COTriage: Applying a Model-Driven Proposal for Improving the Development of Health Information Systems with Chatbots

Julián A. García-García, Nicolás Sánchez-Gómez, and María José Escalona, *University of Seville, 41012 Seville, Spain*

Mercedes Ruiz, *University of Cádiz, 11519 Puerto Real (Cádiz), Spain*

Today, organizations require innovative and flexible solutions to digitize and automate their processes—particularly those processes designed to obtain information from users. Chatbots are one of the most widely used technological options for automating processes. This article aims to integrate chatbot technology with health information systems (HISs) to improve the execution of health-care processes. Specifically, it presents the COTriage framework, which proposes model-driven mechanisms for improving the design and development of process-oriented HISs with chatbot-based triage process integration. The proposal was also instantiated with the COVID-19 triage process and assisted reproduction treatment processes on iMedea (a real HIS). Finally, the proposal was discussed considering the acceptance of end users, as well as the degree of efficiency and effectiveness achieved by the software team who applies COTriage on our case study.

Convergence between digital innovation and process automation can be seen as a relevant factor in improving digitization in companies[1] and enhancing their positioning in markets.[2] It is especially important for companies when innovative technological solutions become competitive, mature, and user friendly.[3] However, such convergence should also involve mechanisms for improving the usability of processes executed by human actors (clients or users).

Chatbots currently constitute one of the most widely used technological options for automating processes as a means of digital business transformation[4]; particularly processes designed to obtain information from users.[5] However, the development of chatbots presents challenges for development teams because it requires special skills (e.g., machine learning and conversation design) that differ from the development of traditional software systems.[6] At the same time, the challenges faced by chatbot developers remain largely unknown, as most existing studies focus on proposing chatbots to perform specific tasks and not on their development.[6] Despite these challenges that exist today, chatbots offer well-known benefits[7] in terms of personalized assistance, cost-effective solutions, flexibility, improved user experience, and so on.

These benefits can also be instantiated in the health-care field[8] where chatbots improve user privacy, reduce the workload of health-care professionals, reduce waiting times for patients, facilitate decision-making, and so on. In some health-care scenarios, however, chatbots are not in themselves able to achieve these benefits or objectives unless they are integrated with health information systems (HISs). For example, chatbots could be used to automate clinical recommendation triage processes[a,9] with patients presenting particular symptoms; the result of the chatbot-based triage could be stored in the patient's medical record for consultation by health-care professionals or it could also automatically activate a specific medical protocol.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Digital Object Identifier 10.1109/MITP.2024.3380916
Date of current version 21 June 2024.

[a]Clinical triage is a selecting and classifying manual method of patients that is mainly used by health-care professionals to 1) assess the immediate risk of death and 2) prioritize care considering the patient's symptoms and pathology, his/her therapeutic needs, and available health resources.

The need to automate health-care processes has always been a challenge for health-care organizations wishing to improve productivity, reduce costs, and improve patient care. The need emerged abruptly in 2020 due to the worldwide SARS-CoV-2[10] pandemic (COVID-19), which seriously and continuously strained health-care services[11] while at the same time revealing weaknesses in global health services due to a lack of human and financial resources. Information and communication technologies (ICTs) have been crucial in mitigating the pandemic's impact,[12] but the tasks of designing and developing HISs with chatbots are complex and require time. We argue that the adoption of chatbots further depends on the ability of the chatbot-based solution (and its underlying health-care process model) to learn and meet the health-care organization's goals and constraints (e.g., reliably integrating clinical patient data between systems by improving the maintainability and quality of these systems).

In respect, the contribution of this article is twofold. On the one hand, it proposes the COTriage framework, which proposes model-driven mechanisms for improving the design and development of a chatbot-based processes and its integration with a process-oriented HIS. Our proposal is focused on a task-oriented chatbot, which allows tasks to be performed within a closed domain using short conversations. On the other hand, this article also presents a supporting tool which facilitates the applicability of our framework in real software engineering environments. In this sense, a case study is instantiated with the COVID-19 triage process and its integration with assisted reproduction treatment[b] (ART) processes[13] on a real HIS (called iMedea[c]), which is used by the Assisted Reproduction Unit of the Spanish clinic Inebir, Inc.[d] After this instantiation, the proposal is discussed considering the acceptance of end users, as well as the degree of efficiency and effectiveness achieved by the software team who applies the COTriage proposal on our case study.

Following this introduction, this article is organized as follows. The "Previous Scenario and Motivation" section describes the previous scenario and our motivation, which have motivated the research for this proposal. Subsequently, the "Overview of the COTriage Framework" section presents the conceptual and theoretical scheme of the COTriage model-driven framework. Later, the "COTriage Modeling Metamodel" and "Supporting Tool and Case Study" sections present the COTriage metamodel to jointly model health-care processes and chatbot-based triage processes, and the supporting tool for applying this framework in practical environments, respectively. Finally, the "Discussion and Threads" and "Conclusions and Future Works" sections present a brief discussion and the main conclusions drawn from this study and sets out some strategic considerations regarding future lines of research, respectively.

