, online sources such as Wikipedia, and other sources.” *Id.* at ¶ [0030]. That corpus is unstructured text, and applying the same engine to extract symptoms and conditions from it is the natural way to automate the updates Datla describes. Indeed, as discussed in the Technology Background section, using NLP for this purpose was well known in the art.

337. Dang confirms that NLP-based graph construction was known in exactly this setting. Dang generates its symptom–disease knowledge graph by “receiving as input one or more electronic documents constituting medical guidelines; extracting a set of diseases and symptoms from the medical guidelines . . . and generating models of the relations among symptoms and diseases based on

the extracted information,” and explains that “[t]hese steps may be performed using, for example, electronic natural language processing systems and methods.” Ex. 1008 at ¶ [0025]. Dang further discloses that nodes “may be connected according to any known method in the art where, using natural language processing and other techniques, relations are identified as between the entities or concepts denoted by two nodes in the graph.” *Id.* at ¶ [0027].

338. Dang discloses that the NLP is initialized using existing diagnostic factor nodes. At step 402, Dang’s system “recognizes features in the medical guideline corpus by recognizing or allocating nodes in the symptom–disease knowledge graph.” Ex. 1008 at ¶ [0042]; FIG. 4. Recognizing a feature by reference to an existing node, and allocating a new node only when no existing node is recognized, means that nodes already in the graph are used when the corpus is processed. That is the same thing the ‘882 patent describes as the NLP engine “mak[ing] use of nodes in the graph database 50 when recognizing concepts in an evidence source.” Ex. 1001 at 12:26–28. Dang applies the same principle to input text, measuring “using semantic matching, a similarity between an input text [and] one or more nodes in the knowledge graph data structure.” Ex. 1008 at ¶ [0009]; *see also id.* at ¶ [0047].

339. A POSITA would have been motivated to initialize the NLP used in Datla’s graph construction with the graph’s existing symptom nodes, as Dang

teaches, for at least three reasons. First, Datla updates its graph continuously from new sources. Ex. 1005 at ¶ [0032]. Each update must decide whether a symptom found in new text is already represented by a node or requires a new one. Without that check, the graph would accumulate duplicate nodes for the same symptom, which would split Datla’s frequency-based edge weights (Ex. 1005 at ¶ [0032]) and spread activation across redundant nodes during Datla’s activation-decay cycles (Ex. 1005 at ¶¶ [0055]–[0057]). Recognizing or allocating nodes, as Dang does at step 402, is the conventional solution. Ex. 1008 at ¶ [0042]; FIG. 4. Second, Datla queries the graph by assigning weights to “a node of the knowledge graph comprising the extracted one or more patient symptoms.” Ex. 1005 at ¶ [0055]. Construction that reuses existing symptom nodes keeps the graph’s vocabulary consistent, which improves the element [15h] matching of patient data to graph nodes. Third, Dang is already part of the claim 15 combination. Using Dang’s node recognition during construction is a use of a known technique for its known purpose, and a POSITA would have had a reasonable expectation of success because Dang performs this step on the same kind of symptom–disease graph built from clinical text. Ex. 1008 at ¶¶ [0025], [0042].

5. Dependent Claim 18

340. Dependent claim 18 recites: “A computer system as recited in claim 15 wherein said graph database includes more than twenty thousand nodes.” I

incorporate by reference my discussion of independent claim 15 above. *Supra* § X.D.2.

341. Claim 18 recites the same threshold as claim 12, with two differences: it depends from claim 15 rather than claim 1, and it recites “nodes” rather than “entries.” Neither difference affects my analysis. Datla’s knowledge graph is expressly made up of “a plurality of nodes connected by one or more edges,” Ex. 1005 at ¶ [0031], and my analysis of claim 12 addressed those nodes. *Supra* § X.A.8. Goldberg, whose SNOMED-based knowledge graph I relied on for claim 12, is also part of the claim 15 combination. *Supra* § X.D.2. The context/demographics nodes that the claim 15 combination adds only increase the node count. *Supra* § X.D.2.g (Element [15f]). My routine-scaling and duplication-of-parts analysis for claim 12 applies equally here. *Supra* § X.A.8.

6. Dependent Claim 19

342. Dependent claim 19 recites: “A computer system as recited in claim 15 wherein said graph database includes more than one hundred thousand nodes.” I incorporate by reference my discussion of independent claim 15 above. *Supra* § X.D.2.

343. Claim 19 recites the same threshold as claim 13, and it differs from claim 13 in the same two respects that claim 18 differs from claim 12. For the reasons given for claim 18, neither difference affects my analysis. *Supra* § X.D.5. As I

explained for claim 13, Goldberg’s integration of SNOMED, which Goldberg describes as “covering hundreds of thousands of medical terms,” supplies well over one hundred thousand nodes. Ex. 1006 at 1:54–56; *supra* § X.A.9. Goldberg is part of the claim 15 combination. My routine-scaling analysis for claim 13, and the ‘882 patent’s statement that “an actual graph database will . . . contain many hundreds of thousands of nodes or more,” Ex. 1001 at 27:67–28:3, apply equally here. *Supra* § X.A.9.

7. Dependent Claim 20

344. Dependent claim 20 recites: “A computer system as recited in claim 15 wherein said healthcare determination is a recommendation, an evaluation, a result, or a conclusion.” I incorporate by reference my discussion of independent claim 15 above. *Supra* § X.D.2.

345. Claim 20 recites the same limitation as dependent claim 14, and my analysis of claim 14 applies equally here. *Supra* § X.A.10. The only difference is the claim on which claim 20 depends. In claim 15, “said healthcare determination” is the determination, displayed after the patient data are matched to the graph, “that said patient has one of said medical conditions.” Ex. 1001 at claim 15. As I explained for Element [15h], Datla identifies “the most probable diagnosis given a clinical narrative,” Ex. 1005 at ¶ [0057], and provides “the ranked medical conditions and/or diagnosis” to the clinician, *id.* at ¶ [0062]. *Supra* § X.D.2.i.

346. For the reasons I discussed for claim 14, that output is a “result” and an “evaluation” of the patient’s likely conditions. It is a “conclusion” as to which condition the patient most probably has. It is a “recommendation” because Datla states that the system “can effectively recommend the diagnosis.” *Id.* at ¶ [0061]. Nothing in the additional reference used for claim 15 (Dang) changes the nature of that output. Indeed, Dang’s output of a top disease list to the user as a suspected disease list, Ex. 1008 at ¶ [0056], is likewise a result and a conclusion.

347. Accordingly, the combination of Datla, Gnanasambandam, Dang, and Goldberg renders claim 20 obvious.

8. Dependent Claim 7

348. Dependent claim 7 recites: “A computer system as recited in claim 1, wherein the instructions, when executed, further cause the one or more processors to: receive input from a person via a computing device indicating a demographics corresponding to one of said diagnostic factors, said demographics indicating a type of person to which said one of said diagnostic factors occurs in the context of said one or more medical conditions; and store said demographics in a demographics entry linked between one of said medical condition entries and one of said diagnostic factors entries representing said one of said diagnostic factors.” I incorporate by reference my discussion of independent claim 1 above. *Supra* § X.A.2.

349. **Receiving demographic input from a person via a computing device.** Datla collects demographic parameters from the patient scenario. Its NLP engine is “configured to extract the at least one patient symptom and at least one demographic parameter from the received natural language input.” Ex. 1005 at ¶ [0007], claim 1. Datla also uses demographics to adjust its diagnostic rankings: “if a disease is not common for a demographic, its rank is lowered.” Ex. 1005 at ¶ [0061]. Datla does not describe a person entering demographic knowledge into the graph. Goldberg teaches that step.

350. Goldberg describes a platform that “includes a data curation module, to enable the entry, editing, reviewing etc. of healthcare data from multiple-sources into a structured database.” Ex. 1006 at 13:13–16. Goldberg explains that “a curation mechanism may include using a templates based editor supporting data entry. Such templates are designed to support different types of clinical facts e.g. etiological, demographic, causal relations between disorders, findings and complications, risk factors, etc.” *Id.* at 13:17–22. The person uses a computing device to enter that data. Goldberg’s Editor 320 is “an interface in the form of a mobile app or web interface for data entry by community members.” *Id.* at 7:58–59.

351. Goldberg’s demographic facts are tied to particular diagnostic factors in the context of particular diseases. FIG. 4 shows “a feature by feature comparison for two selected diseases (CROHN’s and CELIAC).” Ex. 1006 at 4:54–58, FIG. 4.

In that figure, the qualifier “Periods of life (qualifier value):ADULTHOOD” has a likelihood of 30–40 for celiac disease and 40–50 for Crohn’s disease. *Id.* at FIG. 4. That qualifier indicates “a type of person to which said one of said diagnostic factors occurs in the context of said one or more medical conditions,” as claim 7 requires.

352. **Storing the demographics in an entry linked between the condition and factor entries.** This limitation corresponds to element [15f]. A “demographics entry linked between” the condition and factor entries is at least as broad as the “demographics node located between” those nodes that [15f] recites. For the reasons given for element [15f] and claim 27, it would have been obvious to store the demographics in a node like Dang’s Context node. That node is linked to both the Disease node and the Symptom node. *Supra* § X.D.2.g; *infra* § X.F.5; Ex. 1008 at ¶ [0027], FIG. 1A.

353. Datla’s rule that a disease’s “rank is lowered” when it “is not common for a demographic” works only if the system already holds demographic knowledge to compare against. Ex. 1005 at ¶ [0061]. Goldberg supplies a known way to get that knowledge into the graph: expert entry through a template-based editor built for “demographic” clinical facts. Ex. 1006 at 13:17–22. Goldberg also confirms that the system compares that stored knowledge with patient inputs. Its application creates “a differential diagnosis list based on inputs entered for a specific patient (e.g. demographics, history, symptoms, signs, lab results, etc.).” *Id.* at 8:14–16. Using

Goldberg's editor to fill Datla's graph with demographic knowledge would have been a known technique used for its known purpose, and a POSITA would have reasonably expected it to work.

E. Ground 5: Claims 21-41 Lack Adequate Written Description

354. Independent claim 21 recites: "wherein said at least one medical evidence journal provides different taxonomies that arrange one or more of said diagnostic factors entries and/or one or more of said medical condition entries in different hierarchies." Independent claim 32 similarly recites: "wherein said at least one medical evidence journal provides different taxonomies that arrange one or more of said known and unknown diagnostic factors and/or one or more of said medical condition entries in different hierarchies."

355. Independent claims 1 and 15 contain a similar limitation: "wherein said medical evidence sources provide different taxonomies that arrange one or more of said diagnostic factors and/or one or more of said medical conditions in different hierarchies." While claims 1 and 15 separately recite "wherein at least a subset of said medical evidence sources includes a published journal," unlike claims 21 and 32, claims 1 and 15 recite that medical evidence sources provide different taxonomies. Claims 21 and 32, by contrast, recite that at least one medical evidence journal provides different taxonomies.

356. Having reviewed the '882 patent specification, I have not found any disclosure that a medical evidence journal, as opposed to a combination of different types of medical evidence sources, itself provides different taxonomies that arrange diagnostic factors and/or medical conditions in different hierarchies as recited in claims 21-41 of the '882 patent.

357. The specification describes processing “medical evidence” from various sources to build the knowledge graph. The specification discusses extracting clinical concepts from electronic text and structuring the concepts as nodes in a graph database. Ex. 1001 at 7:4–21, 8:10–13. The specification refers broadly to “medical evidence sources” but does not describe any journal source as providing multiple different taxonomies.

358. The specification identifies ontologies as the structures that provide taxonomies and hierarchies. The specification states: “It is realized that multiple clinical ontologies exist, including SNOMED-CT, but that these by themselves do not provide the end-to-end computational representation necessary to make evidence and associated administrative guidelines operational. And although SNOMED-CT provides taxonomies of clinical terms (e.g., fever, heart valve replacements, etc.) it can only perform limited reasoning.” Ex. 1001 at 4:13–19. This passage attributes “taxonomies” to a medical ontology, not a medical evidence journal. The specification further states that “[t]he EDFs may also be connected to other

ontologies, including but not limited to SNOMED-CT, for example to provide standardized synonyms.” *Id.* at 10:46–48. These “ontologies,” including SNOMED-CT, LOINC, RxNorm, are the structures that organize clinical concepts into hierarchies. *See id.* at 5:14–19, 8:48–53, 20:31–40.

359. The specification treats journal articles and other narrative text as sources of clinical content from which diagnostic factors and their relationships to medical conditions are extracted. The specification does not describe any journal as itself providing a taxonomy. A typical journal article discusses clinical findings in narrative form. Taxonomies are provided by ontologies and classification systems such as SNOMED-CT and LOINC. The specification’s usage confirms this distinction: when the specification discusses taxonomies, it points to ontologies (Ex. 1001 at 4:13–21); when it discusses journals and other narrative sources, it describes them as sources of clinical content to be extracted and mapped to ontological structures (Ex. 1001 at 7:4–21).

360. Thus, a POSITA reviewing the ’882 patent specification would not find any disclosure of a medical evidence journal that itself provides different taxonomies arranging diagnostic factors and/or medical conditions in different hierarchies. The specification describes journals as narrative evidence sources and describes ontologies (SNOMED-CT, LOINC, RxNorm) as the structures that provide taxonomies and hierarchies. At no point does the specification describe, illustrate,

or suggest that a journal provides taxonomies, much less different taxonomies arranging factors in different hierarchies. Accordingly, in my opinion, claims 21-41 of the '882 patent lack adequate written description under 35 U.S.C. § 112.

F. Ground 6: Claims 21-22 and 26-31 are Obvious Over Datla, Dang, Gnanasambandam, and MacRae

361. In my opinion, claims 21-22 and 26-31 of the '882 patent are obvious over Datla, Dang, Gnanasambandam, and MacRae.

1. Motivation to Combine

362. For the reasons I provided for Grounds 1 and 4, a POSITA would have been motivated to combine Datla, Gnanasambandam, and Dang. *Supra* §§ X.A.1, X.D.1. Those reasons do not depend on Goldberg, and this ground does not rely on Goldberg.

363. MacRae (Ex. 1013) is the kind of source Datla's corpus already includes. Datla builds its graph from "any source of information, including but not limited to medical journals." Ex. 1005 at ¶ [0030]. Datla also weights edges by whether two nodes "appear within . . . the same medical journal." Ex. 1005 at ¶ [0032]. MacRae is a medical journal article published in BMJ Open and it would have been obvious to include it in Datla's corpus. The incorporation of MacRae does not require modification to Datla's system because it is an instance of the journal content Datla expressly contemplates ingesting. A POSITA seeking to populate Datla's graph with reliable symptom–condition relationships would have drawn on

peer-reviewed articles of this kind. Adding a journal article to Datla’s corpus uses Datla’s existing ingestion and NLP processing for their intended purpose. Ex. 1005 at ¶¶ [0030], [0032]. The result is predictable: a graph whose entries originate in, among other sources, a BMJ journal article.

2. Independent Claim 21

364. Each element of independent claim 21 of the ‘882 patent is disclosed or rendered obvious by the combination of Datla, Dang, Gnanasambandam, and MacRae. I address each element below.

a. Element [21pre]

365. Element [21pre] recites: “A computer system for determining a state of a patient, said computer system comprising.”

366. Element [21pre] is the same as element [15pre]. As discussed in my analysis of element [15pre], *supra* § X.D.2.a, Datla discloses a computer system for determining a state of a patient. I incorporate that analysis by reference here.

b. Element [21a]

367. Element [21a] recites: “one or more processors; and memory that stores instructions, wherein the instructions, when executed by the one or more processors, cause the one or more processors to.”

368. As discussed in my analysis of elements [1a] and [15a], *supra* §§ X.A.2.b, X.D.2.b, Datla discloses one or more processors and memory storing

instructions. I incorporate that analysis by reference here. As discussed below, in the combination, the memory stores instructions that cause the processor(s) to perform the recited functionality in claim 21.

c. Element [21b]

369. Element [21b] recites: “receive a portion of a patient electronic health record (EHR) to produce a plurality of diagnostic factors known to be present or absent in said patient.”

370. For the reasons given for element [15h], the combination receives patient data from a patient electronic health record (EHR). *Supra* § X.D.2.i. Datla processes “MIMIC-II discharge notes,” taken from “comprehensive clinical data obtained from hospital medical information systems.” Ex. 1005 at ¶ [0039]. Datla also processes topic “notes” that “are actual admission notes derived from MIMIC-III.” *Id.* at ¶ [0048]. Dang likewise uses a patient’s “electronic medical record (EMR) for a given visit to a doctor” that “indicates (or confirms)” the patient’s symptoms. Ex. 1008 at ¶ [0036].

371. Datla produces diagnostic factors known to be present or absent. Datla’s NLP engine “extracts at least one patient symptom from the received information.” Ex. 1005 at ¶ [0052]. A symptom extracted from a clinical note as affirmatively documented is a factor known to be present. Datla also teaches that its system “may . . . filter out negated clinical concepts in order to give more weight to

positive clinical manifestations.” *Id.* at ¶ [0054]. To filter negated concepts, the system must first determine that a documented factor is negated, that is, known to be absent. The combination therefore produces diagnostic factors known to be present or absent.

d. Element [21c]

372. Element [21c] recites: “match said diagnostic factors to entries in a graph database that represent said diagnostic factors.”

373. This element is substantively the same as the matching step of element [15h], and I incorporate that analysis by reference. *Supra* § X.D.2.i. As I explained there, Datla queries its knowledge graph by “assigning, to a node of the knowledge graph comprising the extracted one or more patient symptoms, the assigned weight as an activation weight.” Ex. 1005 at ¶ [0055]. When “the TF-IDF weighted clinical concepts extracted from the clinical narrative are used to query the knowledge graph,” Datla “perform[s] all one-hop expansion of the symptom nodes.” *Id.* at ¶ [0057]. Identifying the node that comprises the patient’s extracted symptom is matching that diagnostic factor to the graph entry representing it.

374. Dang likewise detects nodes by “measuring, using semantic matching, a similarity between an input text [and] one or more nodes in the knowledge graph data structure.” Ex. 1008 at ¶ [0009]; *see also id.* at ¶ [0047].

375. Claim 21 recites “entries” rather than claim 15’s “nodes,” but, as I explained for claim 3, each entry in Datla’s graph is a node. *Supra* § X.A.4.

e. Element [21d]

376. Element [21d] recites: “each of said diagnostic factor entries being cross linked to at least one medical condition entry in said graph database representing a medical condition and not including information specific to said medical condition to which said each diagnostic factor entry is linked.”

377. This element combines the cross-linking requirement of elements [1f] and [15e] with the negative limitation of elements [1g] and [15e], and it applies that negative limitation to “each” diagnostic factor entry, as element [15e] does. I incorporate my analysis of elements [1f], [1g], and [15e] by reference. *Supra* §§ X.A.2.g, X.A.2.h, X.D.2.f.