PREVIOUS SCENARIO AND MOTIVATION

Health-care professionals and public and private health-care entities have been subjected to intense and continuous health-care stress due to the COVID-19 pandemic experienced worldwide (mainly from 2020 to 2022), but technological solutions have tried to mitigate this impact.

In this context, COTriage was born to mitigate this impact and reduce the overload of health-care professionals using ICT solutions, as well as improve the effectiveness and efficiency of software experts to develop these technological solutions. For this purpose, COTriage proposes model-driven mechanisms with which to automate the development and design of chatbot-based triage process and its integration with process-oriented HIS. This need was identified after studying the daily work of various health clinics and their HISs. Inebir is a private human fertilization clinic and was one of these studied clinics.

Table 1 presents the average effort (measured in hours and before applying our proposal) of the team of experts and end users to define requirements, design, and develop new ART processes in iMedea (without automatic support for triage processes). This effort is measured in total average hours required by each profile, with an average dedication of eight hours per day; functional analysts, developers, and end users dedicated 60%, 100%, and 30% of the time, respectively. After analyzing this scenario, it is possible to observe that 1638 h was the average time needed to implement an ART process (with average complexity) within iMedea. It is important to note that this process does not include automatic functionalities to execute triage processes, but it contains basic forms to record data.

On the one hand, Inebir uses the HIS called iMedea, which manages and supports the electronic health

[b]ART processes are clinical treatments that have become a very used health service by millions of sentimental couples around the world because of fertility-limiting pathologies, delay in the maternity age, single-parent couples, homosexual couples, and women who wish to face maternity individually, among others.[13]

[c]iMedea: A specialized HIS for assisted reproduction clinics. Website: https://g7innovation.com

[d]Inebir is one of the leading health-care entities in reproductive medicine in Andalusia (Spain), with more than 1200 ART cycles performed annually and more than 5500 ART cycles performed in total since 2020. Website: https://inebir.com/en/

TABLE 1. Previous situation: Average initial costs to design and develop functionalities which support a new ART process and triage processes (supported with basic forms to record data).

Role	Dedication (%)	Hours per day	Total (hours)	Software task	
				Gathering requirements & design	Development
Functional analyst	60	4,8	567	469	98
Developer 1	100	8	894	0	894
Developer 1	100	8	894	0	894
Final user	35	2,8	203	203	0
		Total	**2558**	**672**	**1886**

record (EHR) from a process-based perspective. This system was designed and developed by a team of software experts which was mainly composed by a functional analyst and two developers. This team participated in software development tasks to extend iMedea and support 17 ART processes, which have different complexity levels. This complexity could be established considering different parameters (such as total number of activities, number of stakeholders who are involved in the process, lines of source code to run each ART process, number of integration points with external systems, and so on), but in this article, we focus on number of activities, stakeholders, and lines of source code. In this sense, the average complexity of these ART processes is 22 activities, four stakeholders, and 11,356 lines of source code.

On the other hand, from the perspective of the triage process, these ones were carried out manually and face to face between patients and health-care professionals. During the first months of the pandemic, the number of professionals involved in triage processes increased by 50% compared to the prepandemic situation. This increase was due to the high care stress and the impossibility of suspending ongoing ART processes. In addition, Inebir also increased its staff costs due to new staff hiring to cover temporary sick leave of employees infected by COVID-19. This situation also caused psychological tensions among the clinic staff and in its organizational structure due to the high probability of contagion and uncertainty due to the severity of the disease. Another usual situation was the incorrect introduction of the result of the triage process due to the physical and mental exhaustion of health-care professionals, what usually implied delays in the dissemination of the results of the triage process. This situation was caused because the nursing staff spent a lot of time manually recording the results into the clinic's system, as well as linking these results with the fertilization treatment in the patient's EHR.

After considering this previous scenario, this article presents a proposal to mitigate this situation and facilitate the automation of triage processes (using chatbot-based technologies), as well as improve the acceptance of end users, the quality, and reduce startup time of HIS systems which support these automatic triage processes. For this purpose, this article proposes a model-driven framework that is supported by computer-aided software engineering tools which allow us to apply our framework in real environments.

OVERVIEW OF THE COTRIAGE FRAMEWORK

The COTriage framework considers the benefits (e.g., in terms of quality, efficiency, effectiveness, and maintainability) of applying model-driven principles as opposed to traditional software development processes[14] to achieve our objective.

Figure 1 shows an overview of the COTriage framework. It is based on three phases in which health-care professionals and software engineers collaborate to achieve the objective as follows:

1) *Discovery and modeling phase*: Health-care professionals model health care and triage processes for a specific pathology instantiating the *COTriage Metamodel* [c.f., Figure 1(a)]. The participation of software engineers may be necessary as a modeling support team but is not mandatory because COTriage is supported by user-friendly modeling tools.
2) *Development phase*: Our proposal defines model-to-model transformation rules [c.f., Figure 1(b)] to systematically enrich the *COTriage Model* with platform-independent execution features

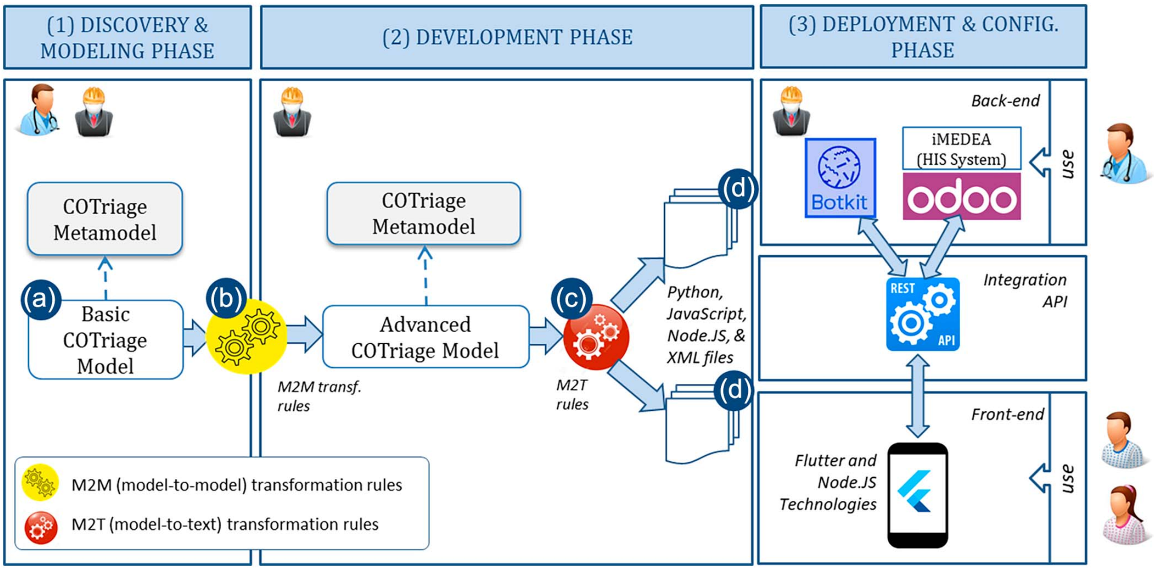

FIGURE 1. Model-driven framework of COTriage. API: application programming interface.

associated with health-care and triage processes. These features were set by the software engineer to define execution parameters for the subsequent running. Model-to-text rules were then applied to systematically obtain the initial version of executable code [c.f., Figure 1(c)], which was instantiated in a specific technological infrastructure: 1) a real HIS (iMedea) supported by ERP Odoo (which is based mainly on Python and XML); 2) an integration layer based on representational state transfer services; 3) a chatbot engine supported by the Botkit framework and Node.JS; and 4) a user app based on Flutter and JavaScript.

3) *Deployment and configuration phase*: In this phase, the software engineer completes the platform-dependent configuration of the executable code [c.f., Figure 1(d)] generated in the previous phase to manually deploy it in the final technological infrastructure.

In our opinion, this COTriage framework provides three main benefits: 1) its metamodel includes different dimensions of health-care process (HP) and chatbot-based triage process modeling (i.e., definition and execution), facilitating the reuse of each dimension across target technologies; 2) the metamodel is totally independent of specific platforms, thus improving the maintenance and evolution of the chatbot; and 3) it can be easily extended to support new technological platforms including new transformation rules and modeling concepts. These benefits contribute to the quick, easy design of new COTriage extensions.

COTRIAGE MODELING METAMODEL

The COTriage metamodel provides primitives to jointly model health-care processes and chatbot-based triage processes in terms of their definition and execution. In this context, artifacts of the triage process are defined as logical decisions, which provide a complete decision logic model complementing the health-care process model.

The definition of the COTriage metamodel is carried out with two main components[15]: 1) an abstract syntax (platform-independent metamodel), which defines modeling concepts and their relationships, and 2) a concrete syntax [in the form of Unified Modeling Language (UML)-Profile] to use our metamodel in real environments. Space restrictions make it impossible to describe the *COTriage Metamodel* in detail, but it is briefly illustrated in Figure 2.

The two main concepts of the COTriage metamodel are represented by the *HealthcareProcess* and *TriageBot* metaclasses.

On the one hand, the process consists of an ordered set of elements represented by the *ProcessElement* metaclass, the purpose of which differs according to its semantics and specialization: *ControlElement*, which make it possible to establish the different paths from the start node to the end node; and *Activity*, which represents an action to be executed in the process. Our metamodel also considers two types of activity: 1) *OrchestrationActivity*, representing an automatically performed activity; and 2) *HumanActivity*, representing