378. Datla discloses that each diagnostic factor entry does not include information specific to the medical condition to which it is linked. A single symptom node in Datla “will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios.” Ex. 1005 at ¶ [0031]. Datla places the condition-specific relationship on the edge, not in the node: edges are “weighted based on a relationship between the nodes,” such as “the frequency with which the two nodes are associated within a source or a corpus.” *Id.* at ¶ [0032]. Storing information specific to one condition in the fever node would be inconsistent with

that node serving “many patient scenarios.” *Id.* at ¶ [0031]; *see also supra* § X.A.2.h (element [1g]); § X.D.2.f (element [15e]).

379. Adding Dang does not change this. Dang associates its conditional probability $P(f|d)$ with the connection between a disease and a feature, not with the feature itself. Ex. 1008 at ¶ [0031]. As I explain for element [21i], a POSITA implementing the combination would store Dang's probabilities on Datla's disease-symptom edges, leaving each symptom node condition-independent. *Infra* § X.F.2.j.

f. Element [21e]

380. Element [21e] recites: “wherein said diagnostic factor entries and said medical condition entries originate in at least one medical evidence journal,”

381. For the reasons given for element [1d], Datla discloses diagnostic factors discussed in its medical evidence sources as indicative of medical conditions. *Supra* § X.A.2.e. Datla also ties that relationship to journal text: each edge may be weighted by “the frequency with which the two nodes are associated within a source or a corpus, or that the two nodes appear within the same source such as an online database or within the same medical journal.” Ex. 1005 at ¶ [0032]. Given that the knowledge graph is built from the medical evidence sources, including medical evidence journals, the diagnostic factor and medical condition entries originate in at least one medical evidence journal. Ex. 1005 at ¶¶ [0030], [0032], [0047]-[0048].

382. MacRae (Ex. 1013), which is the journal article I identify below for element [21f] (*infra* § X.F.2.g), also discusses a diagnostic factor as indicative of a medical condition. MacRae’s Table 1 lists “Asthma” within the “Wheeze-related illness” classification category. Ex. 1013 at 4, Table 1. MacRae further explains that “the medications prescribed and symptoms described were highly predictive for Wheeze illness and for few other conditions.” *Id.* at 7–8.

g. Element [21f]

383. Element [21f] recites: “wherein said at least one medical evidence journal provides different taxonomies that arrange one or more of said diagnostic factors entries and/or one or more of said medical condition entries in different hierarchies.”

384. Element [21f] is similar to element [1d] in claim 1 and element [15c] in claim 15 but differs in that element [21f] requires that “at least one **medical evidence journal** provides different taxonomies that arrange one or more of said diagnostic factors entries and/or one or more of said medical condition entries in different hierarchies.” In other words, unlike claims 1 and 15 which recite that “medical evidence sources” that include a published journal provide different taxonomies, claim 21 specifically requires that a medical evidence **journal** provides the different taxonomies arranging diagnostic factor and/or medical condition entries in different hierarchies. As I discussed above, the specification of the ‘882 patent does not

disclose that a medical evidence journal provides different taxonomies that arrange one or more of said diagnostic factors entries and/or one or more of said medical condition entries in different hierarchies. Rather, the specification describes ontologies (which are part of the broader class of medical evidence sources) as providing taxonomies and hierarchies.

385. I understand that in the co-pending district court litigation, Patent Owner contends that this element is met by MacRae (Ex. 1013), an article published by BMJ Open. Ex. 1019. Specifically, Patent Owner contends that MacRae “puts ‘wheezing-ill’ in a first taxonomy that is based on the hierarchy of ‘classification of consultation’” (MacRae's Figure 1), and “also puts ‘wheeze’ and wheeze-related conditions into another taxonomy, which is a Table 1 that is based on the hierarchy of respiratory classification categories.” *Id.* Patent Owner additionally identifies SNOMED CT as placing “wheezing” in a further hierarchy “based on the hierarchy of respiratory sounds,” and contends that “[i]n the two medical evidence sources above, wheezing is arranged in different hierarchies based on different concepts.” *Id.* at 91–93. Regarding this second theory, I again note that the claim specifically says that a medical evidence journal provides different taxonomies and that SNOMED is not a journal.

386. I have been asked to apply Patent Owner's understanding of this element. Under that understanding, the combination satisfies element [21f] based on

MacRae alone. It would have been obvious to a POSITA that MacRae could have been part of the corpus of medical information described by Datla because MacRae is a medical journal article published in BMJ Open by The BMJ, a well-known and established publisher. According to Patent Owner, when combined with a system like Datla that uses MacRae to build a knowledge graph, the requirements of element [21f] are met.

h. Element [21g]

387. Element [21g] recites: “wherein a first diagnostic factor entry is discussed in part of said at least one medical evidence journal as being indicative of a first medical condition entry.”

388. For the reasons given for element [1d], Datla discloses diagnostic factors discussed in its medical evidence sources as indicative of medical conditions. *Supra* § X.A.2.e. Datla also ties that relationship to journal text: each edge may be weighted by “the frequency with which the two nodes are associated within a source or a corpus, or that the two nodes appear within the same source such as an online database or within the same medical journal.” Ex. 1005 at ¶ [0032].

389. MacRae also discusses a diagnostic factor as indicative of a medical condition. MacRae’s Table 1 lists “Asthma” within the “Wheeze-related illness” classification category. Ex. 1013 at 4, Table 1. MacRae further explains that “the

medications prescribed and symptoms described were highly predictive for Wheeze illness and for few other conditions.” *Id.* at 7–8.

i. Element [21h]

390. Element [21h] recites: “establish one or more application programming interfaces (APIs) for said graph database, said one or more APIs allowing an application to integrate with said graph database to access said diagnostic factor entries and said medical condition entries in said graph database in order to generate a healthcare determination.”

391. This element is substantively the same as elements [1h] and [15g], except that it recites access to “said diagnostic factor entries and said medical condition entries.” For the reasons given for those elements, Gnanasambandam’s APIs allow an application to integrate with the combined system’s knowledge graph to access the linked diagnostic factor and medical condition entries in order to generate a healthcare determination. *Supra* §§ X.A.2.i, X.D.2.h. I incorporate that analysis by reference.

j. Element [21i]

392. Element [21i] recites: “traverse a link from said each matched diagnostic factor entry to its linked medical condition and collecting a weight value

indicative that said each matched diagnostic factor entry is caused by its linked medical condition.”

393. Datla traverses links from matched symptom nodes to linked condition nodes. Datla assigns an activation weight “to a node of the knowledge graph comprising the extracted one or more patient symptoms” and expands “to one or more connected nodes.” Ex. 1005 at ¶ [0055]. Datla performs “all one-hop expansion of the symptom nodes.” *Id.* at ¶ [0057]. Datla’s edges link symptoms “to the diseases and medical conditions associated with those symptoms.” *Id.* at ¶ [0036]. Each edge is “weighted based on a relationship between the nodes,” with the weight “based on the strength of the relationship.” *Id.* at ¶ [0032].

394. Dang supplies a weight value expressly indicative of causation. In Dang, each disease–feature connection “may have a corresponding conditional probability P ,” where $P(f_x|d_n)$ “indicates . . . the likelihood of the feature f_x occurring when the disease d_n [is] observed.” Ex. 1008 at ¶ [0031]. A probability that a finding occurs given a disease measures how likely that finding is to be caused by the disease.

395. It would have been obvious to store, on each of Datla’s symptom–condition edges, the conditional probability of the symptom given the condition, as Dang teaches, and to collect that value when traversing from the matched symptom node to the linked condition node. A POSITA would have been motivated to make

this change for at least two reasons. First, Datla’s co-occurrence frequency does not distinguish the direction of the relationship. A conditional probability of the finding given the disease captures the disease-to-finding relationship on which diagnostic reasoning depends. Second, Dang teaches that combining its TF-IDF and Bayesian approaches on a symptom–disease knowledge graph “improve[s] diagnosis accuracy.” Ex. 1008 at ¶ [0040]. Storing a numerical value on an edge and reading it during traversal was routine, and the change leaves Datla’s traversal intact. A POSITA would therefore have had a reasonable expectation of success.

k. Element [21j]

396. Element [21j] recites: “sum all collected weight values for each linked medical condition to produce a total weight value for said each linked medical condition.”

397. Datla aggregates values at each condition node. For example, Datla’s control module “controls the accumulation of activations at one node.” Ex. 1005 at ¶ [0071].

398. Dang expressly discloses summation per disease: “Confidence of disease=Sum(Confidence of evidence*TF-IDF).” Ex. 1008 at ¶ [0050] (equation 4).

$\text{Confidence of disease} = \text{Sum}(\text{Confidence of evidence} * \text{TF-IDF})$	(equation 4)
--	--------------

Dang's disease confidence is computed "based on feature contributions." Ex. 1008 at ¶ [0050].

399. It would have been obvious to sum, for each linked condition, the weight values collected under element [21i], as Dang's equation 4 teaches. Datla's activation passes through a sigmoid function and is divided among sibling nodes. Ex. 1005 at ¶¶ [0057], [0069]. Dang's explicit sum gives each condition a total that can be broken down into the contribution of each matched factor. A POSITA would have recognized that this makes the ranking transparent and easier to explain to a clinician. Adding values was a routine operation, so a POSITA would have reasonably expected success.

I. Element [21k]

400. Element [21k] recites: "display on a computing device said linked medical conditions in order of total weight value as said healthcare determination."

401. Datla identifies "top ranked diseases and medical conditions" and provides "the ranked medical conditions and/or diagnosis . . . to the clinician" via a user interface "such as a speaker or screen." Ex. 1005 at ¶¶ [0062], [0072].

402. Dang determines a possible disease list with probabilities, "sorts (step 608) the PDL by probability rate," and outputs the top list "via a graphical user interface." Ex. 1008 at ¶ [0056].

403. In the combined system, the conditions would be ranked by the total weight value produced under element [21j], and displayed in that order. For the reasons given for claim 14, the ranked list is a healthcare determination. *Supra* § X.A.10.

3. Dependent Claim 22

404. Dependent claim 22 recites: “A computer system as recited in claim 21, wherein the instructions, when executed, further cause the one or more processors to: display in association with each linked medical condition its corresponding total weight value.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

405. Claim 22 adds one requirement to element [21k], which requires displaying the linked medical conditions in order of total weight value. Claim 22 requires displaying each condition’s total weight value alongside it. As I explained for element [21j], the combination computes a total weight value for each condition by summing the weights collected during traversal, as Dang’s equation 4 does: “Confidence of disease = Sum(Confidence of evidence * TF-IDF).” Ex. 1008 at ¶ [0050]; *supra* § X.F.2.k.

406. Dang shows displaying each disease with its computed value. Dang’s system “determines (step 602) a possible disease list (PDL) with probabilities (P_i , $i=1\sim n$),” “calculates (step 606) a probability rate (R_i) for each disease,” and “sorts

(step 608) the PDL by probability rate.” Ex. 1008 at ¶ [0056]. The resulting list may be output “to a user via a notification including via a graphical user interface.” *Id.* Dang also reports disease-specific numerical scores directly. For example, it describes a likelihood of “6.78 that Q is indicative of the flu and a likelihood of 2.01 that Q is indicative of allergic rhinitis.” *Id.* at ¶ [0038].

407. A POSITA would have found it obvious to display each condition’s total weight value next to that condition for at least two reasons. First, the ordering alone does not tell a clinician whether the top two conditions are nearly tied or far apart, whereas the actual values do convey this. Datla’s purpose is to reduce the clinician’s cognitive burden and to present “appropriate diagnoses and/or treatment for the patient in addition to the rationale for each option.” Ex. 1005 at ¶ [0003]. Second, the value is already calculated to produce the ranking. Datla already provides “the ranked medical conditions and/or diagnosis” to the clinician via “any user interface that allows information to be conveyed, such as a speaker or screen.” *Id.* at ¶ [0062]. Adding a number already in memory to a display is a routine user-interface choice with a predictable result.

4. Dependent Claim 26

408. Dependent claim 26 recites: “A computer system as recited in claim 21, wherein the instructions, when executed, further cause the one or more processors to: produce said plurality of diagnostic factors using natural language processing

(NLP); and initialize said natural language processing with said diagnostic entries of said graph database.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

409. Claim 26 requires the system to “produce said plurality of diagnostic factors using natural language processing (NLP).” Those are the factors produced in element [21b], i.e., the patient’s diagnostic factors known to be present or absent. Claim 26 then requires the system to “initialize said natural language processing with said diagnostic entries of said graph database.” Claim 21 does not recite “diagnostic entries,” but I understand the phrase to refer to the “diagnostic factor entries” of element [21c].

410. **Producing the diagnostic factors using NLP.** Datla produces the patient’s diagnostic factors with NLP. Datla’s system includes a “natural language processing engine configured to extract the at least one patient symptom and at least one demographic parameter from the received natural language input.” Ex. 1005, claim 1. Datla’s NLP engine “extracts at least one patient symptom from the received information.” Ex. 1005 at ¶ [0052]. As I explained for element [21b], these extracted symptoms are diagnostic factors known to be present or absent. *Id.* at ¶¶ [0052], [0054]; *supra* § X.F.2.c.

411. **Initializing the NLP with the graph database’s diagnostic factor entries.** Datla uses the extracted symptoms to query its knowledge graph by

“assigning, to a node of the knowledge graph comprising the extracted one or more patient symptoms, the assigned weight as an activation weight.” Ex. 1005 at ¶ [0055]. Datla does not state that its NLP engine consults the existing symptom nodes when it extracts the patient’s symptoms. Dang teaches that step. Dang detects nodes by “measuring, using semantic matching, a similarity between an input text [and] one or more nodes in the knowledge graph data structure.” Ex. 1008 at ¶ [0009]. Dang’s diagnostic process “detects factual nodes (for example, symptom nodes, test nodes, context nodes)” from “patient complaint records,” “using semantic analysis and matching,” with “Confidence of fact = similarity (input->node).” *Id.* at ¶ [0047]. Dang’s extraction of patient concepts is measured against the graph’s existing nodes, so its NLP is initialized with those nodes.

412. A POSITA would have been motivated to initialize Datla’s NLP engine with the graph’s existing diagnostic factor nodes, as Dang teaches, for at least three reasons. First, Datla’s query works only if an extracted symptom corresponds to a node, because activation is assigned to “a node of the knowledge graph comprising” the extracted symptom. Ex. 1005 at ¶ [0055]. Datla acknowledges that matching free text to graph concepts is imperfect: “certain diagnosis terms from MIMIC-II do not exactly match the Wikipedia page titles.” *Id.* at ¶ [0040]. Recognizing patient concepts against the graph’s own nodes avoids that mismatch. Second, the combined system already matches the patient’s factors to graph entries this way under element

[21c], using Dang’s semantic matching. *Supra* § X.F.2.d. Using the same node vocabulary when the factors are extracted is the natural way to implement that matching. Third, the modification changes only the vocabulary the NLP engine recognizes while leaving Datla’s activation, traversal, and ranking unchanged.

413. A POSITA would have had a reasonable expectation of success, because semantic matching of input text against knowledge-graph nodes was a known, working technique. Ex. 1008 at ¶¶ [0009], [0047].

5. Dependent Claim 27

414. Dependent claim 27 recites: “A computer system as recited in claim 21 wherein a demographics entry of said graph database located between one of said matched diagnostic factor entries and its corresponding linked medical condition includes demographics of people who exhibit said matched diagnostic factor in the context of said linked medical condition, and wherein the instructions, when executed, further cause the one or more processors to: only traverse said link from said one matched diagnostic factor entry to its linked medical condition when information of said patient from said EHR matches said demographics in said demographics entry.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

415. **The demographics entry.** The first requirement corresponds to element [15f]. Element [15f] requires demographics stored in a node “located

between” a condition node and a factor node. Claim 27 recites an “entry” rather than a node and describes the contents as demographics “of people who exhibit said matched diagnostic factor in the context of said linked medical condition.” Neither difference affects my analysis. For the reasons given for element [15f], it would have been obvious to store Datla’s demographic knowledge in a node like Dang’s Context node. Dang’s Context node is connected by one edge to the Disease node and by another to the Symptom node, so it lies between them. *Supra* § X.D.2.g; Ex. 1008 at ¶ [0027], FIG. 1A.

416. I note that Goldberg is not needed here. My analysis of element [15f] cited Goldberg as corroboration that demographic qualifications can be specific to a particular feature–disease relationship. *Supra* § X.D.2.g. Removing Goldberg does not change my opinion. Dang alone supplies the structure: a context node linked to both the disease and the symptom, which Dang’s diagnostic process detects as a factual node. Ex. 1008 at ¶¶ [0027], [0028]–[0029], [0047]. Datla supplies the reason to populate that node with demographics: it uses a patient’s demographic parameters to “query the knowledge-graph 310,” Ex. 1005 at ¶ [0076], and to lower the rank of a disease that “is not common for a demographic,” *id.* at ¶ [0061].

417. Claim 27 adds a requirement not in element [15h]. Element [15h] requires matching patient data to the demographics node. Claim 27 requires that the

system “only traverse” the link from the matched factor to its condition when the patient’s EHR information matches the stored demographics.

418. Datla discloses using patient demographics to adjust its diagnostic output. Datla applies the demographics after ranking: “if a disease is not common for a demographic, its rank is lowered.” Ex. 1005 at ¶ [0061]; *see also id.* at ¶ [0072]. Datla also teaches excluding candidates by demographics, not merely down-weighting them. Datla’s diagnostic-inferencing embodiment filters candidate articles using criteria such as “location, gender, match with topical keywords.” *Id.* at ¶ [0037]. Datla’s retrieval step likewise filters articles “by location (e.g. USA/Canada), demographic information and other contextual information.” *Id.* at ¶ [0047].

419. It would have been obvious to a POSITA to apply the demographic check as a gate on traversal, for at least three reasons. First, a POSITA would have recognized two known ways to handle a demographic mismatch: lower the condition’s rank afterward, as Datla ¶ [0061] does, or exclude it, as Datla ¶¶ [0037] and [0047] do with gender and demographic filters. Choosing exclusion is a choice between two known, predictable options. Second, where the demographics sit on a node between the symptom and the condition, the natural place to check them is when the traversal reaches that node. Third, Datla’s control module already limits how activation propagates, ensuring “there is no runaway activation among the

nodes.” Ex. 1005 at ¶ [0058]. Stopping propagation across a link whose demographic criteria the patient does not meet is the same kind of control and keeps demographically implausible conditions out of the ranked output, consistent with Datla’s goal “to improve the accuracy of the recommendations relative to the patient scenario of interest.” *Id.* at ¶ [0061].

420. **The “EHR” requirement.** The patient information comes from the EHR, as element [21b] requires. *Supra* § X.F.2.c. Dang identifies the EMR as the source of the patient’s symptoms and age, Ex. 1008 at ¶ [0036]. Datla extracts demographic parameters such as “age, height, weight, medical background, sex” from the patient information. Ex. 1005 at ¶ [0051].

421. A POSITA would have had a reasonable expectation of success because comparing a patient’s age or sex with a stored range or value before following an edge is a simple conditional check during graph traversal.