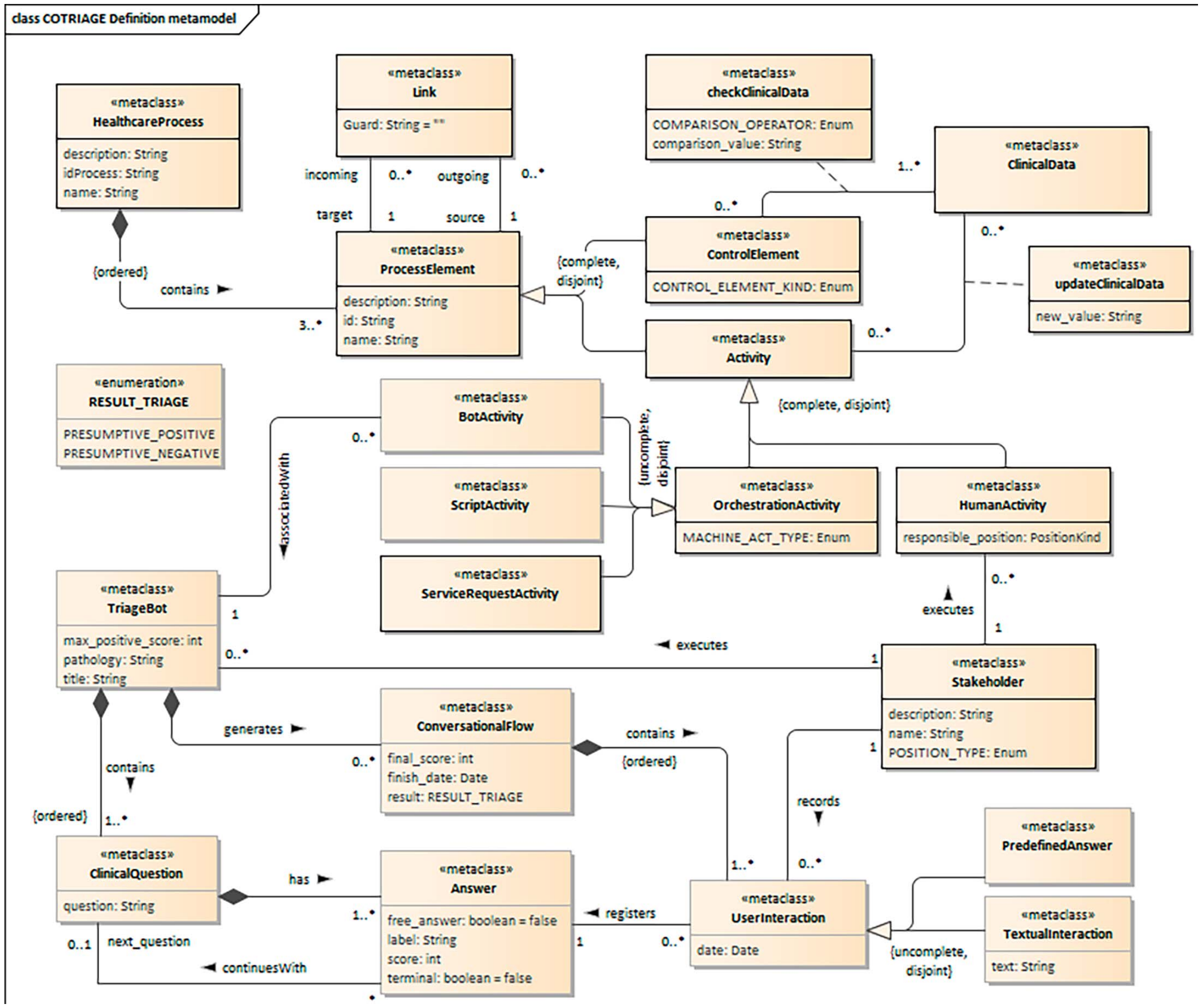

FIGURE 2. COTriage modeling metamodel (simplified excerpt).

an activity performed by a human actor (i.e., the *Stakeholder* metaclass).

On the other hand, the *TriageBot* metaclass represents a chatbot-based triage process associated with the *BotActivity* metaclass (a specialization of *OrchestrationActivity*), which represents a machine task performed by COTriage's chatbot engine. The *TriageBot* metaclass contains an ordered set of clinical questions and can generate a set of conversational flows representing the ordered structure of the patient–chatbot dialogue. Each possible answer to a specific clinical question has an associated score, intended to represent how decisive a symptom is with a view to establishing a presumptive positive for the pathology suffered by the patient. Likewise, a conversational flow contains the result of the triage process (among other attributes) and represents the set of user interactions between the patient and the chatbot. These interactions (represented by the *UserInteraction* metaclass) represent the record of a response provided by the stakeholder within a conversational flow with TriageBot. The proposal also establishes two types of interactions: 1) the *PredefinedAnswer* metaclass, representing the selection of a predefined answer for a specific clinical question; and 2) the *TextualInteraction* metaclass, representing information entered manually by the patient (i.e., stakeholder).

SUPPORTING TOOL AND CASE STUDY

As introduced before, the COTriage proposal includes a modeling tool to support and facilitate its applicability in real software engineering environments. This modeling tool allows to instantiate the COTriage metamodel and generate executable code from these instances. The COTriage modeling tool has been integrated into Enterprise Architect (EA) to facilitate its application in real software engineering environments because EA provides a suitable technological foundation to implement our model-driven framework thanks to its UML extension and model-driven engineering mechanisms, as well as its wide presence in software companies.

Space restrictions make it impossible to explain and show many images of the COTriage supporting tool, but it is briefly described using a real and successful use study as vector director, which is framed in the health-care environment within Inebir (Spanish private assisted reproduction clinic). Specially, COTriage was applied to perform the ART process for fertilization treatment with oocyte donation. In short, before starting this fertility process, it is necessary for the patient and the donor to be evaluated by their gynecologists in their first face-to-face medical visit. Both must go through a COVID-19 triage process before attending this appointment.

This oocyte donation process was modeled by software engineers and health-care professionals using the COTriage modeling tool. The process is complex because it contains 25 activities and 32 clinical questions associated with the COVID-19 triage process, and it also involves many stakeholders (patient, gynecologists, nurse, and embryologist, among others), but Figure 3 presents a simplified view of this model and shows the user working space of the COTriage modeling tool. It allows building models visually and graphically by dragging and dropping elements from the COTriage toolbox [Figure 3(a)] obtained after implementing the COTriage UML-Profile. Figure 3(b) also shows a structured view of the elements associated with the health-care and triage process models defined by the user. Figure 3(c) also presents a simplified view of our case study, which was validated by professionals belonging to Inebir.

After modeling the health-care and triage processes, our EA plugin for COTriage automatically generates skeleton source code according to Odoo (Python and XML) and Botkit (mainly JavaScript) technologies, as well as Python code (database structure and APIRest) associated with bot activities. As mentioned in the "Overview of the COTriage Framework" section, these files provide skeleton source code, which must be completed by the software engineer to finalize its configuration.

Once configured, the source code files are ready to be deployed in the target HIS. Space restrictions make it impossible to explain and show full the source code associated with our case study, but Figure 3(d) shows an example of Python code snippet. Specifically, this source code is associated with the COTriage APIRest (simplified excerpt). This controller was automatically

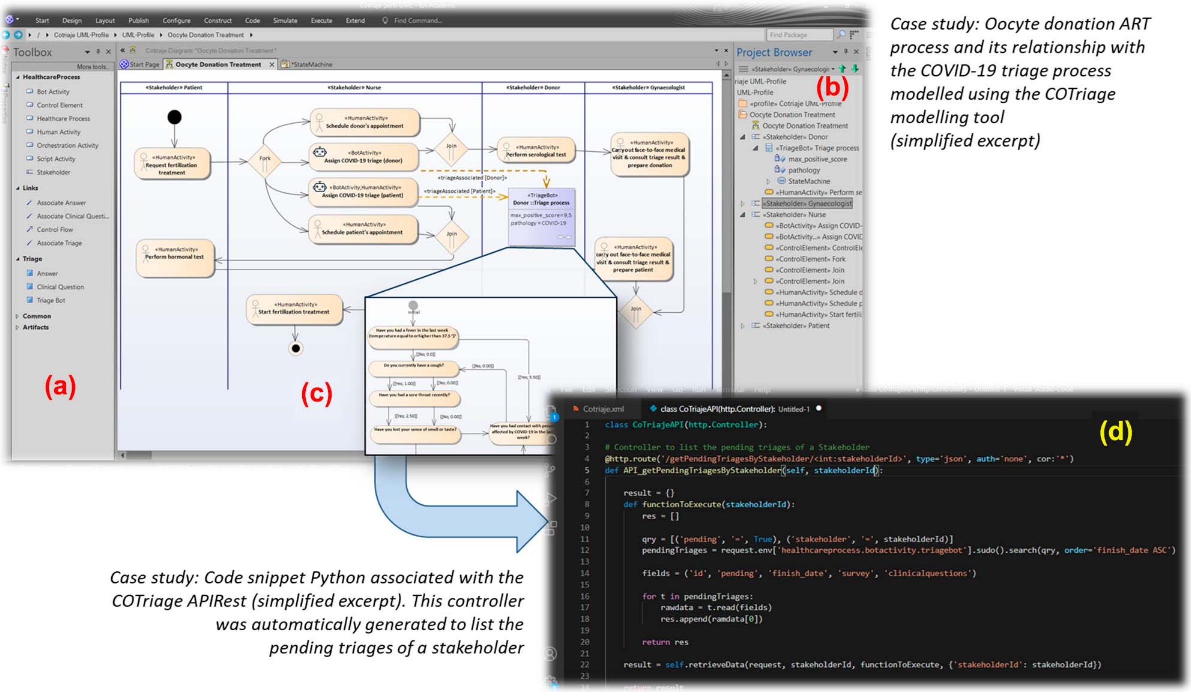

FIGURE 3. COTriage supporting tool and model of the case study (simplified excerpt).