6. Dependent Claim 28

422. Dependent claim 28 recites: “A computer system as recited in claim 21 wherein said display is for the purpose of patient diagnosis, for an administrative medical purpose, or for a medical preauthorization.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

423. Claim 28 refers to “said display,” and the only display recited in claim 21 is element [21k]’s step to “display on a computing device said linked medical

conditions in order of total weight value as said healthcare determination.” Claim 28 asks what that ranked display of medical conditions is for. The claim is disjunctive: the display must be “for the purpose of patient diagnosis, for an administrative medical purpose, or for a medical preauthorization.” The combination meets the first alternative.

424. As I explained for element [21k], Datla identifies “top ranked diseases and medical conditions” and provides “the ranked medical conditions and/or diagnosis . . . to the clinician” through a user interface “such as a speaker or screen.” Ex. 1005 at ¶¶ [0062], [0072]; *supra* § X.F.2.1. Dang likewise sorts its possible disease list “by probability rate” and outputs it “via a graphical user interface.” Ex. 1008 at ¶ [0056]; *supra* § X.F.2.1. In the combined system, the display shows the patient’s linked medical conditions ranked by total weight value.

425. Datla’s system is for the purpose of patient diagnosis. Datla is directed to “automated methods and systems to provide a clinical diagnosis of a patient’s symptoms based on a corpus of medical knowledge.” Ex. 1005 at ¶ [0001]. Datla’s system “accepts natural language input from a medical professional about a patient’s scenario, processes the input, and provides one or more possible diagnoses, tests, and/or treatments.” Ex. 1005 at ¶ [0028]. Datla identifies “the most probable diagnosis given a clinical narrative” and provides “the highest ranked one or more medical conditions, diagnoses, treatments, and/or tests for the patient” to the user.

Ex. 1005 at ¶¶ [0011], [0057]. Datla presents those results so the clinician can act on them, and it seeks to present “appropriate diagnoses and/or treatment for the patient in addition to the rationale for each option.” Ex. 1005 at ¶ [0003]. Dang serves the same end. Its suspected disease list supports “a more targeted medical diagnosis.” Ex. 1008 at ¶ [0025].

426. For claim 14, I explained that Datla’s ranked list of medical conditions is a “result,” an “evaluation,” and a “conclusion” about the patient’s likely conditions, presented to the clinician for decision-making. *Supra* § X.A.10. I incorporate that analysis here. A ranked differential of a specific patient’s likely conditions, displayed to that patient’s clinician, is a display for the purpose of patient diagnosis. That is its only purpose in Datla and Dang. No further modification is required.

7. Dependent Claim 29

427. Dependent claim 29 recites: “A computer system as recited in claim 21 wherein said graph database includes more than twenty thousand entries.” I

incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

428. Claim 29 is the same as claim 12; the only difference is the parent claim. I incorporate by reference my discussion of claim 12. While Goldberg is not part of the ground for claim 29, that does not change my opinion.

429. As I discussed above, Datla builds its graph from sources including “medical journals,” and when using a source with a master category such as “clinical medicine,” “all pages including and under the master top category are analyzed.” Ex. 1005 at ¶ [0030]. Datla’s nodes include “conditions, diagnoses, tests, procedures, medications, or other clinical concepts,” and a single symptom such as fever “will be connected by edges to hundreds of other nodes.” *Id.* at ¶ [0031]. The graph is updated “with new journal articles, new online sources, or other possible sources of new information.” *Id.* at ¶ [0032].

430. My routine-scaling and duplication-of-parts analysis for claim 12 applies equally here. *Supra* § X.A.8. Additional entries are more instances of the node types Datla already discloses, connected by the same weighted edges. The system operates the same way on a larger graph. The ‘882 patent attaches no significance to the claimed threshold. As I explained in the Technology Background, systems with hundreds of thousands of concepts were in routine use before the priority date. *Supra* § VI.

8. Dependent Claim 30

431. Dependent claim 30 recites: “A computer system as recited in claim 21 wherein said graph database includes more than one hundred thousand entries.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

432. Claim 30 uses the same words as claim 13; the only difference is the parent claim. For the reasons given for claim 29, I rely here on the scope of Datla’s graph and on routine scaling. I do not rely on Goldberg’s SNOMED vocabulary, which my claim 13 analysis also cited. Removing Goldberg does not change my opinion.

433. Moving from twenty thousand to one hundred thousand entries adds more instances of the same node types, connected by the same kind of weighted edges, and the system applies the same weighting, traversal, and ranking to the larger graph. *Supra* § X.F.7. Datla teaches a graph covering all of clinical medicine that grows without a stated upper limit as journal literature is added. Ex. 1005 at ¶¶ [0030]–[0032]. The ‘882 patent describes no benefit or change in operation tied to one hundred thousand entries.

9. Dependent Claim 31

434. Dependent claim 31 recites: “A computer system as recited in claim 21 wherein said healthcare determination is a recommendation, an evaluation, a

result, or a conclusion.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

435. Claim 31 recites the same limitation as dependent claim 14, and my analysis of claim 14 applies equally here. *Supra* § X.A.10.

436. In claim 21, the “healthcare determination” is the display of the “linked medical conditions in order of total weight value.” As I explained for claim 21, the combination displays a ranked list of medical conditions. *Supra* § X.F.2. That is the same kind of output I addressed for claim 14: Datla’s “ranked list of one or more medical conditions, diagnoses, treatments, and/or tests for the patient,” which is “provided to the user via the user interface.” Ex. 1005 at ¶ [0007]. For the reasons given for claim 14, a ranked list of likely conditions is a “result” of the ranking process, an “evaluation” of the patient’s likely conditions, and a “conclusion” as to which conditions are most probable. *See id.* at ¶¶ [0011], [0057]. It is also a “recommendation.” *Id.* at ¶ [0061].

G. Ground 7: Claims 23 and 24 are Obvious Over Datla, Dang, Gnanasambandam, MacRae, and Kim

437. In my opinion, claims 23–24 of the ‘882 patent are obvious over Datla, Dang, Gnanasambandam, MacRae, and Kim.

438. My motivation-to-combine analysis for Ground 2 applies equally here. *Supra* § X.B.1. My first and third reasons for adding Kim to the Datla–Goldberg–

Gnanasambandam system in Ground 2 do not depend on Goldberg. Removing Goldberg does not change my opinion.

1. Dependent Claim 23

439. Dependent claim 23 recites: “A computer system as recited in claim 21, wherein the instructions, when executed, further cause the one or more processors to: for each present diagnostic factor, traverse upward in a medical ontology link found in said matched diagnostic factor entry in order to infer other diagnostic factors present in said patient and match these other diagnostic factors to other diagnostic factor entries in said graph database.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

440. For the reasons given for claim 4, it would have been obvious to include in each of Datla’s diagnostic factor entries a unique ontology identifier, as Kim teaches. *Supra* § X.B.2. Kim explains that entries “may be identified by a unique ontology code” and “may themselves be annotated with identifiers from several different ontologies.” Ex. 1012 at ¶¶ [0041], [0051]. That identifier is the “medical ontology link found in said matched diagnostic factor entry.” Kim also teaches using the link to reach related concepts: “[o]nce an ontology code (unique identifier) has

been assigned . . . related concepts as represented in the graph database can be suggested.” *Id.* at ¶ [0052].

441. Dang infers additional findings from detected ones by moving from specific terms to general ones. Dang’s process “may infer (step 504) evidential nodes from the nodes detected (step 502) . . . using reasoning through hyponymy, where a relationship is detected between a generic term (a hypernym) and an instance (hyponym) of the generic term.” Ex. 1008 at ¶ [0048]. A hyponym “is a word or phrase whose semantic field is more specific tha[n] its hypernym.” *Id.*; *see id.* at ¶ [0049]. Reasoning from a present, specific finding to its more general parent is traversing upward.

442. Matching the inferred factors to other entries. Dang adds the inferred nodes to the set of nodes used for diagnosis. It identifies “candidate evidential nodes among the set of factual nodes . . . based on hyponymy relationships between at least two factual nodes,” and adds them “as an inferred evidential node.” Ex. 1008 at ¶ [0010]; *see id.*, claim 9. The inferred factor is therefore matched to another diagnostic factor entry in the graph.

443. Once each entry carries a standard ontology identifier (Kim), the natural way to perform Dang’s hyponymy reasoning is to follow that identifier to the parent concept in the ontology. A POSITA would have expected this to work, because standard ontologies organize concepts in parent–child hierarchies, as I explained in

the Technology Background. *Supra* § VI.A. Datla already preserves source hierarchy in its graph. Ex. 1005 at ¶ [0035], claim 16. Inferring a present parent finding captures matches that exact-term matching would miss, which improves the diagnostic accuracy Datla seeks. Ex. 1005 at ¶ [0061].

2. Dependent Claim 24

444. Dependent claim 24 recites: “A computer system as recited in claim 21 wherein each said diagnostic factor entry includes at least one synonym of said diagnostic factor.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

445. Claim 24 is substantively the same as claim 5. The only differences are that claim 24 depends from claim 21, and that it omits claim 5’s phrase “represented by said each diagnostic factor entry,” which makes it broader. My claim 5 analysis relies on Kim for the synonyms and on Datla for the entries. Dang’s role in the claim 21 combination does not affect it, because claim 21 recites the same diagnostic factor entries. *Supra* § X.F.2. I summarize that analysis below and incorporate it by reference.

446. Kim states that “the graph database includes synonyms, translations, concepts and relationships, and the proposed methods and systems allow to add new synonyms, translations, concepts and relationships.” Ex. 1012 at ¶ [0054]. Kim ties those synonyms to individual concept entries and explains that, “in order to add

translations or synonyms for terms of existing ontology concepts to one or more entries of the graph database, the respective concept might have to be identified using existing terms describing it.” *Id.* at ¶ [0055]. Kim also teaches searching “all preexisting concept synonyms within the graph database.” *Id.* at ¶ [0054]. And it teaches that “synonyms and translations are also covered by the terminology system of the graph database and can be matched to the entered node labels.” *Id.* at ¶ [0060].

447. It would have been obvious to include synonyms in each of Datla’s diagnostic factor entries. In the combined system, for example, the fever entry would include the synonyms used for that finding in the medical literature. A POSITA would have been motivated to make this change for at least two reasons. First, Datla builds its graph from multiple sources that describe the same concept in different words. Datla acknowledges that “certain diagnosis terms from MIMIC-II do not exactly match the Wikipedia page titles.” Ex. 1005 at ¶ [0040]. Storing synonyms on each entry lets the system recognize a factor however a source or a patient record describes it. The same synonyms also help the system avoid redundant nodes, a benefit Kim describes. Ex. 1012 at ¶ [0054]. Second, as I explained for claim 23, it would have been obvious to give each entry a Kim ontology identifier. *Supra* § X.G.1. Standard ontologies provide multiple terms for each concept, so a POSITA who linked each entry to an ontology entry would have found it natural to store that entry’s terms on the node as well.

448. Storing one or more text strings as a node property is routine. Kim confirms that its graph database already “includes synonyms” and allows new synonyms to be added. Ex. 1012 at ¶ [0054]. The change leaves Datla’s matching, traversal, and ranking unchanged.

H. Ground 8: Claim 25 is Obvious Over Datla, Dang, Gnanasambandam, MacRae, and Van Assel

449. In my opinion, claim 25 of the ‘882 patent is obvious over Datla, Dang, Gnanasambandam, MacRae, and Van Assel.

1. Dependent Claim 25

450. Dependent claim 25 recites: “A computer system as recited in claim 21 wherein each said diagnostic factor entry includes a temporality attribute, a negation attribute or an experiencer attribute.” I incorporate by reference my discussion of independent claim 21 above. *Supra* § X.F.2.

451. Claim 25 is substantively the same as claim 6; the only difference is that it depends from claim 21 instead of claim 1. The claim is disjunctive, and I address the negation attribute. For the reasons given for claim 6, *supra* § X.C.1, the combination renders obvious diagnostic factor entries that each include a negation attribute. I incorporate that analysis by reference and summarize it below.

452. Datla’s system may “filter out negated clinical concepts in order to give more weight to positive clinical manifestations.” Ex. 1005 at ¶ [0054]. To filter a concept as negated, the system must first classify it as present or absent. Element

[21b] of claim 21 already requires diagnostic factors “known to be present or absent,” and the combination produces them through this same negation filtering. *Supra* § X.F.2.c.

453. Van Assel “assigns 1, -1, or 0 to the extracted concept to indicate whether the concept is validly present (1), validly absent (-1) or this is unknown (0).” Ex. 1011 at ¶ [0245]. Van Assel treats that value as information about the concept itself: “Asthma is an item of medical data, while the information about the medical data is the confidence, i.e. whether the concept is validly present, validly absent, or this is unknown.” *Id.* at ¶ [0253]. Van Assel applies the same state to nodes: “A node corresponding to an item of medical data can be present, ‘Evidence(node=node, state=PRESENT)’, or absent, ‘Evidence(node=node, state=ABSENT).’” *Id.* at ¶ [0248].

454. A POSITA would have been motivated to store the present/absent status that Datla already determines as an attribute of each matched diagnostic factor entry, as Van Assel teaches. First, Van Assel is in the same field. Like Datla, it performs automated clinical diagnosis using a medical knowledge graph and extracts clinical concepts from text with NLP. Ex. 1011 at ¶¶ [0110], [0113], [0115]; Ex. 1005 at ¶ [0052]. Second, claim 21 already requires the system to know whether each matched factor is present or absent. *Supra* § X.F.2.c. Recording that status on

the entry makes it available when the system traverses the graph and ranks conditions.

455. Storing a binary or ternary value as a node property is a routine graph-database operation. It leaves Datla's matching, traversal, and ranking unchanged, so a POSITA would have reasonably expected success.

456. My claim 6 analysis does not rely on Goldberg, and Dang's role in the claim 21 combination does not affect it. Removing Goldberg does not change my opinion.

I. Ground 9: Claims 32-33 and 38-41 are Obvious Over Zhong, Datla, Gnanasambandam, and MacRae

457. In my opinion, claims 32–33 and 38–41 of the '882 patent are obvious over Zhong, Datla, Gnanasambandam, and MacRae.

1. Motivation to Combine

458. Zhong is the primary reference for this ground. As set forth below, a POSITA would have been motivated to combine Zhong with Datla, Gnanasambandam, and MacRae. My reasons for combining Datla and Gnanasambandam (Ground 1) and for adding MacRae to Datla's corpus (Ground 6) apply here, and I do not repeat them. *Supra* §§ X.A.1, X.F.1.

459. A POSITA would have been motivated to incorporate Datla's condition-independent node architecture into Zhong's knowledge graph for at least the following reasons.

460. Zhong and Datla both operate in the field of knowledge graph-based clinical diagnosis systems. Ex. 1009 at Abstract; Ex. 1005 at ¶¶ [0001], [0028]. Both build knowledge graphs linking symptoms to diseases and use those graphs to support clinical decision-making. A POSITA working on Zhong's question-generation system would naturally look to Datla for techniques to improve knowledge graph architecture.

461. Zhong's knowledge graph links symptom nodes to disease nodes through edges carrying symptom–disease pair values. Ex. 1009 at Claim 1; 3:20–25. Datla expressly discloses condition-independent diagnostic factor nodes: the same symptom node links to hundreds of conditions through separate weighted edges, with condition-specific information stored on the edges rather than in the node itself. Ex. 1005 at ¶¶ [0031]–[0032]. Incorporating Datla's condition-independent architecture into Zhong's knowledge graph would improve scalability (e.g., with each diagnostic factor stored once and reused across all conditions) and would enable Zhong's system to identify candidate symptoms that are shared across multiple diseases without duplicating entries.

462. The combination uses each element for its known purpose: Zhong's iterative question-generation method operating on Datla's condition-independent knowledge graph architecture. The resulting automated diagnosis system that identifies candidate symptoms to ask about, using a knowledge graph in which each

symptom node is condition-independent and linked to multiple diseases through weighted edges is entirely predictable.

463. Zhong does not describe how its knowledge graph is built or where its symptom–disease correlations come from. It teaches only that the graph “may couple to external resources, such as various medical database or websites, for knowledge update such that the knowledge graph may be in line with current medical findings.” Ex. 1009 at 3:25–29. Datla fills that gap. It builds its graph from a corpus that includes “medical journals.” Ex. 1005 at ¶ [0030]. Zhong’s stated goal of keeping its graph “in line with current medical findings” would have led a POSITA to Datla’s journal-based construction. The combination uses each reference for its known purpose.

464. A POSITA would have been motivated to incorporate Gnanasambandam’s API architecture into the Zhong system (including as modified by Datla) for at least the following reasons.

465. Gnanasambandam, like Zhong, operates in the field of medical knowledge graph-based decision support. Gnanasambandam builds knowledge graphs representing knowledge of diseases and uses them for analytics, diagnoses, and healthcare determinations. Ex. 1007 at 51:27–34, 55:50–58, 100:24–28. A POSITA building a medical knowledge graph system would naturally look to Gnanasambandam for techniques to deploy that system as a service.

466. APIs are a standard deployment mechanism. A POSITA would have recognized that publishing APIs for a knowledge graph-based system is the standard way to make such a system accessible to external applications, including clinical decision support tools, electronic health record systems, and other healthcare applications. Gnanasambandam teaches this deployment pattern: the back-end component publishes APIs that perform analytics based on a knowledge graph, and these APIs are provided in a user interface to a client device. Ex. 1007 at 100:24–34; 102:22–24.

467. Zhong’s system is designed for external integration. Its question generation system “may be an on-site system or a remote system communicating to the patient through various communication means, such as telephone, Internet.” Ex. 1009 at 3:10–18. Zhong’s “input data and/or output data may be remotely transmitted from one physical location to another.” *Id.* at 7:35–37. Zhong’s knowledge graph “may be a standalone component coupling to the question generation system.” *Id.* at 2:61–63. This input/output architecture is naturally suited to API-based access.

468. Adding an API layer to a knowledge graph-based diagnostic system involves applying a known technique (API publication, as taught by Gnanasambandam) to a known system to yield the predictable result of an API-

accessible medical knowledge graph system that determines next-best actions for a medical condition.

469. For the reasons given for Ground 6, it would have been obvious to include MacRae, a BMJ journal article, in the journal corpus from which the combined system’s graph is built. *Supra* § X.F.1. Adding Zhong does not affect those reasons, because Zhong expressly contemplates updating its graph from external medical sources. Ex. 1009 at 3:25–29.