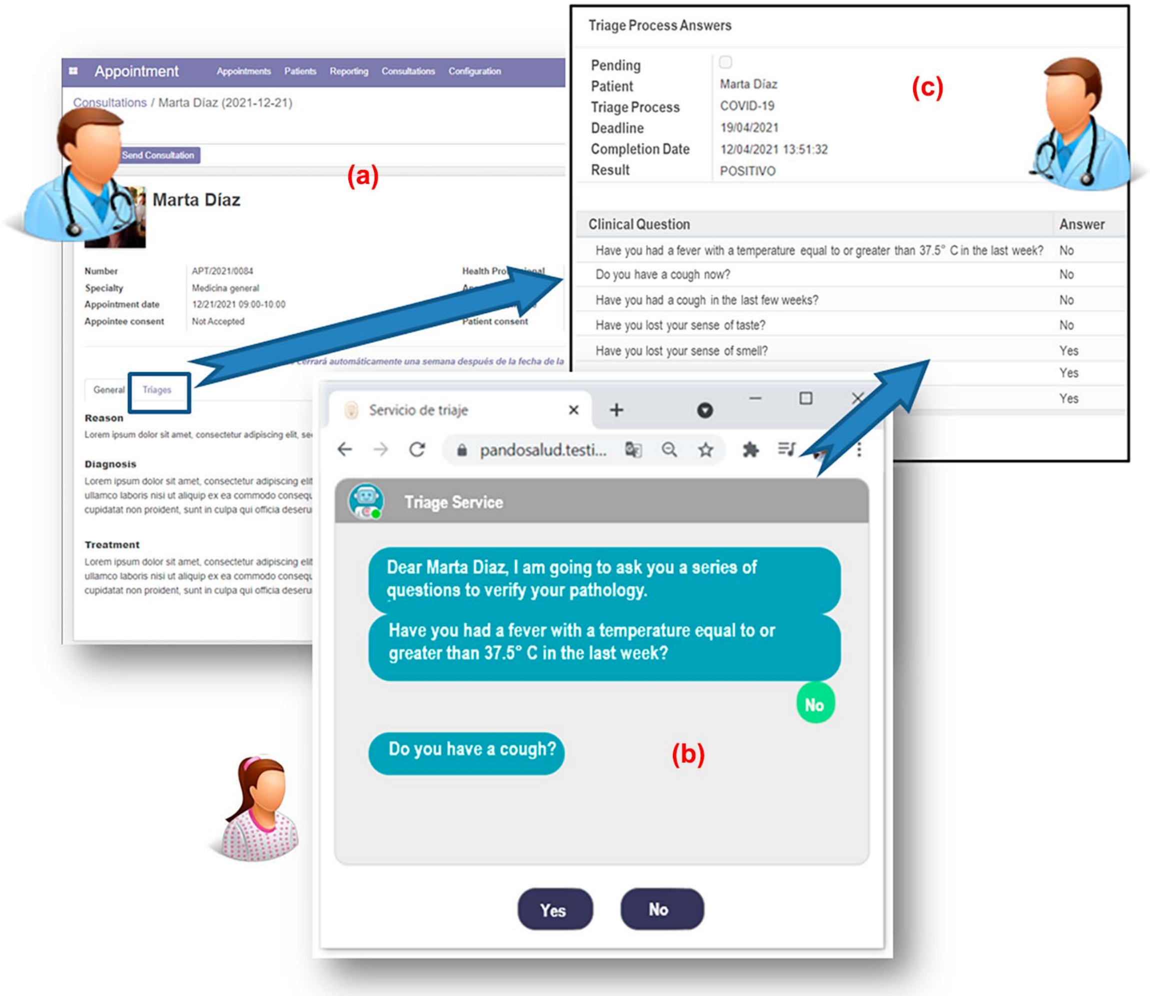

FIGURE 4. Screenshots obtained from the software system after applying the COTriage proposal. (simplified excerpt).

generated to list the pending triages of a stakeholder, and once this code is completed, this one is manually deployed in the COTriage technological platform (c.f., Figure 1).

Moreover, Figure 4 shows some screenshots of the final information system developed with the COTriage proposal (some marks have been added to facilitate understanding of this figure).

Figure 4(a) and (b) is related to the face-to-face medical visit between the patient and her doctor and the assignment of COVID-19 triage to the patient (c.f., Figure 3), respectively. The first one is a human activity and is executed by the health-care professional within the iMedea system. Its aim is to record specific clinical information about the patient. The second one is a bot activity and is executed by the patient using the chatbot engine of the COTriage proposal. Its purpose is to record the patient–bot interaction and to propose the preliminary diagnosis of a specific pathology (in our use case, we identified the COVID-19 pathology) according to the modeled triage process.

Finally, Figure 4(c) shows the integration of the result of the triage and the health-care process associated with the previously mentioned activities.

DISCUSSION AND THREADS

This section discusses benefits and challenges of applying the COTriage proposal to value its suitability on real environments. For this purpose, we compare the previous situation (c.f., the "Previous Scenario and Motivation" section) and are able to measure the degree of efficiency and effectiveness achieved when the COTriage proposal was applied on the real case study introduced in the "Supporting Tool and Case Study" section.

TABLE 2. Current situation: Costs to carry out the instantiation of COTriage on our case study: support of ART processes and its integration with chatbot-based automatic triage processes.

Role	Dedication (%)	Hours per day	Total (hours)	Software task	
				Gathering requirements & design	Development
Functional analyst	60	4,8	252	211	40
Developer 1	100	8	548	0	548
Developer 2	100	8	548	0	548
Final user	35	2,8	87	87	0
		Total	**1435**	**298**	**1136**

On the one hand, regarding the efficiency of the team of software experts, the improvement was clear. Table 2 presents hours used by each profile to carry out the requirement gathering, design, and development tasks of the new functional module related to the case study. As mentioned before, the case study includes the design and development of the oocyte donation ART process and its integration with the COVID-19 disease triage processes (based on automatic chatbot technologies). The dedication of each profile shown in Table 2 coincides with the previous scenario (c.f., Table 1), but it is possible to observe a reduction in the time dedicated to obtain the new system software: 1) regarding the requirement gathering and design tasks, the work team spent 298 total hours (which meant a reduction of 55.60% compared to the previous scenario; 2) regarding the development task, this one was carried out during 1136 total hours by the software team (which represented a reduction of 39.75% of the time).