2. Independent Claim 32

470. Each element of independent claim 32 of the ‘882 patent is disclosed or rendered obvious by the combination of Zhong, Datla, Gnanasambandam, and MacRae. I address each element below.

a. Element [32pre]

471. Element [32pre] recites: “A computer system for determining next-best actions for a medical condition, said computer system comprising:”

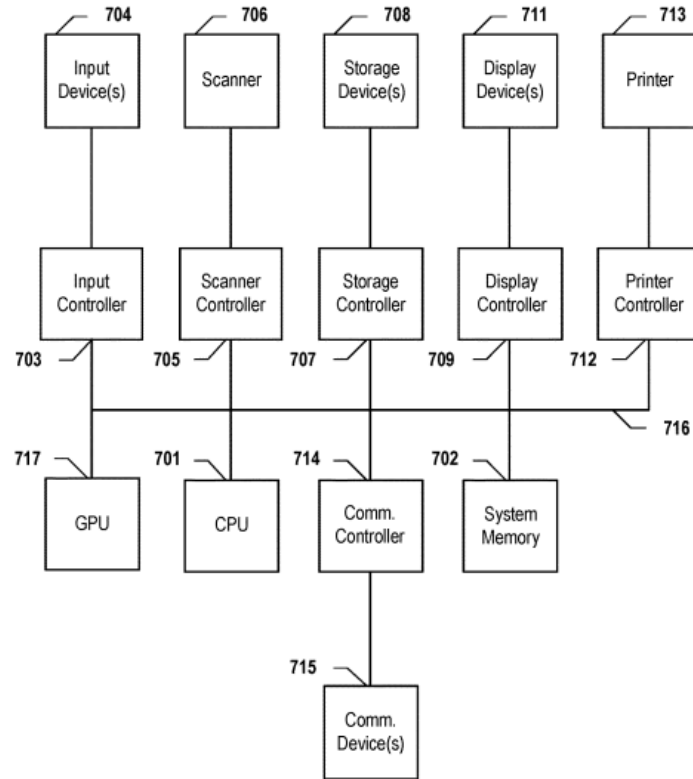
472. To the extent the preamble is limiting, Zhong discloses a computer system for determining next-best actions for a medical condition. Zhong is directed to “[s]ystems and methods . . . for question generation to obtain more related medical information based on observed symptoms from a patient.” Ex. 1009, Abstract. Zhong’s system implements an iterative process in which observed symptoms are collected, possible diseases are identified from a knowledge graph, candidate

symptoms (those not yet observed) are identified, scored, and presented to the clinician as the next actions to take. *Id.* at 3:39–4:31; FIG. 3. Asking the patient about the presence or absence of previously unknown diagnostic factors constitutes “next-best actions” for advancing the diagnosis. The process repeats iteratively until “a disease is diagnosed or the patient reports no more observed symptoms.” Ex. 1009 at 4:27–31.

b. Element [32a]

473. Element [32a] recites: “one or more processors; and memory that stores instructions, wherein the instructions, when executed by the one or more processors, cause the one or more processors to.”

474. Zhong discloses a computer system comprising one or more processors and memory that stores instructions. Zhong’s FIG. 7 depicts computing system 700 with explicitly labeled CPU 701 and system memory 702.



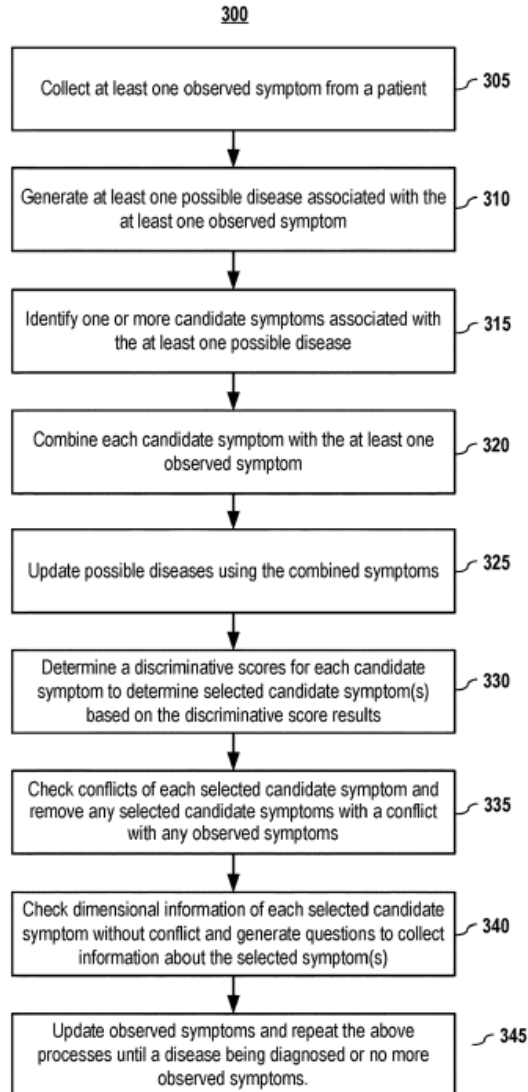
Ex. 1009 at FIG. 7

475. “[S]ystem 700 includes one or more central processing units (CPU) 701 that provides computing resources and controls the computer. CPU 701 may be implemented with a microprocessor, microcontroller, or the like, and may also include one or more graphics processing units (GPU) 717 and/or a floating point coprocessor for mathematical computations.” Ex. 1009 at 6:61–67. “[S]ystem 700 may also include a system memory 702, which may be in the form of random-access memory (RAM), read-only memory (ROM), or both.” *Id.* at 6:67–7:3.

476. Zhong further discloses that memory stores instructions executed by the processor. Zhong’s claim 7 recites: “one or more processors” and “a non-transitory

computer-readable medium or media comprising one or more sequences of instructions which, when executed by at least one processor of the one or more processors, causes the steps to be performed.” Ex. 1009 at Claim 7. Zhong’s claim 12 recites: “one or more processors coupled to the input interface and the output interface” and “a non-transitory computer-readable medium or media comprising one or more sequences of instructions which, when executed by at least one of the one or more processors, causes the steps to be performed.” *Id.* at Claim 12.

477. As set forth below, Zhong’s system performs the functionalities recited in claim 32. Zhong’s Figures 3 through 6 describe a single, nested, iterative pipeline. Figure 3 is the master-level flow defining the complete diagnostic loop.

**FIG. 3**

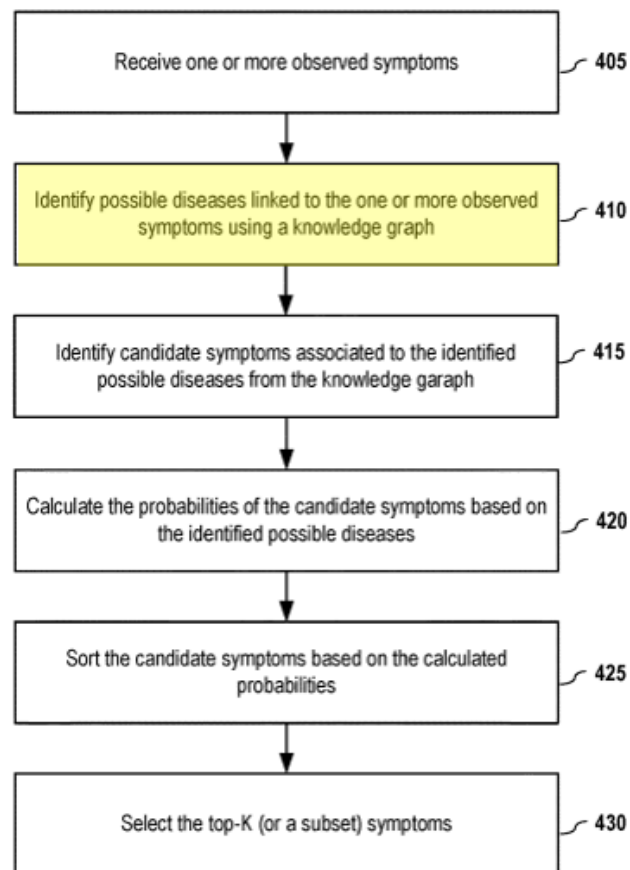
Ex. 1009 at FIG. 3

478. Figures 4, 5, and 6 each elaborate on specific steps within Figure 3: Figure 4 expands on candidate-symptom generation and initial probability-based selection (Steps 310–315 of Figure 3); Figure 5 expands on discriminative scoring and refined selection (Step 330 of Figure 3); and Figure 6 expands on conflict filtering, question generation, and iterative updating (Steps 335–345 of Figure 3).

c. Element [32b]

479. Element [32b] recites: “for each medical condition of a patient having a probability greater than a threshold of being present in said patient, perform the following.”

480. Zhong discloses this threshold-scoping element expressly. Before identifying candidate symptoms, Zhong’s system first narrows the disease set to those above a probability threshold. This occurs at FIG. 4, Step 410.



Ex. 1009 at FIG. 4 (annotated)

481. Zhong discloses that “the most likely diseases may be identified using a graph inference algorithm, such as Markov chain Monte Carlo (MCMC) or Gibbs sampling, etc. In embodiments, the most likely diseases are those diseases that have a probability above a threshold value or it may be the top X number of diseases.” Ex. 1009 at 4:40–46; FIG. 4.

482. Zhong’s claim 1 further confirms this threshold architecture: “forming a set of likely diseases comprising diseases from the plurality of diseases that have a probability above a threshold and a set of unlikely diseases that do not have a probability above the threshold.” Ex. 1009 at Claim 1.

483. Only the above-threshold diseases are carried forward into the candidate-symptom and scoring pipeline. All subsequent steps operate exclusively on these above-threshold diseases.

d. Element [32c]

484. Element [32c] recites: “identify all known diagnostic factor entries in a graph database that each represent a diagnostic factor known to be present or absent in said patient.”

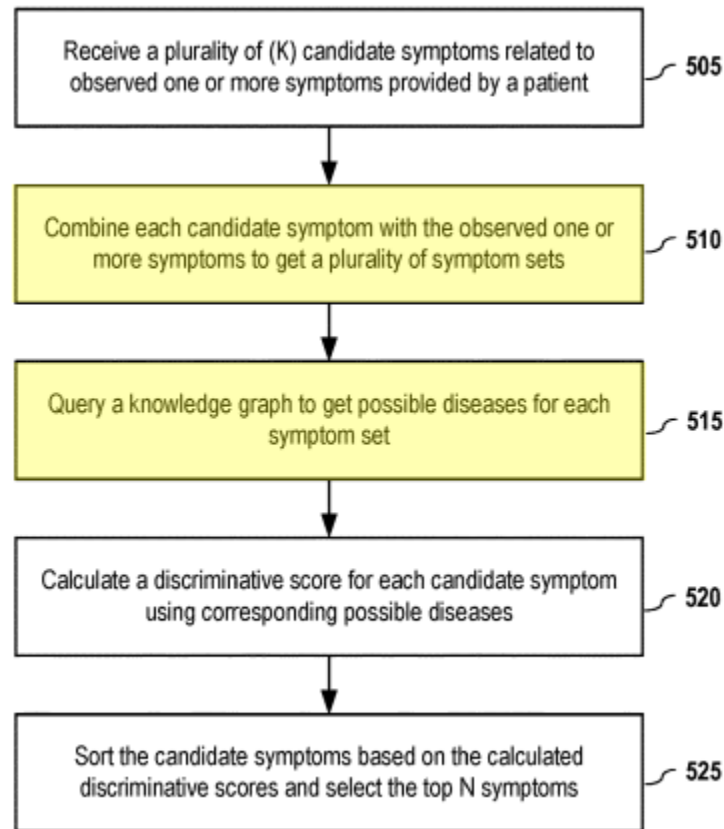
485. Zhong’s system collects observed symptoms from the patient. At FIG. 3, Step 305, “a set of one or more observed symptoms (S_o) is collected from the patient and input via the input interface.” Ex. 1009 at 3:41–44. In subsequent iterations of Zhong’s loop, the observed-symptom set expands to include the

patient's responses to generated questions: “the observed symptoms are updated after the information is collected from the generated questions.” *Id.* at 4:27–31. When a patient answers a generated question by denying the presence of a candidate symptom, that symptom becomes a diagnostic factor known to be absent. This initial symptom collection at Step 305 is a prerequisite data-gathering step — it feeds into the threshold determination at Step 410.

486. The identification of which known diagnostic factors are linked to each above-threshold medical condition occurs at and after Step 410. At FIG. 4, Step 410, the system queries the knowledge graph using the observed symptoms and applies the threshold to identify likely diseases. Ex. 1009 at 4:38–46. To compute disease probabilities, the graph inference algorithm (MCMC or Gibbs sampling) traverses the edges linking symptom nodes to disease nodes — meaning the system has resolved which observed symptom nodes (known diagnostic factor entries) are linked to which above-threshold disease nodes (medical condition entries) as part of the probability computation.

487. In FIG. 4, at step 415, when the system identifies candidate symptoms as the complement of the observed set, it is working with knowledge of which observed symptoms link to each above-threshold disease — otherwise it could not determine the complement. Ex. 1009 at 4:45–52.

488. This post-threshold identification of known diagnostic factors becomes even more explicit in FIG. 5 at steps 510–515. Ex. 1009 at 5:10–16.



Ex. 1009 at FIG. 5 (annotated)

489. At step 510, “each candidate symptom is combined with observed symptoms to get a plurality of symptom sets.” Ex. 1009 at 5:10–12. At step 515, “the symptom sets are used to get likely diseases and unlikely diseases by querying the knowledge graph.” *Id.* at 5:13–16. At these steps, the system is expressly operating on the observed (known) symptoms in the context of the above-threshold diseases, querying the graph to determine how each known symptom, combined with each candidate, affects disease probabilities. This demonstrates that the system has

identified which known diagnostic factors are linked to which medical condition entries, all after the threshold filtering step.

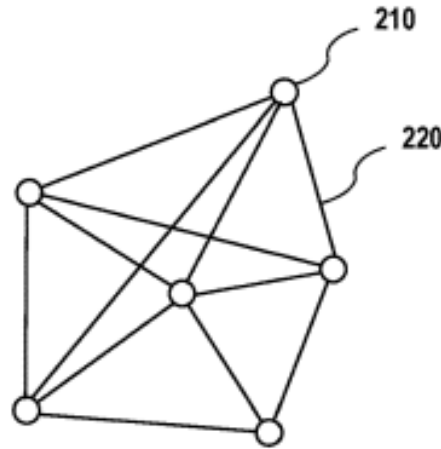
e. Element [32d]

490. Element [32d] recites: “each known diagnostic factor entry being linked to a medical condition entry in said graph database representing said each medical condition but not including information specific to said each medical condition.”

491. Zhong’s knowledge graph links symptom nodes to disease nodes through edges. Zhong’s claim 1 recites: “an edge that links a symptom node to a disease node has an associated symptom–disease pair value that represents a probability between the disease represented by the disease node given an observation of the symptom associated with the symptom node.” Ex. 1009 at Claim 1. The fact that querying the graph with observed symptoms returns possible diseases at step 410 confirms that each known diagnostic factor entry (symptom node) is linked to medical condition entries (disease nodes) in the graph database.

492. The claim requires that each known diagnostic factor entry does “not includ[e] information specific to said each medical condition.” Zhong’s knowledge graph architecture expressly stores condition-specific information on the edges between nodes rather than within the symptom nodes themselves. Zhong’s specification describes the knowledge graph as follows: “a node (e.g., node 210) in

the graph is an entity, such as a symptom, a dimension, a demographic, and a disease, and an edge (e.g., edge 220) in the graph represents a relationship between two entities.” Ex. 1009 at 3:20–25.



Ex. 1009 at FIG. 2

493. Zhong’s claim 1 makes clear that the condition-specific information — the symptom–disease pair value representing the probability of a disease given a symptom — is stored on the edge, not in the symptom node: “an edge that links a symptom node to a disease node has an associated symptom–disease pair value.” Ex. 1009 at Claim 1.

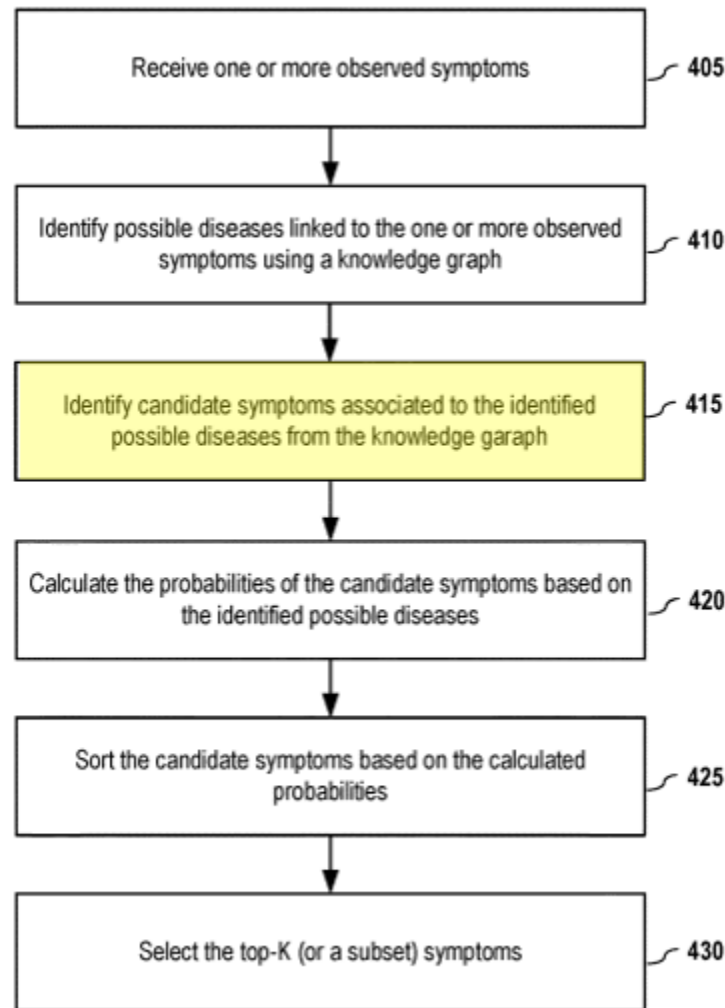
494. A POSITA would understand that Zhong’s placement of condition-specific probability values on the edges rather than in the nodes, combined with Zhong’s express description of a symptom node as an “entity” representing “a symptom” without any suggestion that it carries disease-specific data, as disclosing that Zhong’s symptom nodes are condition-independent. That is, the nodes do not

include information specific to a particular medical condition. This understanding is reinforced by Zhong's architecture: because each symptom node can link to multiple disease nodes through separate edges (each carrying its own symptom–disease pair value), the logical place for condition-specific information is on those edges, not within the symptom node that is shared across diseases.

f. Element [32e]

495. Element [32e] recites: “identify all unknown diagnostic factor entries in said graph database that each represent an unknown diagnostic factor not known to be present or absent in said patient, each unknown diagnostic factor entry being linked to said medical condition entry.”

496. The step that satisfies element [32e] is FIG. 4, Step 415, which occurs immediately after the threshold filtering step at Step 410. At Step 415, the system traverses the knowledge graph from the above-threshold disease nodes outward to connected symptom nodes: “M (an integer number larger than 1) candidate symptoms (S_c) associated with the likely diseases are identified using the knowledge graph.” Ex. 1009 at 4:46–49; FIG. 4, Step 415.



Ex. 1009 at FIG. 4 (annotated)

497. Zhong defines candidate symptoms as the complement of the known set: “a candidate symptom may be an unobserved symptom or a symptom not reported to the automatic diagnosis system.” Ex. 1009 at 3:52–54. This definition maps directly to the claim’s “unknown diagnostic factor not known to be present or absent in said patient.” An unobserved, unreported symptom is one whose diagnostic status has not been established.

498. Zhong further confirms that the candidate symptoms are limited to genuinely unknown factors through its conflict-checking mechanism. At FIG. 3, Step 335 (expanded in FIG. 6, Step 610): “the selected candidate symptoms are checked for conflicts with observed symptoms and any selected candidate symptom with a conflict is removed from further consideration.” Ex. 1009 at 4:14–17; FIG. 3, Step 335. For example, “if a symptom relates to males only but the patient is female, that selected candidate system [sic] has a conflict and is disregarded.” *Id.* at 4:17–19. This conflict-removal step ensures that only genuinely unknown diagnostic factors (i.e., those the patient has not yet been asked about and that do not conflict with observed data) survive for ranking and display.

499. The linkage requirement is satisfied by the identification method. Because candidate symptoms are identified at step 415 by traversing the knowledge graph from the above-threshold disease nodes to connected symptom nodes, each identified candidate symptom is, by the method of its identification, linked to at least one above-threshold disease node in the graph. Zhong confirms the bidirectional nature of these links: “Each candidate symptom may be associate to one or more possible diseases. Similarly, each possible disease may associate to one or more candidate symptoms.” Ex. 1009 at 4:48–52. The knowledge graph structure confirms

that these links are stored as edges carrying symptom–disease pair values. *Id.* at Claim 1.

g. Element [32f]

500. Element [32f] recites: “wherein said known and unknown diagnostic factor entries and said medical condition entry originate in at least one medical evidence journal, wherein said at least one medical evidence journal provides different taxonomies that arrange one or more of said known and unknown diagnostic factors and/or one or more of said medical condition entries in different hierarchies, and wherein a first diagnostic factor entry is discussed in part of said at least one medical evidence journal as being indicative of a first medical condition entry.”

501. Zhong discloses that “the knowledge graph 200 is a graph encoding domain knowledge, where a node (e.g., node 210) in the graph is an entity, such as a symptom, a dimension, a demographic, and a disease, and an edge (e.g., edge 220) in the graph represents a relationship between two entities.” Ex. 1009 at 3:20-25. Zhong further discloses that “the knowledge graph may couple to external resources, such as various medical database or websites, for knowledge update such that the knowledge graph may be in line with current medical findings.” *Id.* at 3:25-29. Thus, Zhong discloses that the knowledge graph can be based on information in medical databases or websites so that it is in line with current medical findings. However,

Zhong does not explicitly state that the entries in the knowledge graph originate in at least one medical evidence journal.

502. Datla expressly discloses constructing a knowledge graph containing entries that originate in at least one medical evidence journal, wherein a first diagnostic factor entry is discussed in part of said at least one medical evidence journal as being indicative of a first medical condition entry.

503. Datla identifies “medical journals” as a source category for its corpus of medical information: “The corpus of medical information can be any source of information, including but not limited to *medical journals*, newspapers, online sources such as Wikipedia, and other sources.” Ex. 1005 at ¶ [0030] (emphasis added). Datla also discloses processing abstracts of medical journal articles in PubMed. *Id.* at ¶¶ [0033], [0047]-[0048]. PubMed is a service of the U.S. National Library of Medicine that provides access to abstracts of articles published in medical journals.

504. Datla’s knowledge graph is built by extracting clinical concepts from narrative text in medical evidence sources, including these medical journals. Datla’s system processes this text to do so: “[T]he system utilizes a graph-building approach that centers on three steps: (i) topical keyword analysis in which the most clinically relevant keywords from the given topic descriptions, summaries, and clinical notes are identified; (ii) diagnostic inference using reasoning based on the topical

keywords to generate the diagnoses, tests, and treatments using the underlying clinical contexts represented within either a key-value memory network or a knowledge graph, both powered by an external clinical knowledge source” Ex. 1005 at ¶ [0034]. The diagnostic factors that populate the knowledge graph are thus extracted from, and discussed in, the narrative text of medical journals and other medical evidence sources that Datla processes.

505. Datla’s medical evidence sources characterize the extracted diagnostic factors as indicative of specific medical conditions. Datla describes the symptom–condition relationship in expressly indicative terms: “a symptom of a fever will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios.” Ex. 1005 at ¶ [0031]. A “symptom of” a condition is a diagnostic factor indicative of that condition. Datla’s system confirms this clinical directionality by performing “diagnostic inference,” i.e., inferring diagnoses from clinical concepts extracted from the source text. *Id.* at ¶ [0034]. This capability satisfies the requirement that “a first diagnostic factor entry is discussed in part of said at least one medical evidence journal as being indicative of a first medical condition entry.”

506. Element [32f]’s taxonomy language is the same as element [21f], except that it refers to “known and unknown diagnostic factors.” That difference does not affect my analysis. As I explained for Ground 5, the specification does not

describe a medical evidence journal that itself provides different taxonomies. *Supra* § X.E. For element [21f], I applied Patent Owner’s contention that MacRae meets this language. *Supra* § X.F.2.g (element [21f]); Ex. 1019 at 88–91. I incorporate that analysis. Under Patent Owner’s understanding, the combination satisfies this language because MacRae is part of the journal corpus from which the combined graph is built.

507. A POSITA would have been motivated to use Datla’s knowledge graph construction approach, including its use of medical journal articles as source material, to populate Zhong’s knowledge graph. Zhong discloses that “the knowledge graph may couple to external resources, such as various medical database or websites, for knowledge update such that the knowledge graph may be in line with current medical findings.” Ex. 1009 at 3:25–29. Zhong further cross-references a co-pending application for “obtaining symptom–disease correlations, which may be formed into one or more knowledge graph.” *Id.* at 3:29–38. But Zhong does not expressly describe how to build the knowledge graph or where its symptom–disease correlations originate.

508. As described above, Datla fills this gap by expressly disclosing the construction of a medical knowledge graph from a corpus that includes medical journals. A POSITA seeking to build or populate a knowledge graph of the kind Zhong describes would naturally look to Datla’s approach, because extracting

symptom–disease associations from published medical literature was a well-known, routine technique for constructing medical knowledge graphs at the time of the invention. Indeed, Zhong suggests such a combination with its teaching that the knowledge graph should be constructed so that it is “in line with current medical findings.” Ex. 1009 at 3:25-29. The combination uses each element for its known purpose: Zhong’s iterative question-generation pipeline operating on a knowledge graph built using Datla’s techniques. The resulting system whose knowledge graph is populated from medical journal articles is entirely predictable, and a POSITA would have had a reasonable expectation of success in making this combination.

h. Element [32g]

509. Element [32g] recites: “establish one or more application programming interfaces (APIs) for said graph database, said one or more APIs allowing an application to integrate with said graph database to access said known and unknown diagnostic factor entries that are linked to said medical condition entry in said graph database in order to generate a healthcare determination.”

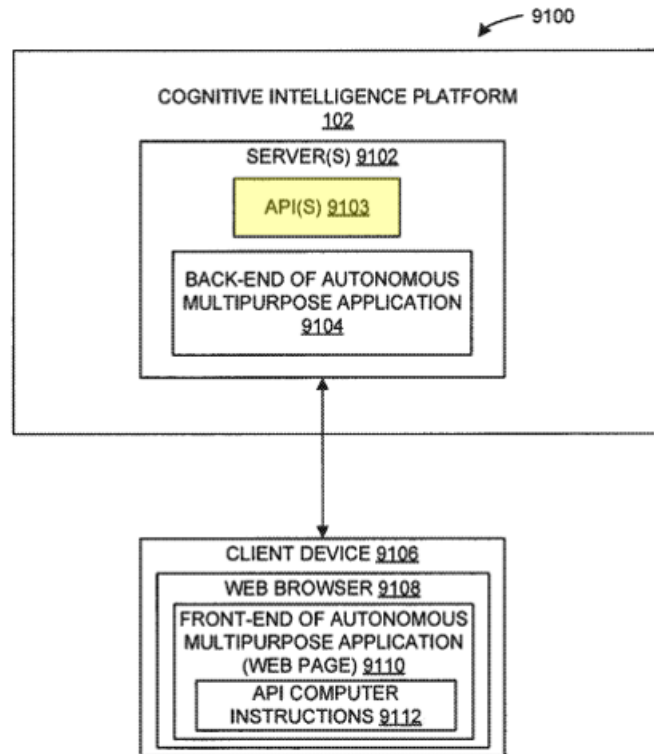
510. Zhong does not expressly disclose establishing an API allowing an application to integrate with said graph database to access said known and unknown diagnostic factor entries that are linked to said medical condition entry in said graph

database in order to generate a healthcare determination. Gnanasambandam discloses this element.

511. Gnanasambandam's API architecture publishes APIs that perform analytics based on a medical knowledge graph and makes them available to client devices running software applications. Ex. 1007 at 100:24–34, 101:12–25. Gnanasambandam expressly discloses “publishing, for implementation in the autonomous multipurpose application, at least one API that uses artificial intelligence to perform analytics based on a knowledge graph pertaining to ontological data of a medical condition, wherein the at least one API is included in a set of APIs.” *Id.* at 1:34–39; claim 1.

512. Gnanasambandam further discloses that “[a]pplication back-end component 9104 is configured to publish one or more APIs 9103. As described previously, the one or more APIs 9103 may use artificial intelligence to perform

analytics based on a knowledge graph pertaining to ontological data of a medical condition.” Ex. 1007 at 100:24–38; FIG. 91.



Ex. 1007 at FIG. 91 (annotated)

513. The APIs are provided to a client device (a computing device running a software application), which subscribes to and invokes them. Ex. 1007 at 102:36–42; FIG. 92, step 9206.

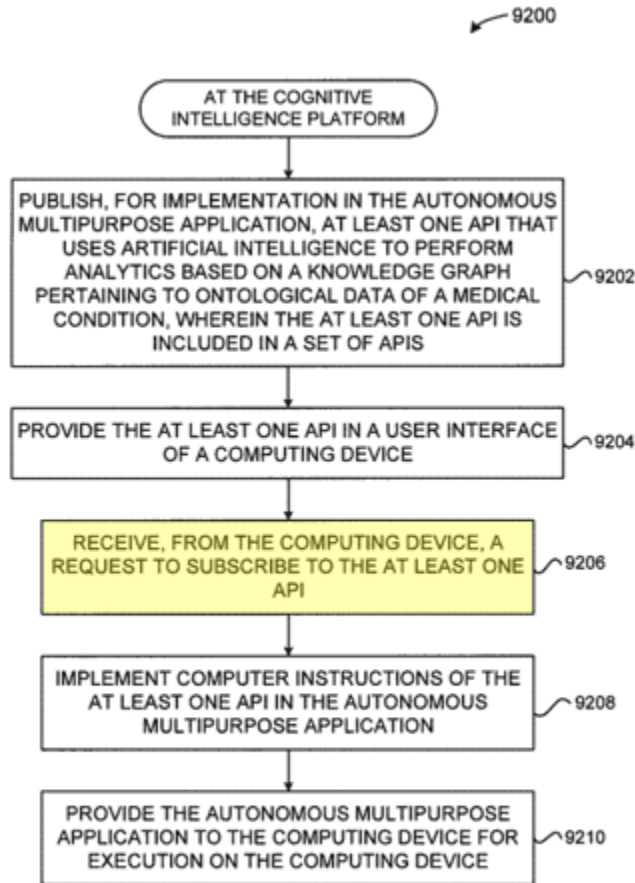


FIG. 92

Ex. 1007 FIG. 92 (annotated)

514. In the combined system, Gnanasambandam's API layer formalizes the input–output pathway for Zhong's system diagnostic engine, enabling external software applications to submit queries.

515. A POSITA would have been motivated to combine Zhong and Gnanasambandam. Zhong's system is expressly designed for external integration. Zhong discloses that its question generation system “may be an on-site system or a remote system communicating to the patient through various communication means,

such as telephone, Internet” (Ex. 1009 at 3:14–17), that its “input data and/or output data may be remotely transmitted from one physical location to another” and “programs that implement various aspects of this invention may be accessed from a remote location (e.g., a server) over a network” (*id.* at 7:36–40), and that the knowledge graph “may be a standalone component coupling to the question generation system” (*id.* at 2:61–63).

516. Zhong’s system is thus a remote, network-accessible, modular knowledge graph-based diagnostic service: exactly the type of system for which publishing APIs is the standard deployment mechanism. A POSITA building such a system would have recognized that an API is the conventional way to allow external applications, such as electronic health record systems, clinical decision support tools, or patient-facing diagnostic applications, to integrate with the knowledge graph, submit observed symptoms, and receive the generated next-best-action questions. Gnanasambandam’s express disclosure of publishing APIs for a medical knowledge graph (Ex. 1007 at 100:24–34) confirms that this was indeed the standard approach in the field.

i. Element [32h]

517. Element [32h] recites: “identify a rank of each unknown diagnostic factor entry that indicates a likelihood that said represented unknown diagnostic factor indicates said each medical condition, said rank being stored in a database

entry link between said unknown diagnostic factor entry and said medical condition entry.”

518. Zhong discloses a two-stage ranking process for candidate symptoms (the unknown diagnostic factors), both stages occurring after candidate-symptom identification at step 415.

519. **First-stage ranking (FIG. 4, Steps 420–425).** After identifying candidate symptoms, the system calculates their probabilities: “the probabilities of the identified candidate symptoms are calculated based on the likely disease.” Ex. 1009 at 4:52–54; FIG. 4. The equation is:

$$\max_{S_c} P(S_c | S_o)$$

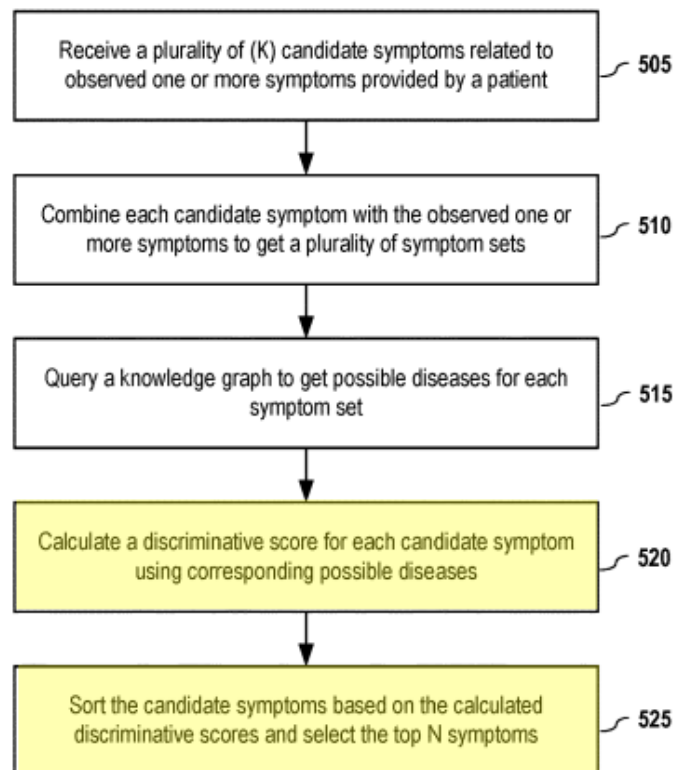
Id. at 4:58–61. The candidates are then sorted at step 425: “the identified candidate symptoms are sorted based on the calculated probabilities.” *Id.* at 4:64–66; FIG. 4. At step 430, the top-K candidates are selected from the sorted list for further refinement. *Id.* at 4:66–5:3.

520. **Second-stage ranking (FIG. 5, Steps 520–525).** The second stage refines the ranking using a discriminative scoring mechanism. For each of the top-K candidates, “a discriminative (or discriminability) score is calculated for each candidate symptom using corresponding possible diseases. In embodiments, the discriminative score is the maximized probability gap between top likely diseases

and unlikely diseases from the possible diseases.” Ex. 1009 at 5:33–38; FIG. 5. The equation is:

$$\max_{S_n} P(D_p, S_o \cup S_n) - P(D_n, S_o \cup S_n)$$

Id. at 5:39–42. “The candidate symptoms are sorted based on the calculated discriminative scores and the top-N candidate symptoms are selected among the K candidate symptoms as selected candidate symptoms.” *Id.* at 5:55–58; FIG. 5, Step 525.



Ex. 1009 at FIG. 5 (annotated)

521. This two-stage pipeline produces the “rank of each unknown diagnostic factor entry that indicates a likelihood that said represented unknown diagnostic factor indicates said each medical condition” required by the claim.

522. The claim requires the rank to be “stored in a database entry link between said unknown diagnostic factor entry and said medical condition entry.” Zhong’s knowledge graph stores a probability value on the edge between each symptom node and each disease node: “an edge that links a symptom node to a disease node has an associated symptom–disease pair value that represents a probability between the disease represented by the disease node given an observation of the symptom associated with the symptom node.” Ex. 1009 at Claim 1; 8:63–67. This symptom–disease pair value stored on the edge (database entry link) between the unknown diagnostic factor entry (symptom node) and the medical condition entry (disease node) is the rank indicating the likelihood. The probability and discriminative score calculations at Steps 420 and 520 use these edge-stored values as inputs to compute the rankings.

j. Element [32i]

523. Element [32i] recites: “display said unknown diagnostic factors represented by said unknown diagnostic factor entries in association with said each

medical condition in order of said rank of said each unknown diagnostic factor as said healthcare determination.”

524. After the candidates have been ranked at Step 330 and filtered for conflicts at Step 335, Zhong’s system generates questions and presents them to the patient or clinician at Step 340. FIG. 6, Step 620 provides the details of this step: “questions are generated based on the selected candidate symptoms without conflicts. In embodiments, if a symptom has related dimension information, questions related to that dimensional information may also be generated. The questions may be presented to the patient to collect answers.” Ex. 1009 at 6:13–19; FIG. 6, Step 620.

525. These generated questions present the unknown diagnostic factors to the user via the output interface: “The question generation system 140 may also send questions 114 to the patient via an output interface 130 to obtain more relevant information for more accurate diagnosis.” Ex. 1009 at 2:64–67; FIG. 1. Zhong expressly discloses output interface 130 as the display mechanism.

526. Element [32i] requires displaying the unknown diagnostic factors “in association with said each medical condition.” I address this sub-requirement in two ways: first, Zhong’s pipeline satisfies this limitation because the displayed factors are structurally associated with the medical condition through the system’s processing logic; second, to the extent “in association with” requires that the medical

condition be visually displayed alongside the unknown diagnostic factors, it would have been obvious to a POSITA to modify Zhong's display to include the associated medical condition, because Zhong's system already maintains that association internally.