On the other hand, regarding the efficiency metric, it is important to mention that thanks to the COTriage proposal, it has been possible to automatically obtain the 47.24% associated with the source code of the study case. Specifically, the proposal automated the obtaining of 5365 lines of source code associated with the case study.

Obviously, the previous measures are only a case study, but the instanced processes have a complexity close to the average complexity of the processes implemented before using the proposal. Nevertheless, this validation offers very attractive results to continue with the improvement and refinement of the mechanisms driven by COTriage models.

Finally, it is possible to discuss the benefits from the perspective of health professionals and the clinic as a health organization. The chatbot-based technological solution reduced the number of health personnel assigned by Inebir to assess the COVID-19 symptoms of patients. In fact, the personnel assigned to these tasks is similar to the personnel involved during the pre-COVID-19 stage. In addition, this reduction in the number of professionals (who were dedicated to detecting COVID-19 symptoms in patients) was thanks to the automatic chatbot tool because each patient could answer the questions of the triage process themselves without the initial presence of a health professional. If the result was presumptive positive, the patient was referred to their reference hospital for COVID-19 treatment. This workflow and information streamlined communications and results, reducing the risk of infection through direct contact between patients and health-care professionals.

Moreover, the advantages in the efficiency (based on dedication) described in Table 2 is obviously quite encouraging. However, it is possible to detect some issues in the application of our approach, as follows:

1) The use of model-driven paradigm facilitates the daily work of analysts and software experts, but concepts such as metamodels, UML profiles and transformations are very abstract and complex for them. Software experts recognized that they used the approach without understanding these concepts and were just following the process.
2) This situation is aggravated when nontechnical users are aware of these concepts. Our modeling tool mitigates this effect, but we have detected comprehension problems at the beginning. In this sense, we will extend our proposal in future versions to reduce the cognitive load and improve the user experience.

3) The team of experts also highlight that the proposal guides them to establish an objective and structured way to carry out the design and development of chatbot solutions for triage, but the proposal is limited to specific technologies. However, even if they do not have much experience, they recognize that they feel directed and coordinated.

CONCLUSION AND FUTURE WORK

Convergence between digital innovation and process automation can be seen as a relevant, necessary factor in improving digitization in companies, particularly in processes designed to obtain information from customers or users. For health-care organizations, the need to automate HP to reduce costs and improve patient care has always been a challenge. This need emerged abruptly and rapidly in 2020 due to the worldwide SARS-CoV-2 pandemic.

ICTs have been crucial to mitigating the pandemic's impact, but the design and development of HIS are complex tasks and require time. This complexity increases when it is necessary to integrate chatbot technologies. This article presents the COTriage framework, which proposes model-driven techniques and tools for improving the design and development of process-oriented HIS and their integration with chatbot-based triage processes.

The COTriage approach was also instantiated in a real context on the iMedea system via a triage process associated with COVID-19. After this instantiation, the proposal was discussed considering metrics such as the degree of efficiency and effectiveness achieved by the software team who applies the COTriage proposal on our case study. Regarding the efficiency metric, it has been possible to automatically obtain the 47.24% associated with the source code of the study case (obtaining of 5365 lines of source code associated with the case study). Regarding effectiveness, the software team spent 298 total hours (which meant a reduction of 55.60% compared to the previous scenario) in requirement gathering and design tasks, as well as 1136 total hours (which represented a reduction of 39.75% of the time) in development tasks. These measures are only a case study, but this validation offers very attractive results to continue with the improvement and refinement of the mechanisms driven by COTriage models.

As future work, we plan to extend our framework with 1) chatbot creation tools, 2) mechanisms to extend COTriage with new platform-independent concepts and new transformations to generate code for other chatbot technologies, 3) integrate machine learning libraries to generalize our triage process engine, and 4) include new mechanisms for interaction with users (e.g., images). We also plan to generalize the COTriage framework for application and validation in other business contexts (e.g., education, industry, and e-commerce).

ACKNOWLEDGMENTS

This research was supported by 1) the EQUAVEL project PID2022-137646OB-C31 and PID2022-137646OB-C33 both funded by MICIU/AEI/10.13039/501100011033 and by FEDER, UE and 2) the Early Testing in Medical Robotics Process Automation (TRoPA) project (CEI-12) of the Andalusian Regional Government's Ministry of Economy, Knowledge, Enterprise, and Universities (CECEU). Finally, the authors declare that this research has been carried out in accordance with the General Data Protection Regulation (GDPR) on personal information privacy in the European Union.