527. Zhong's pipeline establishes an association between the unknown diagnostic factors and each medical condition. The unknown diagnostic factors are identified at Step 415 by traversing the knowledge graph from above-threshold disease nodes: "candidate symptoms (S_c) associated with the likely diseases are identified using the knowledge graph." Ex. 1009 at 4:46–49. The discriminative scoring at Step 520 is calculated "for each candidate symptom using corresponding possible diseases." *Id.* at 5:34–36. And the conflict check at Step 335/610 evaluates candidates against the observed symptoms for the current disease hypothesis. At every stage, the candidates are processed in the context of — and because of — a specific above-threshold medical condition. The questions generated at Step 340/620 are therefore "in association with" the medical condition because they exist only as a consequence of the system's having identified that medical condition as above-threshold.

528. To the extent the phrase "in association with said each medical condition" requires that the medical condition be visually displayed alongside the

unknown diagnostic factors, this additional feature would have been obvious to a POSITA in view of Zhong.

529. Zhong’s system already maintains the association between each unknown diagnostic factor and its corresponding above-threshold medical condition throughout the pipeline. At Step 415, the candidate symptoms are identified by traversal from specific above-threshold diseases. Ex. 1009 at 4:45–52. At Step 520, the discriminative scores are calculated “using corresponding possible diseases.” *Id.* at 5:33–37. Zhong expressly states that “[e]ach candidate symptom may be associate to one or more possible diseases. Similarly, each possible disease may associate to one or more candidate symptoms.” *Id.* at 4:48–52. Thus, the association data is already computed and available in Zhong’s system at the time of display. The only additional step required to visually display the medical condition alongside the unknown diagnostic factors is to include the condition’s name in the output, using association data that the system has already generated.

530. A POSITA would have found it obvious to include the medical condition in the display for at least the following reasons.

531. **Clinical utility and interpretability.** Displaying the associated medical condition alongside each unknown diagnostic factor provides clinical context to the clinician. A question about whether a patient has “double quotidian fever” is more informative when the clinician can see that the system is asking this

question because the patient is being evaluated for a specific suspected condition. In the context of differential diagnosis where multiple above-threshold conditions may be under consideration simultaneously, a POSITA would recognize that labeling which condition each question relates to is beneficial for the clinician to prioritize follow-up actions and understand the diagnostic reasoning. Displaying diagnostic recommendations organized by or labeled with their associated medical condition was a well-known convention in clinical decision support systems at the time of the invention.

532. **Minimal modification with predictable results.** Because Zhong’s system already tracks which above-threshold diseases each candidate symptom is associated with, displaying the condition name alongside the generated questions requires only that the system include in its output information it has already computed. This is a straightforward design choice in user interface presentation: the underlying data is unchanged, and the modification involves only choosing to display a piece of information that the system already possesses. Having questions displayed with their associated medical condition identified is entirely predictable and involves no technical challenge beyond routine software implementation.

533. Element [32i] requires displaying the unknown diagnostic factors “in order of said rank.” The specification of the ‘882 patent describes that unknown factors “may be listed as a group of the top actions to ask about or perform next or

may be listed in order of their likelihood of indicating the associated condition, with or without an explicit ranking.” Ex. 1001 at 30:25-28. The phrase “with or without an explicit ranking” confirms that the plain and ordinary meaning of “in order of said rank” does not require that a numerical ranking or score be visually displayed alongside each factor. It is sufficient that the factors are presented in an order that reflects their ranking. Zhong’s pipeline satisfies this through a sort-then-select-then-output sequence that logically preserves rank order at every stage.

534. At the first stage, “the identified candidate symptoms are sorted based on the calculated probabilities.” Ex. 1009 at 4:64–66; FIG. 4, Step 425. The top-K candidates are then selected from the sorted list. At the second stage, “the candidate symptoms are sorted based on the calculated discriminative scores and the top-N candidate symptoms are selected among the K candidate symptoms as selected candidate symptoms.” *Id.* at 5:55–58; FIG. 5, Step 525.

535. The selected candidates (the top-N from a sorted list) then pass through the conflict check at Step 335/610, which removes candidates contradicting observed data but does not reorder the remaining candidates. The surviving candidates proceed to Step 340/620 for question generation and output.

536. Selecting the top-N candidates from a sorted list and outputting them preserves rank order: the selection process takes candidates in descending order of their discriminative scores, and no subsequent step in Zhong’s pipeline reorders

them. A POSITA would understand that the output reflects the ranking even in the absence of an express statement that “questions are presented in ranked order,” because the sort→select→output pipeline is a standard algorithmic pattern that maintains the established ordering.

537. Even if element [32i] were found to require that the display visually convey the rank ordering to the user, for example, by listing factors in descending order of their scores, by numbering them, or by displaying explicit ranking values alongside each factor, and if it were found that Zhong does not disclose this, this additional feature would have been obvious to a POSITA in view of Zhong.

538. As discussed above, Zhong’s system already computes and maintains the ranking information. At Step 425, “the identified candidate symptoms are sorted based on the calculated probabilities.” Ex. 1009 at 4:64–66. At Step 525, “the candidate symptoms are sorted based on the calculated discriminative scores and the top-N candidate symptoms are selected.” *Id.* at 5:55–57. The discriminative scores and the sorted ordering are already computed at the time of display. Presenting the factors in a manner that visually conveys that ordering requires only that the system include in its output information it has already generated.

539. A POSITA would have found it obvious to display the unknown diagnostic factors in a manner that visually conveys their rank order for at least the following reasons.

540. **Clinical prioritization.** In a differential diagnosis workflow where multiple unknown diagnostic factors may be presented as next-best actions, the clinician must decide which factor to investigate first. Displaying the factors in rank order or with explicit ranking values enables the clinician to prioritize the most diagnostically informative actions. A POSITA would recognize that presenting ranked information in a manner that conveys the ranking is fundamental to the utility of any ranked output. Presenting ranked results in order of their ranking, or with explicit scores or ordinal numbers, was a ubiquitous convention in information retrieval and decision support systems at the time of the invention. Search engines, recommendation systems, and clinical scoring tools all presented results in descending order of relevance or score.

541. **Minimal modification with predictable results.** Because Zhong's system has already sorted the candidate symptoms by their discriminative scores before selecting the top-N for output, displaying those factors in rank order requires no additional computation; the system need only output the factors in the order they were selected. Adding explicit ranking values (such as ordinal numbers or scores) would similarly require only that the system include already-computed discriminative scores in the display output. Displaying unknown diagnostic factors

in a manner that visually conveys their ranking would have been entirely predictable and involves no technical challenge beyond routine implementation.

542. Element [32i] recites that the display occur “as said healthcare determination.” The ‘882 patent uses “healthcare determination” in a manner that encompasses a next-best-action (NBA) output. Dependent claim 41 confirms that “said healthcare determination is a recommendation, an evaluation, a result, or a conclusion.” The specification describes the healthcare determination output as including ranked conditions, unknown factors that should be queried, and investigations that should be performed: “the NBA engine 70 executes . . . and produces the likely, ranked conditions of the patient, along with for each, unknown factors that should be queried and investigations that should be performed.” Ex. 1001 at 9:7–11. The specification further explains that “the next-best actions for a particular condition may be a question to ask the patient about a specific diagnostic or risk factor, or a specific laboratory test or examination to be performed upon the patient.” Ex. 1001 at 9:13–17.

543. In Zhong, the generated questions asking the patient about the presence or absence of previously unknown diagnostic factors are next-best-action recommendations for the clinician. A POSITA would understand that Zhong’s questions about the highest-ranked candidate symptoms constitute a “healthcare determination” because they are the system’s determination of what clinical

information should be gathered next to advance the diagnosis. The display of unknown diagnostic factors in order of rank is at minimum a recommendation (recommending which symptoms to inquire about next) and a result (the output of the discriminative scoring process). Each of these independently qualifies as a “healthcare determination” under the ‘882 patent’s own usage of the term, as confirmed by dependent claim 41’s disjunctive recitation of “a recommendation, an evaluation, a result, or a conclusion.”

3. Dependent Claim 33

544. Dependent claim 33 recites: “A computer system as recited in claim 32, wherein the instructions, when executed, further cause the one or more processors to: identify said unknown diagnostic factor entries by traversing down through said graph database from said each medical condition entry and disregarding said known diagnostic factor entries.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

545. Zhong expressly discloses this element. Zhong identifies unknown diagnostic factor entries (candidate symptoms) by traversing from above-threshold disease nodes outward to connected symptom nodes in the knowledge graph. Ex. 1009 at 4:45–52; FIG. 4, Step 415. Zhong defines candidate symptoms as the complement of the observed (known) set: “a candidate symptom may be an unobserved symptom or a symptom not reported to the automatic diagnosis system.”

Id. at 3:52–54. By defining candidate symptoms as those not in the known set, the system disregards all known diagnostic factor entries.

546. Zhong further enforces this strategy by expressly checking for and removing conflicts with observed symptoms: “the selected candidate symptoms are checked for conflicts with observed symptoms and any selected candidate symptom with a conflict is removed from further consideration.” Ex. 1009 at 4:14–18; FIG. 3, Step 335. This two-step process that involves identifying candidates as the complement of the known set, then removing any conflict with observed data is traversing down through the graph database from the medical condition entry and disregarding known diagnostic factor entries.

4. Dependent Claim 38

547. Dependent claim 38 recites: “A computer system as recited in claim 32 wherein traversing said graph database from one of said unknown diagnostic factor entries to a first one of said medical conditions produces a first rank for said one of said unknown diagnostic factor entries in the context of said first medical condition which is different from a second rank produced by traversing from said one of said unknown diagnostic factor entries to a second one of said medical conditions.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

548. Zhong discloses this element. In Zhong’s knowledge graph, each edge from a symptom node to a disease node carries its own “symptom–disease pair value” representing the probability of the disease given the symptom. Ex. 1009 at Claim 1. Because a single symptom node can link to multiple disease nodes through separate edges (*id.* at 4:48–52), traversing from one unknown diagnostic factor entry to a first medical condition follows one edge and produces a first pair value (rank), while traversing from the same factor to a second medical condition follows a different edge and produces a second pair value (rank). These pair values will generally differ because the probability of a given symptom indicating disease A is different from the probability of that same symptom indicating disease B. This is the fundamental design of probabilistic knowledge graphs.

549. The specification of the ‘882 patent provides an express example using the fever EDF: “the noise value for fever 310 will be the same for condition 300 as well as condition 340, although the rank of factor groups 304 and 344 will be different.” Ex. 1001 at 25:12-14. Factor group 304 holds fever’s rank in the context of condition 300, and factor group 344 holds a different rank for the same fever in the context of condition 340. Zhong’s edge-stored pair values function identically: each edge carries a condition-specific probability, and different edges from the same symptom node to different disease nodes carry different probabilities. A POSITA

would understand that Zhong’s architecture necessarily produces different ranks for the same unknown diagnostic factor in the context of different medical conditions.

5. Dependent Claim 39

550. Dependent claim 39 recites: “A computer system as recited in claim 32 wherein said graph database includes more than twenty thousand entries.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

551. Claim 39 is the same as claims 12 and 29; only the parent claim differs. The combined system’s graph is built with Datla’s techniques from a corpus covering clinical medicine. *Supra* § X.I.2.g (element [32f]). My routine-scaling and duplication-of-parts analysis for claims 12 and 29 therefore applies equally here. *Supra* §§ X.A.8, X.F.7. Zhong’s graph holds the same kinds of entities, “a symptom, a dimension, a demographic, and a disease.” Adding more of them changes neither Zhong’s inference nor its question generation behavior. Ex. 1009 at 3:20–25.

6. Dependent Claim 40

552. Dependent claim 40 recites: “A computer system as recited in claim 32 wherein said graph database includes more than one hundred thousand entries.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

553. Claim 40 is the same as claims 13 and 30. For the reasons given for claim 30, I rely on the scope of Datla's graph and on routine scaling. *Supra* § X.F.8. The '882 patent states that "an actual graph database will be much more complex and contain many hundreds of thousands of nodes or more," and it ties no benefit to the one-hundred-thousand threshold. Ex. 1001 at 27:67–28:3.

7. Dependent Claim 41

554. Dependent claim 41 recites: "A computer system as recited in claim 32 wherein said healthcare determination is a recommendation, an evaluation, a result, or a conclusion." I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

555. As discussed in my analysis of element [32i] above, the questions generated by Zhong's system asking the patient about the presence or absence of previously unknown diagnostic factors in order of their ranking constitute a "healthcare determination" within the meaning of this claim. The display of unknown diagnostic factors in order of rank is at minimum a recommendation (recommending which symptoms to inquire about next) and a result (the output of the discriminative scoring process). Each of these independently qualifies under the disjunctive recitation of "a recommendation, an evaluation, a result, or a conclusion."

J. Ground 10: Claim 34 is Obvious Over Zhong, Datla, Gnanasambandam, MacRae, and Dang

556. In my opinion, claim 34 of the ‘882 patent is obvious over Zhong, Datla, Gnanasambandam, MacRae, and Dang.

557. For the reasons I discuss for independent claim 32, a POSITA would have been motivated to combine Zhong, Datla, and Gnanasambandam. *Supra* § X.I.1. A POSITA would also have been motivated to incorporate Dang’s teachings into that combination, for three reasons.

558. Dang, like Zhong, concerns knowledge-graph-based diagnosis that uses a patient’s symptoms to identify likely diseases and then generates the next question to ask. Zhong “generate[s] a set of possible diseases . . . by querying a knowledge graph that comprises symptom–disease associations” and then generates questions to collect information about candidate symptoms. Ex. 1009, Abstract; claim 1. Dang “[e]valuate[s] a likelihood value for one or more diseases based on the knowledge graph data structure” and “[g]enerate[s] a best next inquiry question for use in a medical diagnosis process.” Ex. 1008, Abstract. A POSITA working on Zhong’s system would naturally have looked to Dang.

559. Both references attach conditional probabilities to the relationship between a disease and a finding. Zhong uses a symptom–disease pair value. Zhong, claim 1. Dang uses a conditional probability $P(f_x|d_n)$ that “indicates, for a pair of disease and feature (d_n, f_x) , the likelihood of the feature f_x occurring when the

disease d_n [is] observed.” Ex. 1008 at ¶ [0031]. Dang’s techniques therefore apply directly to Zhong’s graph without changing the kind of information it holds.

560. Zhong’s nodes are not limited to symptoms and diseases. A node “is an entity, such as a symptom, a dimension, a demographic, and a disease.” Ex. 1009 at 3:20–25. Zhong does not prescribe a particular internal structure for the graph and refers to a co-pending application for obtaining symptom–disease correlations. *Id.* at 3:29–38. A POSITA would have understood that Zhong’s graph could incorporate additional node types, such as those Dang teaches.

1. Dependent Claim 34

561. Dependent claim 34 recites: “A computer system as recited in claim 32 wherein said rank of said each unknown diagnostic factor entry is located within a factor group entry located between said medical condition entry and said each unknown diagnostic factor entry.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

562. As I explained for dependent claim 16, Dang discloses intermediate feature nodes located between disease nodes and symptom nodes. *Supra* § X.D.3. Dang’s knowledge graph 100B “includes the following three sets of nodes: a set of factual nodes (symptom nodes, test nodes and context nodes), a set of feature nodes, and a set of disease nodes.” Ex. 1008 at ¶ [0028]. Symptom nodes connect to features, and disease nodes connect to feature nodes. *Id.* at ¶¶ [0030]–[0031]; FIG.

1B. Each feature node belongs to the feature set of a particular disease. *Id.* at ¶ [0032]. I incorporate that analysis by reference here.

563. A POSITA would have understood Dang’s feature node to be a “factor group entry.” A Dang feature node collects one or more factual nodes, including symptom nodes, and connects them to a disease. Ex. 1008 at ¶¶ [0028]–[0031]; FIG. 1B. In FIG. 1B, for example, symptom nodes s_1 and s_2 and context node c_1 connect to feature node f_1 . *Id.* at FIG. 1B. The ‘882 patent’s factor group plays the same role. It groups EDFs for a condition and holds their condition-specific rank. Ex. 1001 at 20:60–21:8.

564. As I also explained for claim 16, Dang associates $P(f|d)$ with the connection between a disease node and a feature node. Ex. 1008 at ¶ [0031]. It would have been obvious to hold that value, the rank of the linked diagnostic factors for that disease, within the feature node itself. *Supra* § X.D.3. Because each feature node is specific to one disease (Ex. 1008 at ¶ [0032]), moving the value from the connection into the node changes neither its meaning nor how it is used. Claim 34 requires only that the factor-level rank be located within the factor group entry, and it does not require a ranking of the patient’s conditions to be determined by reference to that entry.

565. A POSITA would have been motivated to add Dang’s intermediate feature nodes to Zhong’s knowledge graph, and to hold the rank within them, for at least four reasons.

566. First, Zhong’s method depends on distinguishing between diseases. Zhong selects candidate symptoms by calculating a discriminative score, “the maximized probability gap between top likely diseases and unlikely diseases from the possible diseases.” Ex. 1009 at 5:35–38. A POSITA would have recognized that grouping a disease’s findings into features, and holding the disease-specific value for each group, would give Zhong’s scoring a structured, disease-by-disease view of which findings bear on which disease. Dang’s formulas break each disease’s score into feature-level contributions. Ex. 1008 at ¶¶ [0049]–[0050]. This makes it easier to see why one candidate symptom separates likely diseases from unlikely ones better than another.

567. Second, Zhong already represents information about a symptom’s context in the graph. Its nodes include “a dimension” and “a demographic.” Ex. 1009 at 3:20–25. Zhong also uses dimensional information such as “frequency, intensity, duration, severity” when generating questions, and teaches that it “may be obtained from a knowledge graph.” *Id.* at 6:8–12. Dang’s feature nodes combine symptom and context nodes before they reach the disease. Ex. 1008 at FIG. 1B; ¶¶ [0028]–[0030]. A POSITA would have recognized that Dang’s feature nodes provide a

structured way to associate a disease with a group of related findings and their context, which is a natural fit for the kinds of entities Zhong's graph already contains.

568. Third, holding the disease-specific rank in an intermediate entry keeps each symptom node condition-independent, as element [32d] requires. *Supra* § X.I.2.e. The combined Zhong–Datla system already keeps condition-specific values off the symptom node by placing them on edges. *Supra* § X.I.2.e. Holding those values in a disease-specific feature node preserves that separation. It also lets the same symptom node participate in feature groups for many diseases.

569. Fourth, the choice between holding a value on an edge and holding it in an intermediate node was a routine design choice. Knowledge graphs were commonly built either as labeled property graphs, in which edges carry properties and their values, as in Zhong, or as triple stores, in which each item of information is a node. A POSITA would have understood that a value carried on an edge can equally be held in a node placed on that link.

570. A POSITA would have reasonably expected success. Zhong calculates disease probabilities from “symptom–disease pair values obtained from the knowledge graph” using a graph inference model. Ex. 1009, claim 1. Zhong's graph inference would read the same value off of the feature node instead of from the edge. Storing a numerical value as a property of a graph node, and reading it during

traversal, was a routine graph-database operation. Dang shows that a symptom–disease knowledge graph with intermediate feature nodes and disease-specific conditional probabilities could be built and used for diagnosis. Ex. 1008 at ¶¶ [0028]–[0031], [0049]–[0050].

K. Ground 11: Claims 35 and 36 are Obvious Over Zhong, Datla, Gnanasambandam, MacRae, and Kim

571. In my opinion, claims 35 and 36 of the ‘882 patent are obvious over Zhong, Datla, Gnanasambandam, MacRae, and Kim.

572. For the reasons given for Ground 9, a POSITA would have combined Zhong, Datla, Gnanasambandam, and MacRae. *Supra* § X.I.1. For the reasons given for Grounds 2 and 7, a POSITA would have added Kim’s ontology-annotated graph entries. *Supra* §§ X.B.1, X.G. Those reasons apply equally with Zhong as the primary reference. Zhong’s graph is updated from “various medical database or websites.” Ex. 1009 at 3:25–29. A graph drawn from multiple sources benefits from Kim’s identifiers and synonyms, because they let the same concept be recognized however a source describes it. They also prevent redundant concepts. Ex. 1012 at ¶¶ [0050], [0054].

1. Dependent Claim 35

573. Dependent claim 35 recites: “A computer system as recited in claim 32, wherein the instructions, when executed, further cause the one or more processors to: for each known diagnostic factor, traverse upward in a medical ontology link

found in said identified known diagnostic factor entry in order to infer other known diagnostic factors present in said patient.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

574. For the reasons given for claims 4 and 23, it would have been obvious to include a unique ontology identifier in each diagnostic factor entry, as Kim teaches, and I incorporate that analysis. *Supra* §§ X.B.2, X.G.1. Kim’s entries “may be identified by a unique ontology code” and “may themselves be annotated with identifiers from several different ontologies.” Ex. 1012 at ¶¶ [0041], [0051]. That identifier is a “medical ontology link found in said identified known diagnostic factor entry.” Zhong’s known diagnostic factors are its observed symptoms, which are stored in the same symptom nodes as every other symptom. *Supra* § X.I.2.d (element [32c]).

575. The link brings the ontology’s hierarchy into the graph. Kim teaches using the identifier to reach other concepts in the graph: “[o]nce an ontology code (unique identifier) has been assigned . . . related concepts as represented in the graph database can be suggested,” based on factors including “proximity within the graph database.” Ex. 1012 at ¶ [0052]. Medical ontologies organize concepts through ancestor–descendant relationships, and these relationships “enable traversal and reasoning across the graph.” *Supra* § VI.C. Datla likewise adds an edge between two nodes “according to a hierarchical relation of the corpus of medical information.”

Ex. 1005, claim 16. In the combined system, then, each known diagnostic factor entry is linked through its ontology identifier to its parent concepts, and the graph holds those parent–child relations as edges.

576. Zhong identifies likely diseases by running a graph inference algorithm, “such as Markov chain Monte Carlo (MCMC) or Gibbs sampling,” over the observed symptoms. Ex. 1009 at 4:40–44. That algorithm traverses the edges connected to the observed symptom nodes to compute probabilities. *Supra* § X.I.2.d (element [32c]). Zhong repeats this every time the observed symptoms are updated, “until a disease is diagnosed or the patient reports no more observed symptoms.” Ex. 1009 at 4:29–35. In a graph that holds the ontology’s parent–child edges, each pass of that inference follows the edge from a known finding up to its parent. The algorithm does not distinguish hierarchical edges from symptom–disease edges; it propagates evidence along every connected edge. A patient known to have a specific finding, such as bacterial pneumonia, is also known to have its parent concept, pneumonia. By following the upward link, the system infers the parent finding as present. That is traversing upward in the ontology link “in order to infer other known diagnostic factors present in said patient.”

577. Even if inferring the parent were not the automatic result of Zhong’s iteration, it would have been obvious. Zhong queries its graph with the observed symptoms. *Supra* § X.I.2.d. Inferring the parent findings of those symptoms captures

disease associations recorded at a more general level. It also keeps an already-known finding from being asked about as a candidate symptom. *Supra* § X.I.3 (claim 33). Following a stored hierarchical link is a routine graph operation, and it leaves Zhong's scoring and question generation unchanged.

2. Dependent Claim 36

578. Dependent claim 36 recites: “A computer system as recited in claim 32 wherein each said known and unknown diagnostic factor entry includes at least one synonym of said represented known and unknown diagnostic factor.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

579. Claim 36 is substantively the same as claim 24, except that it requires synonyms in both the known and the unknown diagnostic factor entries. For the reasons given for claim 24, *supra* § X.G.2, it would have been obvious to include at least one synonym in each diagnostic factor entry, as Kim teaches, and I incorporate that analysis. Kim teaches that “the graph database includes synonyms, translations, concepts and relationships.” Ex. 1012 at ¶ [0054]. In Zhong, a symptom is “known” or “unknown” only relative to a particular patient, and both kinds are stored in the same symptom nodes. *Supra* §§ X.I.2.d, X.I.2.f (elements [32c], [32e]). Adding synonyms to each symptom node therefore adds them to both. Synonyms also help Zhong identify unknown factors. A candidate symptom whose synonym already

appears among the observed symptoms is recognized as known and is not asked about. Ex. 1009 at 3:52–54.

L. Ground 12: Claim 37 is Obvious Over Zhong, Datla, Gnanasambandam, MacRae, and Van Assel

580. In my opinion, claim 37 of the ‘882 patent is obvious over Zhong, Datla, Gnanasambandam, MacRae, and Van Assel.

581. For the reasons given for Ground 9, a POSITA would have combined Zhong, Datla, Gnanasambandam, and MacRae. *Supra* § X.I.1. For the reasons given for Ground 8, a POSITA would have added Van Assel’s present/absent attribute. *Supra* § X.H.1. Those reasons apply equally with Zhong as the primary reference. Like Zhong, Van Assel performs knowledge-graph-based diagnosis. Zhong’s system also already tracks each symptom’s status as observed or not observed. Ex. 1009 at 3:42–44, 3:52–54.

1. Dependent Claim 37

582. Dependent claim 37 recites: “A computer system as recited in claim 32 wherein each said known and unknown diagnostic factor entry includes a temporality attribute, a negation attribute or an experiencer attribute.” I incorporate by reference my discussion of independent claim 32 above. *Supra* § X.I.2.

583. Claim 37 is substantively the same as claims 6 and 25, except that it covers both known and unknown diagnostic factor entries. The claim is disjunctive,

and I address the negation attribute. I incorporate my analysis of claims 6 and 25. *Supra* §§ X.C.1, X.H.1.

584. Zhong’s known diagnostic factors are the patient’s observed symptoms, including symptoms the patient denies in answer to a generated question. *Supra* § X.I.2.d (element [32c]). Zhong’s unknown factors are candidate symptoms, which are “unobserved” or “not reported.” Ex. 1009 at 3:52–54. Datla likewise determines whether an extracted concept is negated. Ex. 1005 at ¶ [0054].

585. Van Assel “assigns 1, -1, or 0 to the extracted concept to indicate whether the concept is validly present (1), validly absent (-1) or this is unknown (0).” Ex. 1011 at ¶ [0245]. In the combined system, each known entry would carry 1 or -1, and each unknown entry would carry 0. Every known and unknown diagnostic factor entry would therefore include a negation attribute indicating whether the factor is present or not present.

586. The stored value also simplifies two operations Zhong already performs. First, Zhong defines its candidates as the complement of the observed set. Ex. 1009 at 3:52–54. Second, Zhong removes candidates that conflict with observed symptoms. *Id.* at 4:14–18. A stored present/absent/unknown value turns each of these operations into a simple check of the attribute. Storing a ternary value as a node property is routine, and it leaves Zhong’s inference, scoring, and question

generation unchanged. A POSITA would therefore have reasonably expected success.

M.Ground 13: Claims 1-9 and 11-41 are Directed to Ineligible Subject Matter

587. In my opinion, claims 1-9 and 11-41 of the '882 patent are directed to ineligible subject matter under 35 U.S.C. § 101. The challenged claims are directed to the abstract idea of collecting, organizing, storing, and providing access to medical data using conventional computer components and well-known techniques, without any inventive concept.

588. While there are differences between independent claims 1, 15, 21, and 32, the abstract idea in these claims is the same. Below, I address the commonality among these claims, then independent claim 1 in full, then claims 15, 21, and 32, to the extent that they contain different limitations relative to claim 1.

1. The Challenged Claims are Directed to an Abstract Idea

a. The Independent Claims Share the Same Core Features

589. Each independent claim of the '882 patent recites the same three core features:

- **A graph database.** The graph database links diagnostic factors to medical conditions.

- **Condition-independent diagnostic factor entries.** Each claim requires that the diagnostic factor entry or node “not include information specific to” the linked medical condition.
- **An API.** Each claim requires establishing “one or more application programming interfaces (APIs) for said graph database,” allowing an application to integrate with the graph database to access the linked entries “in order to generate a healthcare determination.”

590. I note that during prosecution of the ‘882 patent, the applicant added the API limitation to every independent claim in response to the Examiner’s § 101 rejection. Ex. 1002 at 57–67.

591. The independent claims differ in what they do with these common core functionalities:

- **Claim 1** is directed to “integrating medical evidence text into a computational form.” It receives electronic text from medical evidence sources, retrieves diagnostic factors, and stores the factors and conditions as cross-linked entries.
- **Claim 15** is directed to “determining a state of a patient.” It adds a demographics node and matches patient chart data to the graph to display a healthcare determination.

- **Claim 21** is also directed to “determining a state of a patient.” It receives a portion of the patient’s EHR, matches the patient’s diagnostic factors to the graph, collects and sums weight values, and displays conditions in order of total weight.
- **Claim 32** is directed to “determining next-best actions for a medical condition.” It identifies unknown diagnostic factors linked to conditions above a probability threshold and displays them in order of a rank stored in the link.

592. I note that the Examiner’s reasons for allowance track these same four groups. For each group, the Examiner relied on the combination of the claimed data arrangement with the API limitation. Ex. 1002 at 17–20.

b. Independent Claim 1

593. Independent claim 1 recites a computer system that performs the following functional steps:

- Receive electronic text in a plurality of medical evidence sources, at least a subset of which includes a published journal [1b];
- Retrieve diagnostic factors associated with medical conditions [1c];
- Where the medical evidence sources provide different taxonomies [1d], create a graph database cross-linking the diagnostic factors and medical conditions [1e];

- Store the diagnostic factors and medical conditions as separate, linked entries [1f], with the diagnostic factor entry not including information specific to the linked medical condition [1g]; and
- Establish APIs allowing an application to integrate with the graph database to access the linked entries in order to generate a healthcare determination [1h].

594. The only additional computing elements are “one or more processors” and “memory that stores instructions” in element [1a]. These are generic components that the specification describes in conventional terms. Ex. 1001 at 31:65–32:14; FIGS. 11A–11B (describing a computer system that “may have many physical forms,” with “[p]rocessor(s) 922” coupled to “memory 924”). The Examiner likewise treated the processors and memory as generic components. Ex. 1002 at 94–96.

595. Claim 1 is directed to collecting medical evidence from published sources, identifying the diagnostic factors associated with each medical condition, organizing those relationships as linked entries, and making them available to other software. The focus of the claim is not on any specific improvement to computer technology. Claim 1 claims the functional result of making medical evidence accessible and queryable. The specification describes the problem in those terms, acknowledging that publishers already assembled published evidence into diagnostic reference content through human-centric activity. Ex. 1001 at 1:59–64.

596. The core activities described in claim 1 are processes that medical professionals and informaticists have performed, in substance, for decades. These include reading medical literature, identifying the signs and symptoms associated with each condition, recording those associations in structured references, and keeping each finding as a reusable concept that can be associated with many conditions. Before the ‘882 patent’s priority date, clinicians and medical informaticists routinely did this, and organized the results in structured taxonomies and knowledge bases. *Supra* §§ VI.A–VI.D. Claim 1 automates this well-established process using conventional computing components without improving any of those components.

597. The graph database recited in claim 1 is a conventional data structure. The claim does not specify a particular way of storing, indexing, or retrieving the entries. As I explained in my Technology Background, knowledge graphs organizing medical conditions and their associated findings were well established before the priority date. *Supra* § VI.C. Datla, for example, discloses a knowledge graph of symptom and condition nodes connected by weighted edges within a system that includes a database. Ex. 1005 at ¶¶ [0031], [0078]; Ex. 1001 at 8:10–18 (“[a]ny of a wide variety of other types of databases may be used to store this information such as a relational database or object store or object-oriented database,” including a JanusGraph graph database).

c. Independent Claim 15

598. Claim 15 shares claim 1's core features as it recites extracting diagnostic factors from medical evidence sources, creating a cross-linked graph database, storing each factor and condition as its own node with condition-independent factor nodes, and establishing APIs (elements [15b]–[15e], [15g]). My analysis of claim 1 applies to those limitations. *Supra* § X.M.1.b.

599. Claim 15 adds two limitations:

- Storing, for at least one diagnostic factor, the “demographics required to indicate one of said medical conditions” in a demographics node located between the condition node and the factor node [element 15f]; and
- Matching “patient data from a patient medical chart” to the diagnostic factor nodes and the demographics node “in order to display said healthcare determination that said patient has one of said medical conditions” [element 15h].

600. These limitations describe what a clinician does when deciding whether a patient's findings point to a particular condition. The clinician compares the patient's symptoms with the findings known to be associated with the condition and confirms that the patient's age or sex is consistent with the condition presenting that way. The '882 patent describes this same comparison: demographics nodes “corresponding to matched EDF nodes are taken into account when reasoning occurs

to determine if the patient has the particular condition related to that EDF node.” Ex. 1001 at 24:40–44.

601. Comparing a patient’s findings and demographics with stored knowledge and reporting a likely diagnosis is data gathering, comparison, and reporting. It is not a technological improvement. Goldberg confirms that patient demographics and history were understood to be patient data compared against stored knowledge to generate a diagnosis. Ex. 1006 at 8:14–16. Claim 15 is therefore directed to the same abstract idea as claim 1.

d. Independent Claim 21

602. Claim 21 shares many of the same features as claim 1: a cross-linked graph database whose diagnostic factor entries do not include condition-specific information, and APIs (elements [21d], [21h]). Claim 21 adds the following steps:

- Receive a portion of a patient EHR to produce diagnostic factors known to be present or absent [21b];
- Match those factors to entries in the graph database [21c];
- Traverse the link from each matched factor to its linked condition and collect a weight value [21i];
- Sum the collected weights for each condition [21j]; and
- Display the linked conditions in order of total weight [21k].

603. These steps describe tallying the evidence for each candidate diagnosis. A clinician does this when reviewing a chart, noting which findings are present or absent, considering how strongly each finding points to each condition, and ranking the possible conditions accordingly. The summing step in element [21j] is nothing more than basic arithmetic. Dang expresses the same operation as “Confidence of disease=Sum(Confidence of evidence*TF-IDF).” Ex. 1008 at ¶ [0050]. Displaying the conditions in ranked order is reporting the result of that tally.

604. Claim 21 also requires that the entries originate in at least one medical evidence journal that provides different taxonomies in elements [21e]–[21g]. These limitations specify only the content and source of the information. As I explained in Section X.E, the specification does not describe any journal as providing taxonomies. It describes ontologies such as SNOMED-CT as the structures that provide taxonomies. *Supra* § X.E. Under any reading, specifying where the medical information comes from does not change what the claim is directed to. Claim 21 is therefore directed to the same abstract idea as claim 1.

e. Independent Claim 32

605. Claim 32 shares many of the same features as claim 1: diagnostic factor entries linked to medical condition entries and not including condition-specific information, and APIs. Claim 32 adds the following steps:

- For each condition with a probability above a threshold [32b], identify the known diagnostic factor entries [32c] and the unknown diagnostic factor entries linked to that condition [32e];
- Identify a rank for each unknown factor stored in the link between the factor and the condition [32h]; and
- Display the unknown factors in association with each condition, in order of rank, as the healthcare determination [32i].

606. These steps describe the ordinary next step in differential diagnosis: identifying which findings have not yet been elicited for the conditions under consideration, and deciding which to ask about first. A clinician who suspects a condition asks about the findings that would best confirm or rule it out, beginning with those most strongly associated with the condition. The '882 patent describes the output in these terms: the NBA engine produces “the likely, ranked conditions of the patient, along with for each, unknown factors that should be queried and investigations that should be performed.” Ex. 1001 at 9:7–11. A next-best action

“may be a question to ask the patient about a specific diagnostic or risk factor.” *Id.* at 9:13–15.

607. Applying a threshold, identifying unasked questions, ranking them, and displaying them is analysis and reporting of information. Claim 32 also requires that the entries originate in a medical evidence journal providing different taxonomies (element [32f]), which again specifies only the source and content of the information. Claim 32 is therefore directed to the same abstract idea as claim 1.

f. The Claimed Arrangements Do Not Improve Computer Functionality

608. During prosecution, the applicant argued that the claims described “a specific data architecture (similar to the *Enfish* case) that can integrate and normalize information from different medical evidence sources.” Ex. 1002 at 68. I disagree. Each data arrangement recited in the independent claims decides only where a piece of medical information is recorded. None changes how the database stores, retrieves, or processes data. I address each arrangement below.

609. **Condition-independent diagnostic factor entries (claims 1, 15, 21, 32).** Keeping a diagnostic factor entry free of condition-specific information, so that one entry can be linked to many conditions, was a routine design principle in knowledge engineering. As one of many examples, Datla’s graph already follows this architecture: a single symptom node “will be connected by edges to hundreds of other nodes, as a fever is a symptom of many patient scenarios,” and the relationship

between two nodes is carried on the weighted edge. Ex. 1005 at ¶¶ [0031]–[0032]. Zhong likewise places condition-specific probability values on edges rather than in symptom nodes. Ex. 1009 at 3:20–25, claim 1. Deciding which data to put in a node, and which on an edge, is a choice about content, not an improvement to the database.

610. Demographics node between condition and factor nodes (claim 15).

Adding a node type to hold an attribute value such as an age range or sex, and comparing it with a patient’s demographics, was a routine graph-database technique. For example, Dang’s meta-model already includes a context node linked to both a disease node and a symptom node. Ex. 1008 at ¶ [0027]; FIG. 1A. The ‘882 patent describes the purpose of the demographics node as a matter of data placement: demographics “may be held separate from the EDF in a demographics node, thus also ensuring that the EDF remains independent and applicable to any of a number of conditions.” Ex. 1001 at 11:59–64.

611. Weight values on links (claim 21) and ranks stored in links (claim 32). Storing a numerical value on an edge in a knowledge graph and reading it during traversal was routine. Whether a value sits on an edge or in an intermediate node was likewise a routine design choice. Knowledge graphs were commonly built either as labeled property graphs, in which edges carry properties, or as triple stores, in which each item of information is a node. Dang associates its conditional probability $P(f|d)$ with the connection between a disease and a feature, and Zhong stores a

symptom–disease pair value on each edge. Ex. 1008 at ¶ [0031]; Ex. 1009 at claim 1.