REFERENCES

1. A. Van Looy, "A quantitative and qualitative study of the link between business process management and digital innovation," *Inf. Manage.*, vol. 58, no. 2, 2021, Art. no. 103413, doi: 10.1016/j.im.2020.103413.
2. T. Grisold, J. Vom Brocke, S. Gross, J. Mendling, M. Röglinger, and K. Stelzl, "Digital innovation and business process management: Opportunities and challenges as perceived by practitioners," *Commun. Assoc. Inf. Syst.*, vol. 49, no. 1, 2021, Art. no. 27, doi: 10.17705/1CAIS.04927.
3. L. Cannavacciuolo, G. Capaldo, and C. Ponsiglione, "Digital innovation and organizational changes in the healthcare sector: Multiple case studies of telemedicine project implementation," *Technovation*, vol. 120, Feb. 2023, Art. no. 102550, doi: 10.1016/j.technovation.2022.102550.
4. A. Miklosik, N. Evans, and A. M. A. Qureshi, "The use of chatbots in digital business transformation: A systematic literature review," *IEEE Access*, vol. 9, pp. 106,530–106,539, 2021, doi: 10.1109/ACCESS.2021.3100885.
5. A. Rapp, L. Curti, and A. Boldi, "The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots," *Int. J. Human-Comput. Stud.*, vol. 151, Jul. 2021, Art. no. 102630, doi: 10.1016/j.ijhcs.2021.102630.
6. A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, "Challenges in chatbot development:

A study of stack overflow posts," in *Proc. 17th Int. Conf. Mining Softw. Repositories*, Jun. 2020, pp. 174–185, doi: 10.1145/3379597.3387472.
7. H. Mydyti and A. Kadriu, "The impact of chatbots in driving digital transformation," *Int. J. E-Services Mobile Appl. (IJESMA)*, vol. 13, no. 4, pp. 88–104, 2021, doi: 10.4018/IJESMA.2021100106.
8. T. R. Spiller et al., "Development of health care workers' mental health during the SARS-CoV-2 pandemic in Switzerland: Two cross-sectional studies," *Psychol. Med.*, vol. 52, no. 7, pp. 1395–1398, 2022, doi: 10.1017/S0033291720003128.
9. "Coronavirus disease 2019 (COVID-19): Situation report," World Health Organization, Geneva, Switzerland, 2020. [Online]. Available: https://iris.who.int/bitstream/handle/10665/331686/nCoVsitrep02Apr2020eng.pdf?sequence=1
10. L. Wilson and M. Marasoiu, "The development and use of chatbots in public health: Scoping review," *JMIR Human Factors*, vol. 9, no. 4, 2022, Art. no. e35882, doi: 10.2196/35882.
11. L. A. Wolf, A. M. Delao, K. Evanovich Zavotsky, and K. M. Baker, "Triage decisions involving pregnancy-capable patients: Educational deficits and emergency nurses' perceptions of risk," *J. Continuing Educ. Nurs.*, vol. 52, no. 1, pp. 21–29, 2021, doi: 10.3928/00220124-20201215-07.
12. Y. Elgazzar, R. El-Shahawy, and Y. Senousy, "The role of digital transformation in enhancing business resilience with pandemic of COVID-19," in *Proc. ITAF Digit. Transformation Technol.*, Singapore: Springer-Verlag, 2022, pp. 323–333, doi: 10.1007/978-981-16-2275-5_20.
13. S. Prochowska and W. Nizański, "Infertility in Toms: Clinical approach, experiences and challenges," *J. Feline Med. Surg.*, vol. 24, no. 9, pp. 837–846, 2022, doi: 10.1177/1098612X221118753.
14. C. Verbruggen, and M. Snoeck, "Practitioners' experiences with model-driven engineering: A meta-review," *Softw. Syst. Model.*, vol. 22, no. 1, pp. 111–129, 2023, doi: 10.1007/s10270-022-01020-1.
15. A. Iung et al., "Systematic mapping study on domain-specific language development tools," *Empirical Softw. Eng.*, vol. 25, no. 5, pp. 4205–4249, 2020, doi: 10.1007/s10664-020-09872-1.

JULIÁN A. GARCÍA-GARCÍA is a professor and researcher in the Department of Computer Languages and Systems, University of Seville, 41012 Seville, Spain. His research interests include the areas of software engineering applied in the health-care environment, business process management, model-driven engineering, and quality assurance. García-García received his Ph.D. degree in computer science from the University of Seville. Contact him at juliangg@us.es.

NICOLÁS SÁNCHEZ-GÓMEZ is a professor in the Department of Computer Languages and Systems, ETSII, University of Seville, 41012 Seville, Spain. His research interests include people management, ICT project management, customer management, and practical application of methodologies and techniques of computer engineering. Sánchez-Gómez received his Ph.D. degree in computer engineering from the University of Seville. Contact him at nicolassg@us.es.

MARÍA JOSÉ ESCALONA is a full professor with the Department of Computer Languages and Systems, University of Seville, 41012 Seville, Spain. Her research interests include software engineering, specifically to software requirements, software early quality assurance, and early software testing. Escalona received her Ph.D. degree in computer engineering with international accreditation from the University of Seville. Contact her at mjescalona@us.es.

MERCEDES RUIZ is an associate professor in the Department of Computer Science and Engineering, University of Cadiz, 11519 Puerto Real (Cádiz), Spain. Her research interests include gamification, serious games, simulation modeling, and data-driven decision making. Mercedes received her Ph.D. degree in computer science from the University of Seville. Contact her at mercedes.ruiz@uca.es.