612. These arrangements are not improvements to database technology. They do not make the database faster, smaller, or more reliable, and they do not change how queries are executed. They specify the medical content of the entries and where that content is recorded.

613. The applicant also argued that the claims “integrate and normalize information from different medical evidence sources.” Ex. 1002 at 68. Integrating information from multiple medical sources into a single knowledge graph was a widely practiced approach before the priority date. *Supra* § VI.C. I note that the claims recite no particular technique for normalizing that information.

g. The Prosecution History and API Limitation

614. The prosecution history confirms that the claims are directed to an abstract idea. The Examiner rejected claims 1–30 as reciting steps that “can be practically performed by a human, either mentally or with the use of pen and paper,” placing them in the “Mental Processes” grouping of abstract ideas. Ex. 1002 at 94–95. The Examiner found that the additional elements amounted to insignificant extra-solution activity and did not integrate the abstract idea into a practical application or

provide an inventive concept. *Id.* at 95–96. The Examiner applied that analysis separately to claims 1–11, claims 12–23, and claims 24–30. *Id.* at 93–103.

615. The applicant responded by adding the API limitation to every independent claim, and argued that it overcame the rejection. Ex. 1002 at 57–69. The Examiner then allowed the claims. For each claim group, the Examiner’s reasons for allowance rest on the combination of a data arrangement with the API limitation. *Id.* at 17–20.

616. The API limitation involves establishing an API that allows an application to integrate with the graph database to access the linked entries. An API is a generic software interface that enables communication between software components. As I explained in my Technology Background, exposing medical knowledge bases through APIs was a well-established design pattern before the priority date. *Supra* § VI.E. The ‘882 patent only identifies standard, off-the-shelf APIs: “FHIR, USCDI (United States Core Data for Interoperability), CDS Hooks, and vendor-specific APIs.” Ex. 1001 at 6:4–8.

617. The applicant argued that the API limitation overcame the § 101 rejection because “an API is a software construct that a human cannot practically perform in his or her mind.” Ex. 1002 at 68–69. I disagree. I understand that the relevant question is whether the claim as a whole is directed to an abstract idea, not whether every individual step can be performed mentally. In other words, the fact

that a human cannot literally implement an API does not make the claims non-abstract. The underlying process is the one clinicians and medical informaticists have always performed: organizing medical knowledge, comparing a patient's findings against it, and reporting a likely diagnosis or the next question to ask. The API only makes the result available to other software.

h. The Dependent Claims Are Also Directed to an Abstract Idea

618. Dependent claims 2–9, 11–14, 16–20, 22–31, and 33–41 do not alter the abstract nature of the independent claims. Their additional limitations fall into the groups below. The same types of limitations recur across the four claim families, so I address them in groups.

619. **Display and presentation (claims 2, 22, 28).** Claim 2 recites displaying a portion of the graph database, including a medical condition entry, its diagnostic factor entries, and the links between them. Claim 22 recites displaying each linked condition's total weight value. Claim 28 recites that the display is for patient diagnosis, an administrative medical purpose, or medical preauthorization. These limitations specify what information is displayed and why. They are part of reporting the result of the abstract idea.

620. **Organization of entries (claims 3, 7, 8, 9, 11, 16, 27, 34, 38).** These claims fall into three sets:

- **Entries as nodes and shared factors.** Claim 3 specifies that the entries are nodes. Claim 8 specifies that one diagnostic factor is linked to two conditions. Claims 9 and 38 specify that traversing to different conditions produces different ranks for the same factor.
- **Laboratory tests.** Claim 11 adds a condition-independent laboratory-test entry.
- **Intermediate entries.** Claims 7 and 27 add a demographics entry between a condition and a factor, and claim 27 adds traversing that link only when the patient’s demographics match. Claims 16 and 34 add an intermediate node or factor group entry that holds a rank.

These limitations recite the categories of medical information stored in the graph and where condition-specific values are recorded. As I explained above, those are data-placement choices, not improvements to database technology. *Supra* § X.M.1.f.

621. Ontology identifiers and hierarchy traversal (claims 4, 23, 35).

Claim 4 requires that each diagnostic factor entry include a unique identifier linking to an entry in a medical ontology. Claims 23 and 35 recite traversing upward in a medical ontology link to infer other diagnostic factors present in the patient. Traversing parent–child relationships in a medical ontology is the same hierarchical reasoning a clinician performs when looking up the broader concept “fever” after observing “double quotidian fever.”

622. **Synonyms (claims 5, 24, 36).** Claims 5, 24, and 36 recite that each diagnostic factor entry includes at least one synonym of the diagnostic factor it represents. Recording that different terms refer to the same finding, for example that “being feverish” and “high temperature” both mean fever, is what clinicians and medical terminologies have long done. The ‘882 patent describes synonyms only as alternative names that “allow the EDF to be recognized across different sources that use different terms.” Ex. 1001 at 10:66–11:3. These limitations specify additional content stored in the entry and do not change what the claims are directed to.

623. **Presence, negation, and temporality attributes (claims 6, 25, 37).** These claims recite that diagnostic factor entries include a temporality, negation, or experienter attribute. Recording whether a finding is present, absent, or unknown is part of recording the patient’s clinical information.

624. **Natural language processing (claims 17, 26).** Claims 17 and 26 recite producing diagnostic factors using NLP initialized with the graph database’s diagnostic factor entries or nodes. These limitations specify the tool used to read text. They do not change what the claims are directed to. The ‘882 patent lists only conventional NLP processes, such as word and sentence segmenters, negation cues, and named entity recognition, and describes initializing the NLP engine with ontology terms or graph nodes only as a preference. Ex. 1001 at 7:57–8:3; 12:15–17; 23:7–10.

625. **Identification of unknown factors (claim 33).** Claim 33 recites identifying unknown diagnostic factor entries by traversing down from each condition entry and disregarding the known factor entries. This is the ordinary step of listing a condition’s findings and setting aside those already known.

626. **Graph size and the form of the determination (claims 12–14, 18–20, 29–31, 39–41).** Claims 12, 18, 29, and 39 require more than twenty thousand entries or nodes. Claims 13, 19, 30, and 40 require more than one hundred thousand. These limitations specify the scale of the data stored. Claims 14, 20, 31, and 41 recite that the healthcare determination is “a recommendation, an evaluation, a result, or a conclusion.” These limitations label the output and do not change the fundamental character of the claims.

2. The Challenged Claims Do Not Recite an Inventive Concept

627. When viewed individually or as an ordered combination, the Challenged Claims do not recite an inventive concept that transforms the abstract idea into patent-eligible subject matter. Each element was well known and widely used in the field before the ‘882 patent’s priority date. The prior art references I discuss above disclose each element. *Supra* §§ IX, X.A–X.L.

a. The Core Features of the Independent Claims Were Conventional

628. **The Graph Database.** The claimed graph database is generic and conventional. Building a medical knowledge graph that cross-links conditions and

their associated findings was well established in the art. *Supra* § VI.C. Datla discloses a knowledge graph in which “[e]ach root node of the graph is a symptom, and the remaining nodes are conditions, diagnoses, tests, procedures, medications, or other clinical concepts,” within a system that includes a database. Ex. 1005 at ¶¶ [0031], [0078]. Zhong discloses a knowledge graph in which each node is “an entity, such as a symptom, a dimension, a demographic, and a disease.” Ex. 1009 at 3:20–25. The ‘882 patent does not claim a new type of database or a new way of storing or retrieving data. Ex. 1001 at 8:10–18.

629. **The Data Arrangements.** The data arrangements in the claims are conventional:

- **Condition-independent factor entries.** Keeping diagnostic factor entries free of condition-specific information was a routine design principle;
- **Demographics nodes.** Storing attribute values such as age range or sex in a graph node was a routine technique;
- **Weights and ranks on links.** Storing numerical values on edges, or in intermediate nodes, was a routine choice between well-known graph designs.

630. Datla, Dang, and Zhong each disclose these arrangements, as I explain in my analysis of Grounds 1, 4, 6, and 9. *Supra* §§ X.A, X.D, X.F, X.I. None of these arrangements supplies an inventive concept, because none changes how the database operates; they merely record where medical information is kept.

631. **Medical Evidence Sources and Ontologies.** The medical evidence sources recited in the claims are conventional. Peer-reviewed medical journals, including The BMJ, were well-known sources of clinical evidence, and practitioners routinely built knowledge graphs from them. *Supra* §§ VI.B, VI.C. Datla builds its graph from “any source of information, including but not limited to medical journals.” Ex. 1005 at ¶ [0030]. MacRae is an article published in a BMJ journal.

632. The taxonomies and hierarchies that the claims attribute to the medical evidence sources are supplied by pre-existing ontologies such as SNOMED-CT and LOINC. These are third-party controlled vocabularies that were widely used in clinical informatics for decades before the priority date. *Supra* § VI.A. The specification acknowledges that “multiple clinical ontologies exist, including SNOMED-CT,” and that SNOMED-CT “provides taxonomies of clinical terms.” Ex. 1001 at 4:13–20. Incorporating information from existing sources and ontologies is not an inventive concept.

633. **APIs.** The claimed APIs are generic and conventional. By 2022, API-based querying of medical knowledge graphs to return structured results to software applications for generating healthcare determinations was well known. *Supra* § VI.E. Examples include: (1) BioPortal, which has provided RESTful APIs for querying biomedical ontologies, including SNOMED-CT and LOINC, since 2006; (2) The National Library of Medicine’s UMLS, which has long provided API-based

terminology services; and (3) HL7's FHIR standard, which includes terminology-service APIs. Gnanasambandam discloses an "[a]pplication back-end component 9104 [that] is configured to publish one or more APIs 9103" that "may use artificial intelligence to perform analytics based on a knowledge graph." Ex. 1007 at 100:24–34. The claims recite no particular API structure or technique. The '882 patent identifies only "FHIR, USCDI (United States Core Data for Interoperability), CDS Hooks, and vendor-specific APIs." Ex. 1001 at 6:4–8. Using an API to provide programmatic access to a medical knowledge base was a well-established design pattern, and it does not supply an inventive concept.

634. **Matching, Weighting, Thresholds, Ranking, and Display.** The remaining functional steps of claims 15, 21, and 32 were conventional operations of clinical decision support systems:

- Matching patient data to graph nodes. Datla assigns an activation weight "to a node of the knowledge graph comprising the extracted one or more patient symptoms," and Dang matches input text to graph nodes using semantic similarity. Ex. 1005 at ¶ [0055]; Ex. 1008 at ¶ [0009];
- Summing weights per condition. Dang discloses "Confidence of disease=Sum(Confidence of evidence*TF-IDF)," and adding values was a routine operation. Ex. 1008 at ¶ [0050];

- Applying a probability threshold. Zhong discloses that “the most likely diseases are those diseases that have a probability above a threshold value.” Ex. 1009 at 4:43–46.
- Ranking candidate findings. Zhong sorts candidate symptoms by calculated probability and then by discriminative score. Ex. 1009 at 4:64–66, 5:55–58;
- Using demographics to adjust results. Datla lowers a disease’s rank when it “is not common for a demographic.” Ex. 1005 at ¶ [0061].

635. Displaying ranked results was equally conventional. Presenting results in descending order of relevance or score was a ubiquitous convention in information retrieval and decision support systems at the time of the invention. Displaying diagnostic recommendations organized by or labeled with their associated medical condition was likewise a well-known convention in clinical decision support systems. Datla provides “the ranked medical conditions and/or diagnosis” to the clinician through a user interface, and Dang sorts its possible disease list by probability rate and outputs it through a graphical user interface. Ex. 1005 at ¶ [0062]; Ex. 1008 at ¶ [0056].

636. **Healthcare Determination Output.** Each independent claim requires that the API allow access “in order to generate a healthcare determination,” and claims 15, 21, and 32 require displaying that determination. These limitations describe the purpose of the output. They do not specify any novel mechanism for

achieving it. Outputting a ranked list of likely conditions or a ranked list of questions to ask recites the intended use of the data rather than a technical improvement.

637. When considered as an ordered combination, none of the independent claims recites an inventive concept.

638. Claim 1 recites receiving medical evidence text, extracting diagnostic factors, storing factors and conditions as linked entries in a graph database, and making the entries available through an API. Datla discloses constructing a knowledge graph from a medical corpus. Gnanasambandam discloses publishing APIs for a medical knowledge graph. The combination is a conventional arrangement of conventional components, each performing its ordinary function.

639. Claim 15 adds a demographics node and matching of patient chart data to produce a displayed determination. Datla and Dang disclose matching patient data to graph nodes and using demographic information in diagnosis, and Dang discloses a context node between disease and symptom nodes. *Supra* § X.D.2 (elements [15f], [15h]). The ordered combination is the conventional flow of a knowledge-graph-based diagnostic system: build the graph, match the patient's data, and report the likely condition. The '882 patent describes that same flow. Ex. 1001 at 23:7–10; 24:40–44; FIG. 5.

640. Claim 21 adds receiving EHR data, matching the patient's diagnostic factors, collecting and summing link weights, and displaying conditions in ranked

order. Datla and Dang disclose each of those steps, in the same order, in the same kind of symptom–disease knowledge graph. *Supra* § X.F.2 (elements [21b]–[21k]). The combination adds nothing beyond the sum of its parts.

641. Claim 32 adds applying a probability threshold, identifying known and unknown factors, ranking the unknown factors by a value stored in the link, and displaying them in order. Zhong discloses this same sequence as an iterative question-generation process over a knowledge graph, and Gnanasambandam supplies the API. *Supra* § X.I.2. The combination is a conventional arrangement of conventional components.

642. The prosecution history of the ‘882 patent confirms the absence of an inventive concept. The API limitation the applicant added in response to the § 101 rejection was well known in the prior art, as the references analyzed above demonstrate. The Examiner’s reasons for allowance for each claim group rest on the API combined with the claimed data arrangement. Ex. 1002 at 17–20. As I explained above, both were conventional. *Supra* § X.M.2.a.

b. The Dependent Claims

643. The dependent claims fare no better. Their additional limitations either narrow the abstract idea by specifying what medical information is stored, compared, or reported, or recite conventional computing features.

644. **Display and presentation (claims 2, 22, 28).** Displaying a portion of a knowledge graph, a condition's total weight value, or a display for diagnostic purposes is generic output functionality. Goldberg discloses displaying portions of a medical knowledge graph, including diseases, findings, and their links. Ex. 1006 at 14:32–42. Adding a number already in memory to a display is a routine user-interface choice.

645. **Organization of entries (claims 3, 7, 8, 9, 11, 16, 27, 34, 38).** The groups in these claims are conventional:

- **Shared factors and condition-specific ranks.** Storing one reusable entry for each finding or test, linking it to multiple conditions, and placing condition-specific ranks on links or intermediate entries were standard data-organization choices. Datla discloses shared symptom nodes linked to multiple conditions through weighted edges.
- **Intermediate entries.** Dang discloses intermediate feature nodes and context nodes.
- **Demographic gating.** Comparing a patient's age or sex with a stored value before following an edge is a simple conditional check during graph traversal.

646. **Ontology identifiers and hierarchy traversal (claims 4, 23, 35).** Annotating graph entries with ontology codes was a known technique, and off-the-shelf tools were available for doing it. Kim discloses entries that “may be identified

by a unique ontology code.” Ex. 1012 at ¶ [0041]. Traversing upward in a medical ontology to infer broader concepts is the standard way of navigating a hierarchical terminology such as SNOMED-CT. *Supra* § VI.A.

647. **Synonyms (claims 5, 24, 36).** Storing synonyms in a graph entry was a known technique. Kim teaches that “the graph database includes synonyms, translations, concepts and relationships.” Ex. 1012 at ¶ [0054]. *Supra* §§ X.B, X.G, X.K. The ‘882 patent attributes no technical function to synonyms beyond allowing an entry “to be recognized across different sources that use different terms.” Ex. 1001 at 10:66–11:3. Adding alternative names to an entry is routine vocabulary management, not an inventive concept.

648. **Presence, negation, and temporality attributes (claims 6, 25, 37).** Storing a present/absent/unknown value as a node property is routine. Van Assel discloses assigning “1, -1, or 0 to the extracted concept to indicate whether the concept is validly present (1), validly absent (-1) or this is unknown (0).” Ex. 1011 at ¶ [0245].

649. **Natural language processing and identification of unknown factors (claims 17, 26, 33).** NLP engines for extracting clinical concepts were conventional, and Datla discloses one. Ex. 1005, claim 1. Matching extracted concepts to existing graph nodes is what Dang’s semantic matching does. Ex. 1008 at ¶ [0009]. Identifying unknown factors as the complement of the known set is what Zhong

discloses. Ex. 1009 at 3:52–54. The ‘882 patent describes NLP initialization only as a preference and discloses no particular technique for performing it. Ex. 1001 at 12:15–17 (NLP engine “[p]referably” initialized using a lexicon such as an ontology); 23:7–10 (“[p]referably, the NLP engine is initialized with the EDF and EDI nodes from the graph database”).

650. Graph size and the form of the determination (claims 12–14, 18–20, 29–31, 39–41). The node-count thresholds merely specify scale. The ‘882 patent attaches no technical significance to them and states only that “an actual graph database will be much more complex and contain many hundreds of thousands of nodes or more.” Ex. 1001 at 27:67–28:3. Graph and ontology-based systems with hundreds of thousands of concepts, including SNOMED-CT itself, were routine. Labeling the output as a recommendation, evaluation, result, or conclusion adds no technical mechanism.

3. Conclusion

651. In summary, the Challenged Claims recite the abstract idea of collecting, organizing, and analyzing medical data in a knowledge graph and making the results available to software. None of the claim elements, individually or as an ordered combination, provides an inventive concept that transforms the claims into patent-eligible subject matter. Every component of the claimed systems was well known and widely used in clinical informatics before the ‘882 patent’s priority date.

XI. SECONDARY CONSIDERATIONS

652. I have found no evidence either in the '882 patent's prosecution history or elsewhere that supports any secondary considerations arguments, or evidence of nexus to any Challenged Claim. As I have discussed herein, any purported solutions to problems or unexpected results in the '882 patent were already well known in the art.

653. To the extent Patent Owner asserts the existence of secondary considerations, or presents any additional evidence or arguments, I reserve the right to address any such arguments and evidence.

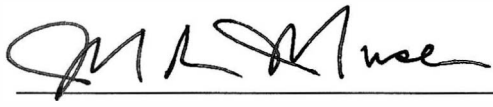
XII. ADDITIONAL REMARKS

654. The opinions I have expressed in this Declaration are based on the information and materials currently available to me. I reserve the right to supplement, amend, or modify my opinions if additional information becomes available, including any arguments, evidence, or declarations submitted by Patent Owner or any other party in this proceeding.

655. I am prepared to provide testimony on the matters addressed in this Declaration if called upon to do so by the Board or any party in this proceeding.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Dated: October 3, 2026

By: 
Mark A. Musen, M.D., Ph.D.